Subtleties in Security Definitions for Predicate Encryption with Public Index

Johannes Blömer and Gennadij Liske[*]

Paderborn University, Germany
bloemer@upb.de, gennadij.liske@upb.de

Abstract. We take a critical look at established security definitions for predicate encryption (PE) with public index under chosen-plaintext attack (CPA) and under chosen-ciphertext attack (CCA). In contrast to conventional public-key encryption (PKE), security definitions for PE have to deal with user collusion which is modeled by an additional key generation oracle. We identify three different formalizations of key handling in the literature implicitly assumed to lead to the same security notion. Contrary to this assumption we prove that the corresponding models result in two different security notions under CPA and three different security notions under CCA. Similarly to the recent results for PKE and conventional key-encapsulation mechanism (KEM) (Journal of Cryptology, 2015) we also analyze subtleties in security definitions for PE and predicate key-encapsulation mechanism (P-KEM) regarding the so-called "no-challenge-decryption" condition. While the results for PE and PKE are similar, the results for P-KEM significantly differ from the corresponding results for conventional KEM. Our analysis is based on appropriate definitions of semantic security and indistinguishability of encryptions for PE under different attacks scenarios. These definitions complement related security definitions for identity-based encryption and functional encryption. As a result of our work we suggest security definitions for PE and P-KEM under different attack scenarios.

Keywords: predicate encryption with public index, predicate key-encapsulation mechanism, semantic security definition, indistinguishability definition, chosen-plaintext attack, chosen-ciphertext attack

1 Introduction

Cryptographic primitives and schemes considered in modern cryptography have increasingly become more complex over the last decades. Clearly, one has to abstract from many details when novel constructions or techniques are presented. Therefore, it is all the more important that security models are rigorously studied and made accessible to the cryptographic community through comprehensive and perspicuous explanation, especially when these models are translated into novel contexts. Recently, issues in one of the best-studied security definitions for public-key encryption (PKE) under chosen-ciphertext attack were detected [5]. For digital signatures and for symmetric-key encryption the well-known security definitions were reconsidered in [15]. These works showed that the mentioned established definitions do not provide a comprehensive model of adversarial behavior and fail to take into account many attack scenarios. If even these security definitions suffer from weaknesses, what about newly and more involved cryptographic schemes and their security models which are often justified by their origin in *well-known* definitions?

In this paper we look at predicate encryption (PE) with public index and predicate key-encapsulation mechanism (P-KEM) with public index. On the one hand, these cryptographic primitives have been extensively studied due to their usefulness in various cryptographic applications. On the other hand, established security definitions for PE and P-KEM lean to a large extent on the corresponding security definitions in context of PKE and KEM even though the functionality of predicate-based schemes is more complex compared to conventional PKE. We analyze the security definitions for PE and P-KEM, formalize the assumed adversarial behavior and identify several subtleties in the established definitions. As a result we propose well-grounded security definitions for these cryptographic primitives under different attack scenarios.

[*] The authors were partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901).

Predicate encryption with public index (PE) (also called payload hiding PE) is a relatively new but already established and well-studied primitive which can be used to realize fine-grained access control to data by cryptographic encryption. In a PE scheme for predicate R the data are encrypted under ciphertext indices cInd, which specify access requirements and are not confidential. The users hold secret keys with key indices kInd, which represents their access rights. A user with a secret key for kInd can reconstruct the message, encrypted under cInd, if the predicate is satisfied by the indices, that is if $\mathrm{R}\left(\mathrm{kInd}, \mathrm{cInd}\right) = 1$. Predicate key-encapsulation mechanism (P-KEM) can be seen as specialized PE which can only be used to encrypt a random bit string. Therefore P-KEMs are applied in hybrid constructions together with symmetric-key encryption schemes.

In contrast to conventional PKE, in PE schemes all user secret keys are generated from a single master secret key by a trusted authority. The authority is responsible for key generation; hence, it manages access to the encrypted data. The study of PE actually started when A. Shamir introduced the idea of identity-based encryption (IBE) [21], a predicate encryption for the equality predicate. The first fully-functional IBE was presented in [9] whereas the study of PE for more involved predicates started with [20]. Furthermore, the more general concept of functional encryption (FE) was introduced in [11].

If we look at the security models previously used in the context of PE, we recognize that they originate from the security models for IBE (cf. [20]), which in turn go back to the security models for PKE (cf. [9]). An intuitive security requirement for encryption schemes is that the ciphertext should not reveal any information about the encrypted message to anyone who is not allowed to get access to the message. Semantic security (SS) is an intuitive simulation-based formalization of this requirement whereas indistinguishability (IND) of encryptions is an alternative definition. One evidence of a *good* security definition is its robustness under slightly adapted variations and under reasonable attack scenarios. In fact, in the case of conventional PKE a lot of different extensions and variants of semantic security definitions have been proved to be equivalent (cf. [12]). In particular, the involved definition of what we call adaptive multiple-challenge SS was proved to be equivalent to the simple single challenge SS-definition, which in turn is equivalent to the IND-definition. These equivalences hold under (passive) chosen-plaintext attack (CPA) and (active) chosen-ciphertext attack (CCA). Amongst other results this gave strong evidence that the SS-definition is the right formalization of the required security properties for PKE. On the other hand, the equivalent IND-definition is more easier to handle and hence, it is widely used in security proofs for PKE schemes.

Starting with [9] the indistinguishability definition for PKE was adopted and used for IBE [22, 13, 14, 23, 16] and for more sophisticated PE [20, 3, 18, 24, 25, 17, 1, 7]. Consequently, in [2] security models for IBE were studied under different attack scenarios. One of the results of this work was an equivalence proof of SS-definitions and IND-definitions. Similar to the PKE this result holds under CPA and under CCA even in the adaptive multiple-challenge scenario. Contrary, for the more general context of functional encryption the IND-definition was proved not to be suitable (cf. [11, 19]). Surprisingly and seemingly contradicting to the previous results, in [11] the authors also proved that semantic security can not be achieved even for IBE. This impossibility result was identified in [6, 4] as a consequence of the so-called key-revealing selective-opening attacks (SOA-Ks) which were implicitly covered by the SS-definitions of [19, 11], but were not considered in [2]. The analysis of [19, 11, 6, 4] was restricted to the CPA attack scenario.

At first, it looks like the semantic security and the indistinguishability of encryptions for PE are well studied and well understood. On the one hand, the results for FE under CPA can be applied to PE and the results for IBE cover CCA-security and can be extended to PE. On the other hand, in [5] the authors already mentioned that the identified issues in the definitions for PKE also appear in the definitions for IBE. Furthermore, in [7] the authors used a more sophisticated indistinguishability definition for PE and justified this by specific properties of PE. These intricacies motivated us to reconsider security definitions in the context of predicate-based schemes.

1.1 Main Contribution

We now explain our main contribution in more detail.

How to handle user secret keys? Whereas in the context of conventional PKE there is only a single secret key in question, in PE schemes there are many user secret keys generated from the master secret key. Actually, several users may hold (different) keys for the same key index. Already security definitions for

IBE [9] explicitly prevent user collusion and formalize this by an additional key generation oracle. But the IBE schemes in [9] have a very special property. Namely, there is a unique user secret key for every identity and as a result, the existence of different user secret keys was not considered, neither under CPA nor under CCA. In PE key indices and more complex and specify the access rights of the user. We identify three different formalizations for PE regarding the user secret keys in the literature and name these as follows:

`One-Key model` (OK-model)
`One-Use model` (OU-model)
`Covered-Key model` (CK-model)

In the first two models the adversary (denoted by $\mathcal{A}$) has access to oracles $\mathbf{KGen}(\cdot)$ and $\mathbf{Dec}(\cdot,\cdot)$ under CCA. For the first oracle $\mathcal{A}$ specifies a key index and receives a secret key for this key index. For the second oracle $\mathcal{A}$ specifies a ciphertext as well as a key index, and the ciphertext will be decrypted using a secret key for the specified key index. The OK-model and the OU-model differ in the handling of the user secret keys in these oracles. In the OK-model a unique secret key for kInd is generated and stored if this index is submitted by $\mathcal{A}$ for the first time. This user secret key is used to answer all oracle queries related to kInd. In particular, oracle query $\mathbf{KGen}(\mathrm{kInd})$ always results in the same key. In turn, in the OU-model the challenger generates a new secret key for every key generation query and for every decryption query. Hence, every generated user secret key is used only once. Under CCA the OK-model has previously been used e.g. in [9, 13, 24, 25] and the OU-model has previously been used e.g. in [10, 14]. The CK-model has previously been used in [7]. In this model the user secret keys are generated and numbered in the co-called covered key generation oracle $\mathbf{CKGen}\,(\cdot)$. The adversary can ask to reveal the generated keys, which realizes the functionality of the original key generation oracle but now the adversary can also make more specific decryption queries. For such a query $\mathcal{A}$ specifies the number of the secret key which has to be used for decryption. This models the actual situation in the reality, where different keys even for the same key index are held and used by different users and an adversary might be able to realize chosen-ciphertext attack on these users and their secret keys (cf. [7]).

The three identified models have previously been used and most researcher seem to think they refer to the same security notion. In [8] the authors shortly discuss the OK-model and the OU-model for the case that the key generation algorithm in (H)IBE is probabilistic and state that "*The resulting security definitions [...] seem incomparable, and there does not appear to be a reason to prefer one over the other.*" As a consequence, the authors assume that the key generation algorithm is deterministic, which, as discussed above, is a plausible assumption in the context of IBE. However, the key generation algorithm in PE for sophisticated predicates is always probabilistic and it is a usual case that several keys for the same key index are held by different users. We prove for predicate-based schemes that under CPA the OK-model is weaker than the other two models, whereas CK-model and OU-model are equivalent. Under CCA we show that all three resulting security notions are different and that the security notion achieved from CK-model is the strongest one. Both results hold for PE as well as for P-KEM. Hence, we examine the CK-security of known PE schemes which were proved to satisfy the weaker security notions under CCA. Obviously, the CCA-secure scheme from [9] with unique secret keys is CK-secure. For other schemes this is not obvious and we have to look at their security proofs in detail. Interestingly, for many known schemes we can argue that they also satisfy CK-security, but the arguments differ for every single scheme. For some schemes the question regarding CK-security remains open.

When and how should a challenge decryption be disallowed? We consider this question in the context of PE and P-KEM following the results in [5] for PKE and KEM. While it is not surprising that we can prove similar results for PE, the situation is different when key-encapsulation mechanisms are considered. Namely, in the context of conventional KEM six different security notions were identified and proved to be equivalent in [5]. First of all, we consider two additional security notions (due to the additional key generation oracle) and prove that four of the eight security definitions are too weak in general. The other four notions are in fact equivalent, but some reductions between these notions are not tight. We deduce that in the context of P-KEM we can disallow the decapsulation query on the challenge encapsulation only in the second query phase and can, equivalently, model this restriction both in the penalty-style (SP) and in the exclusion-style (SE). In contrast to this result, the first query phase can be completely dropped for conventional KEM [5].

Reasonable restrictions of adversaries in SS*-definitions and* IND*-definitions.* We present an appropriate semantic security definition for PE and appropriate indistinguishability definitions for PE and P-KEM. On the one hand, the analysis of our above mentioned results is based on these definitions. On the other hand, we justify additional restrictions of adversaries previously used in all security definitions for PE. Namely, for meaningful security notion we have to require that adversaries do not query a user secret key neither in the first query phase nor in the second query phase if this key can be used to decrypt the challenge ciphertext. To the best of our knowledge in all previously used security definitions for predicate-based schemes this restriction was modeled in exclusion-style, that is, adversaries which violate this restriction are not even considered. But in security definitions for PE and P-KEM the challenge ciphertext index is not specified in the first query phase and one can not decide if a key index, submitted in this phase by adversary, matches the challenge ciphertext index. As observed in [5] for no-challenge-decryption condition, exclusion-style adversarial restrictions regarding the first query phase are counterintuitive and may have unpredictable consequences. We show that with slightly modified security definitions these issues do not arise with restrictions on key queries in the first phase.

Furthermore, we prove the equivalence of SS-definition and IND-definition for PE under different attack scenarios including chosen-ciphertext attack, which was not explicitly covered by previous results for IBE and FE. Finally, based on our SS-definition for PE we discuss the impossibility results known from the context of functional encryption.

Conclusion. Based on our results we first of all conclude that the simpler indistinguishability definition for PE is appropriate for both CPA and CCA attack scenarios as long as key-revealing selective-opening attacks are not concerned. Under chosen-ciphertext attack we suggest to use the CK-model to handle user secret keys while under chosen-plaintext attack simpler OU-model is appropriate. This suggestion hold for PE as well as for P-KEM. Finally, under CCA the SE-model is the most advisable in order to handle the no-challenge-decryption condition for both PE and P-KEM.

Organization. In Section 2 we present the preliminaries including syntactical definitions of PE and P-KEM. Section 3 contains SS-definition and IND-definition for PE, the security definition for P-KEM, as well as their analysis and the formal treatment of adversarial restrictions. Based on the presented definitions we look at different formalizations regarding handling of user secret keys in Section 4. Finally, in Section 5 we consider security notions originating from restrictions of adversaries to query the decryption of the challenge ciphertext.

2 Preliminaries

2.1 Notation

We denote by $\alpha := a$ the assignment of the value a to the variable α. We also use the operator $:=$ to define new objects. Let X be a random variable on a finite set S. We denote by $[X]$ the support of X. This notation can be extended to ppt algorithms, since every ppt algorithm $\mathcal{A}$ on input x defines a finite output probability space which we denote by $\mathcal{A}(x)$. That is, $[A(x)]$ denotes all possible outcomes of A on input x. In turn, we write $\alpha \leftarrow X$ to denote sampling of an element from S according to the distribution defined by X ($y \leftarrow A(x)$ for ppt algorithms). We also write $\alpha \leftarrow S$ when sampling an element from S according to the uniform distribution.

We apply the following general conventions by description of algorithms: if we write that an algorithms takes $x \in S$ as input, the output of the algorithm must be the error symbol $\perp$ if $x \notin S$. In particular, there must be an efficient membership test for the set S. Contrarily, if we write that an algorithm takes element x as input, x is just the identifier of the corresponding input and the algorithm itself does not make any demands on x.

2.2 Predicate Encryption With Public Index

In this subsection we shortly recall the formal definitions of predicate families and predicate encryption schemes with public index following [1, 7]. We notice that in this work we often call these schemes just predicate encryptions.

Let Ω, Σ be arbitrary sets. A *predicate family* $\mathcal{R}_{\Omega,\Sigma}$ is a set of binary relations $\mathcal{R}_{\Omega,\Sigma} = \left\{\mathrm{R}_\kappa : \mathbb{X}_\kappa \times \mathbb{Y}_\kappa \to \{0,1\}\right\}_{\kappa \in \Omega \times \Sigma}$, where $\mathbb{X}_\kappa$ and $\mathbb{Y}_\kappa$ are sets called the *key index space* and the *ciphertext index space* of R_κ, respectively. The following conditions must hold:

- *Membership test for* $\mathbb{X}_\kappa$: There exists a polynomial-time algorithm which on input $(\kappa, \mathrm{kInd}) \in (\Omega \times \Sigma) \times \{0,1\}^*$ returns one if and only if $\mathrm{kInd} \in \mathbb{X}_\kappa$.
- *Membership test for* $\mathbb{Y}_\kappa$: There exists a polynomial-time algorithm which on input $(\kappa, \mathrm{cInd}) \in (\Omega \times \Sigma) \times \{0,1\}^*$ returns one if and only if $\mathrm{cInd} \in \mathbb{Y}_\kappa$.
- *Easy to evaluate* : There exists a polynomial-time algorithm which on input $(\kappa, \mathrm{kInd}, \mathrm{cInd}) \in (\Omega \times \Sigma) \times \mathbb{X}_\kappa \times \mathbb{Y}_\kappa$ returns $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd})$.

For example, in key-policy ABE $\mathbb{X}_\kappa$ is a set of Boolean formulas ϕ over variables $x_1, \ldots, x_n$, set $\mathbb{Y}_\kappa$ is the power set of the x_i's, and $\mathrm{R}_\kappa(\phi, \gamma) = 1 \Leftrightarrow \phi(\gamma) = 1$ ($x_i \in \gamma$ for $\gamma \in \mathbb{Y}_\kappa$ means $x_i = 1$). Let $\kappa = (\mathrm{des}, \sigma)$. Indices $\mathrm{des} \in \Omega$ specify some general description properties of the corresponding predicates (e.g. the size of γ in the example above might be restricted), and indices $\sigma \in \Sigma$ specify domain of computation which depends on the security parameter (e.g. $\mathbb{Z}_p$ for prime p). In the following definition we use some additional conventions, which will be subsequently explained.

Definition 2.1. *A* ***predicate encryption with public index*** Π *for predicate family* $\mathcal{R}_{\Omega,\Sigma}$ *and message space* $\mathcal{M} = \{0,1\}^*$ *consists of four ppt algorithms:*

Setup $\left(1^\lambda, \mathrm{des}\right) \to (\mathrm{msk}, \mathrm{pp}_\kappa)$: *takes as input security parameter* λ*, and description parameter* des. *It outputs a master secret key and public parameters. The algorithm determines among other elements* $\sigma \in \Sigma$ *and the relation index* $\kappa = (\mathrm{des}, \sigma) \in \Omega \times \Sigma$ *is (implicitly) included in* pp_κ*.*

KeyGen $\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{msk}, \mathrm{kInd}\right) \to \mathrm{sk}$: *takes as input public parameters for relation index* κ*, the master secret key* msk*, and a key index* $\mathrm{kInd} \in \mathbb{X}_\kappa$*. It generates a user secret key* sk *for* kInd.

We denote by $\mathbb{UK}_{\mathrm{kInd}} \supseteq [\mathrm{KeyGen}(\mathrm{pp}_\kappa, \mathrm{msk}, \mathrm{kInd})]$ *a set of syntactically correct secret keys for* kInd *with efficient membership test (given* pp_κ*). Furthermore, we define* $\mathbb{UK}_{\mathrm{pp}_\kappa} := \biguplus_{\mathrm{kInd} \in \mathbb{X}_\kappa} \mathbb{UK}_{\mathrm{kInd}}$.

Enc $\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{cInd}, m\right) \to \mathrm{CT}$: *takes as input public parameters for relation index* κ*, a ciphertext index* $\mathrm{cInd} \in \mathbb{Y}_\kappa$*, and a message* $m \in \mathcal{M}$*. It outputs a ciphertext* CT *of* m *under* cInd.

We denote by $\mathbb{C}_{\mathrm{cInd}} \supseteq \bigcup_{m \in \mathcal{M}} [\mathrm{Enc}(\mathrm{pp}_\kappa, \mathrm{cInd}, m)]$ *a set of syntactically correct encryptions under* cInd *with efficient membership test (given* pp_κ*). Furthermore, we define* $\mathbb{C}_{\mathrm{pp}_\kappa} := \biguplus_{\mathrm{cInd} \in \mathbb{Y}_\kappa} \mathbb{C}_{\mathrm{cInd}}$.

Dec $\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{sk}, \mathrm{CT}\right) \to m$: *takes as input public parameters for relation index* κ*, a secret key* $\mathrm{sk} \in \mathbb{UK}_{\mathrm{kInd}} \subseteq \mathbb{UK}_{\mathrm{pp}_\kappa}$ *and a ciphertext* $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}} \subseteq \mathbb{C}_{\mathrm{pp}_\kappa}$*. It outputs a message* $m \in \mathcal{M}$ *or the error symbol* $\perp \notin \mathcal{M}$*. If* $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}) = 0$*, the output is* $\perp$*.*

Correctness: *For every security parameter* λ*, every* $\mathrm{des} \in \Omega$*, every* $(\mathrm{msk}, \mathrm{pp}_\kappa) \in \left[\mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)\right]$*, every* $m \in \mathcal{M}$*, every* $\mathrm{kInd} \in \mathbb{X}_\kappa$ *and* $\mathrm{cInd} \in \mathbb{Y}_\kappa$ *which satisfy* $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}) = 1$ *it must hold*

$$\Pr\left[\mathrm{Dec}\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{KeyGen}\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{msk}, \mathrm{kInd}\right), \mathrm{Enc}\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{cInd}, m\right)\right) = m\right] = 1 \ .$$

An important property of PE with public index is that the ciphertext index can be efficiently computed from every syntactically correct ciphertext. The same can be assumed for user secret keys and their indices. Hence, we introduce the set $\mathbb{UK}_{\mathrm{pp}_\kappa}$ and the set $\mathbb{C}_{\mathrm{pp}_\kappa}$ in order to make implicit syntactical checks in the algorithms and in the security experiments explicit. If this kind of checks is required by the scheme, the sets have to be defined appropriately. This prevents implementation errors and inaccuracy of definitions. The requirement $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}} \subseteq \mathbb{C}_{\mathrm{pp}_\kappa}$ should be read as follows: the (public) ciphertext index of CT satisfies $\mathrm{cInd} \in \mathbb{Y}_\kappa$ and CT is an element of $\mathbb{C}_{\mathrm{cInd}}$. Similar notation is used for user secret keys.

We assume for convenience that public parameters pp_κ have length at least λ, and that λ and $\kappa \in \Omega \times \Sigma$ can be efficiently determined from pp_κ. Hence, we avoid to write 1^λ as input of the algorithms except for the setup algorithm. Furthermore, if the public parameters pp_κ are fixed and obvious from the context, we also avoid to write pp_κ as input of the algorithms.

2.3 Predicate Key Encapsulation Mechanism With Public Index

In practice, PE schemes are usually used to encrypt a symmetric secret key, which then is used to encrypt the actual message. Special PE schemes constructed extra for this application are also called predicate key-encapsulation mechanism. We use the same notational conventions as for PE and hence, simplify the description.

Definition 2.2. *A **predicate key encapsulation mechanism with public index** Π for predicate family $\mathcal{R}_{\Omega,\Sigma}$ and family of key spaces $\mathcal{K} = \{\mathbb{K}_\lambda\}_{\lambda\in\mathbb{N}}$ consists of four ppt algorithms:*

Setup $\left(1^\lambda, \mathrm{des}\right) \to (\mathrm{msk}, \mathrm{pp}_\kappa)$: *takes as input security parameter λ and description index* des. *It outputs a master secret key and public parameters.*

KeyGen $\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{msk}, \mathrm{kInd}\right) \to \mathrm{sk}$: *takes as input public parameters for relation index κ, a master secret key* msk, *and a key index* $\mathrm{kInd} \in \mathbb{X}_\kappa$. *It generates a user secret key* sk *for* kInd.

Encaps $\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{cInd}\right) \to (\mathrm{K}, \mathrm{CT})$: *takes as input public parameters for relation index κ and a ciphertext index* $\mathrm{cInd} \in \mathbb{Y}_\kappa$. *It outputs a key* $\mathrm{K} \in \mathbb{K}_\lambda$ *and an encapsulation* CT *of this key under* cInd.

Decaps $\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{sk}, \mathrm{CT}\right) \to \mathrm{K}$: *takes as input public parameters for relation index κ, a secret key* $\mathrm{sk} \in \mathbb{UK}_{\mathrm{kInd}} \subseteq \mathbb{UK}_{\mathrm{pp}_\kappa}$ *and an encapsulation* $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}} \subseteq \mathbb{C}_{\mathrm{pp}_\kappa}$. *It outputs a key* $\mathrm{K} \in \mathbb{K}_\lambda$ *or the error symbol* $\perp \notin \mathbb{K}_\lambda$. *If* $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}) = 0$, *the output is* $\perp$.

Correctness: *For every security parameter λ, every* $\mathrm{des} \in \Omega$, *every* $(\mathrm{msk}, \mathrm{pp}_\kappa) \in \left[\mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)\right]$, *every* $m \in \mathcal{M}$, *every* $\mathrm{kInd} \in \mathbb{X}_\kappa$ *and* $\mathrm{cInd} \in \mathbb{Y}_\kappa$ *which satisfy* $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}) = 1$, *and every* $(\mathrm{K}, \mathrm{CT}) \in \left[\mathrm{Encaps}\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{cInd}\right)\right]$ *it must hold*

$$\mathrm{Decaps}\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{KeyGen}\left(1^\lambda, \mathrm{pp}_\kappa, \mathrm{msk}, \mathrm{kInd}\right) \mathrm{CT}\right) = \mathrm{K} \ .$$

We furthermore define the smoothness for P-KEM, which is similar to the definition of smoothness for conventional KEMs (cf. [5]).

Definition 2.3. *Let* Π *be a* P-KEM *with public index for predicate family* $\mathcal{R}_{\Omega,\Sigma}$. *Furthermore, let* $\mathrm{des} \in \Omega$ *and* $\lambda \in \mathbb{N}$ *be arbitrary. Define*

$$\mathrm{Smth}_\Pi(\lambda, \mathrm{des}) := \mathbf{E}\left[\max_{\substack{\mathrm{cInd}\in\mathbb{Y}_\kappa,\\ \mathrm{CT}\in\{0,1\}^*}} \left(\Pr_{(\mathrm{K},\mathrm{CT}')\leftarrow\mathrm{Encaps}(1^\lambda,\mathrm{pp}_\kappa,\mathrm{cInd})}\left[\mathrm{CT}' = \mathrm{CT}\right]\right)\right] \ ,$$

where the expected value is taken over $(\mathrm{msk}, \mathrm{pp}_\kappa) \leftarrow \mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)$. Π *is called **smooth** if for every* $\mathrm{des} \in \Omega$ *the function* $\mathrm{Smth}_\Pi(\cdot, \mathrm{des})$ *is negligible.*

3 Semantic Security and Indistinguishability for PE

In this section we first define two formal security definitions or rather definitional templates for PE: a semantic security (SS) template and an indistinguishability (IND) template. We explicitly notice that the definitions are not novel, but as already mentioned in the introduction there exist important subtleties in these definitions in context of PE which must be considered. The templates prescind from some details and prepare the formal treatment of adversarial behavior and some further definitional aspects considered in the following sections. Furthermore, they allows us to compare these two well-known security definitions in the context of PE independently of the concrete attack scenario.

3.1 Semantic Security Template

We first define a basic version of semantic security for PE. We begin with formal definition of adversaries with only few pure syntactic restrictions. Let Π be a PE scheme with public index for a predicate family $\mathcal{R}_{\Omega,\Sigma}$. Formally, a semantic security adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against Π is a pair of algorithms with oracle access such that $\mathcal{A}_1$ given among other things correctly generated public parameters pp_κ outputs a triple $(\mathrm{cInd}^*, \tau, St)$. The first element is referred to as the **challenge ciphertext index** and has to satisfy $\mathrm{cInd}^* \in \mathbb{Y}_\kappa$; τ is referred to as the **challenge template** and itself contains a triple of circuits $\tau = (\hat{\mathcal{M}}, h, f)$ such that the number of input bits of h and f is equal to the number of output bits of $\hat{\mathcal{M}}$; the last element St is referred to as the **state information** and there are no demands on it. For technical reasons, explained below, we allow $\mathcal{A}_1$ to output an error symbol $\perp$. The set of SS-adversaries against Π is denoted by $\mathbf{A}^{\mathrm{SS}}_\Pi$ or just $\mathbf{A}^{\mathrm{SS}}$ if Π is obvious from the context.

The definition of semantic security is a simulation-based definition and hence, we require two probability experiments presented in Figure 1. The real experiment on the left side is parametrized by attack scenario ATK and by adversary $\mathcal{A} \in \mathbf{A}^{\mathrm{SS}}$, whereas in the simulation experiment on the right side algorithm

```
$\textbf{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$ : $(\mathrm{msk}, \mathrm{pp}_\kappa) \leftarrow \mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)$;
$\left(\mathrm{cInd}^*, \left(\hat{\mathcal{M}}, h, f\right), St\right) \leftarrow \mathcal{A}_1^{\mathbf{O}_1^{\mathrm{ATK}}(\cdot)}\left(1^\lambda, \mathrm{pp}_\kappa\right)$;
Output 0 if $\mathcal{A}_1$ outputs $\perp$; $\hat{m} \leftarrow \hat{\mathcal{M}}\left(\mathcal{U}_{\mathrm{poly}(\lambda)}\right)$;
$\mathrm{CT}^* \leftarrow \mathrm{Enc}\left(\mathrm{cInd}^*, \hat{m}\right)$; $\nu \leftarrow \mathcal{A}_2^{\mathbf{O}_2^{\mathrm{ATK}}(\cdot)}\left(\mathrm{CT}^*, h\left(\hat{m}\right), St\right)$;
Output    $\nu = f\left(\hat{m}\right) \quad \wedge \quad \overline{\mathrm{BadQuery}}$;
```

```
$\textbf{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \mathrm{des})$ :
$\left(\left(\hat{\mathcal{M}}, h, f\right), St\right) \leftarrow \mathcal{A}'_1\left(1^\lambda, \mathrm{des}\right)$;
Output 0 if $\mathcal{A}'_1$ outputs $\perp$;
$\hat{m} \leftarrow \hat{\mathcal{M}}\left(\mathcal{U}_{\mathrm{poly}(\lambda)}\right)$; $\nu \leftarrow \mathcal{A}'_2\left(h\left(\hat{m}\right), St\right)$;
Output    $\nu = f\left(\hat{m}\right)$;
```

Fig. 1. Semantic security experiments for PE.

$\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ is (for a moment) arbitrary. In the real experiment, we use oracles $\mathbf{O}_1(\cdot)$, $\mathbf{O}_2(\cdot)$ in order to model additional powers of $\mathcal{A}$ in the first and in the second query phase. Concrete specifications of these oracles depend on attack scenario ATK and will be considered later in detail. The probability event BadQuery will be used to define the restrictions on the oracle queries of $\mathcal{A}$ formally. Think for a moment about the usual restriction in context of PE that $\mathcal{A}$ is not allowed to query a user secret key if this key can be used to decrypt the *challenge ciphertext* CT^*. In this section we only notice that in all considered attack scenarios the event BadQuery can be efficiently recognized *at the end* of the experiment and the oracles can be efficiently realized using the public parameters and the master secret key.

By the definition, the challenge message $\hat{m}$ is chosen according to the probability distribution implicitly specified by $\hat{\mathcal{M}}$. The notation $\hat{\mathcal{M}}\left(\mathcal{U}_{\mathrm{poly}(\lambda)}\right)$ means that polynomially many input bits of $\hat{\mathcal{M}}$ are chosen uniformly at random. Algorithms $\mathcal{A}_2$ and $\mathcal{A}'_2$ are given the value $h(\hat{m})$. The real adversary $\mathcal{A}_2$ is also given an encryption of $\hat{m}$ which is called the **challenge ciphertext**. The output of the experiments is equal one if $\mathcal{A}_2$ respectively $\mathcal{A}'_2$ correctly predict $f(\hat{m})$.

We call an algorithm $\mathcal{A}'$ a simulator for $\mathcal{A}$ with respect to $\mathrm{des} \in \Omega$ if for every possible challenge template $\hat{\tau}$ outputted by $\mathcal{A}_1$ (incl. $\perp$) it holds

$$\Pr\left[\mathcal{A}'_1 \text{ outputs } \hat{\tau}\right] \geq \Pr\left[\mathcal{A}_1 \text{ outputs } \hat{\tau} \wedge \overline{\mathrm{BadQuery}}\right] \quad .$$

The advantage $\text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}(\lambda, \mathrm{des})$ of $\mathcal{A}$ against $\mathcal{A}'$ under attack scenario ATK is defined by

$$\Pr\left[\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) = 1\right] - \Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \mathrm{des}) = 1\right] \quad .$$

We say that the advantage of an adversary $\mathcal{A} \in \mathbf{A}^{\mathrm{SS}}_{\Pi}$ under attack scenario ATK is negligible if for all $\mathrm{des} \in \Omega$ there exists a ppt simulator $\mathcal{A}'$ of $\mathcal{A}$ with respect to des such that the advantage $\text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}(\lambda, \mathrm{des})$ of $\mathcal{A}$ against $\mathcal{A}'$ under attack scenario ATK is negligible. Finally, semantic security is defined as follows.

Definition 3.1. *A predicate encryption scheme with public index* Π *is called **semantically secure** under attack scenario* ATK *(or* SS-ATK*-secure) if every ppt adversary* $\mathcal{A} \in \mathbf{A}^{\mathrm{SS}}_{\Pi}$ *has negligible advantage under attack scenario* ATK.

Intuitively, semantic security states that a ppt adversary $\mathcal{A}$ cannot learn anything about the message $\hat{m}$ from its encryption CT^* except for negligible probability. Formally, this is proved by providing a simulator $\mathcal{A}'$, which can perform as good as $\mathcal{A}$ but is not given the challenge ciphertext CT^*. Compared to the real adversary, simulator $\mathcal{A}'_1$ gets $\mathrm{des} \in \Omega$ instead of pp_κ as input. This is due to the fact that all inputs of $\mathcal{A}'_2$ are independent of the concrete public parameters. We explicitly mention that in the more general context of functional encryptions the authors of [4] revealed issues regarding the possibility of the simulators to generate the public parameters for themselves. But they also proved that *all or nothing* schemes (including PE), where the user secret keys only allow to reconstruct the original message rather than some function on this message, are not affected.

Our definition differs from the SS-definitions for PKEs [12], IBE [2], and FE [19, 11, 6, 4] in two ways. First of all, we make the restrictions of adversaries *explicit* by presenting the BadQuery event and a penalty for adversary if it violates the restrictions of the experiment. This kind of formalization goes back to [5], where the authors showed that not carefully reasoned assumptions about adversarial behavior may lead to weaknesses in security definitions. Due to the explicit penalty through the BadQuery event we slightly modified the definition of the simulators. We notice that a simulator of $\mathcal{A}$ is allowed to output an arbitrary challenge template in the case that $\mathcal{A}$ violates the restrictions of the experiment. In turn, the challenge templates must be identically distributed in the experiments $\text{SS-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$ and $\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \text{des})$ if $\mathcal{A}$ never causes the BadQuery event (that is, $\mathcal{A}$ does not violate the restrictions of the experiment). This is the usual condition in SS-definitions in order to relate the computation of simulator $\mathcal{A}'$ to the computation of adversary $\mathcal{A}$.

The second difference is due to the presented possibility of adversary to abort after the first query phase, which is already the result of the formal treatment of adversarial restrictions. As already mentioned in the introduction, in all known security definitions for PE one does not even consider adversaries which query user secret keys in *both* query phases which can be used to decrypt the challenge ciphertext. That is, adversaries which violate this restriction are not considered at all. We identified a problem with this modeling for general predicate families. Namely, in order to justify this assumption for the first query phase, one has to show that given a polynomially large set of (corrupted) key indices chosen by $\mathcal{A}$, a ciphertext index cInd^* which does not match all these key indices can be efficiently find. For sophisticated predicates this can be a hard problem. For example in key-policy attribute-based schemes, in order to find a set of attributes, which can be used as the challenge ciphertext index, one has to find an assignment which does not satisfy the given Boolean formulas (corrupted key indices). In order to deal with this problem we allow the adversary to output an error symbol after the first query phase which is treated as an early guess and is not penalized. This modeling enables us to keep the mentioned restriction in the first query phase. The introduced possibility to abort does not influence the security guaranties, since $\mathcal{A}$ has to abort before she gets the challenge and in the case that $\mathcal{A}$ violates the restrictions she is penalized even if she outputs $\perp$. This ensures that the possibility to output $\perp$ does not contribute to the advantage of $\mathcal{A}$. On the one hand, our extension simplifies the description of our constructed adversaries in the cases where $\mathcal{A}_1$ has to abort. On the other hand, we have to take care of this output during the simulation of adversaries and in the formal security analysis.

3.2 Indistinguishability Templates for PE and P-KEM

Next, we define indistinguishability adversaries and an indistinguishability template in the usual single-challenge form for PE as well as for P-KEM. Also here we abstract from concrete attack scenarios and make the restrictions of adversaries explicit through BadQuery events.

Let Π be a PE scheme for predicate family $\mathcal{R}_{\Omega,\Sigma}$. An indistinguishability adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against Π is a pair of algorithms with oracle access such that $\mathcal{A}_1$ given correctly generated public parameters pp_κ outputs the error symbol $\perp$ or a tuple $(\text{cInd}^*, m_0, m_1, St)$ satisfying $\text{cInd}^* \in \mathbb{Y}_\kappa$, $m_0, m_1 \in \mathcal{M}$ and $|m_0| = |m_1|$, whereas $\mathcal{A}_2$ always outputs a bit. The set of IND-adversaries against Π is denoted by $\mathbf{A}^{\text{IND}}_{\Pi}$ or just by $\mathbf{A}^{\text{IND}}$ if Π is obvious from the context. The set $\mathbf{A}_{\Pi,\text{P-KEM}}$ of adversaries against P-KEM scheme Π is defined similarly except for the outputs of $\mathcal{A}_1$, which does not contain the messages. The indistinguishability experiments are presented in Figure 2, where in the case of P-KEM the family of key spaces of Π is denoted by $\mathcal{K} = \{\mathbb{K}_\lambda\}_{\lambda\in\mathbb{N}}$.

```
$\textbf{IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$ :
$b \leftarrow \{0,1\}$ ; $(\text{msk}, \text{pp}_\kappa) \leftarrow \text{Setup}(1^\lambda, \text{des})$ ;
$(\text{cInd}^*, m_0, m_1, St) \leftarrow \mathcal{A}_1^{\mathbf{O}_1^{\text{ATK}}(\cdot)}(1^\lambda, \text{pp}_\kappa)$ ;
Output  $b \wedge \overline{\text{BadQuery}}$ if $\mathcal{A}_1$ outputs $\perp$;
$\text{CT}^* \leftarrow \text{Enc}(\text{pp}_\kappa, \text{cInd}^*, m_b)$ ;
$b' \leftarrow \mathcal{A}_2^{\mathbf{O}_2^{\text{ATK}}(\cdot)}(\text{CT}^*, St)$ ;
Output  $b = b' \wedge \overline{\text{BadQuery}}$;
```

```
$\textbf{P-KEM}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$ :
$b \leftarrow \{0,1\}$ ; $\text{K}_1 \leftarrow \mathbb{K}_\lambda$; $(\text{msk}, \text{pp}_\kappa) \leftarrow \text{Setup}(1^\lambda, \text{des})$ ;
$(\text{cInd}^*, St) \leftarrow \mathcal{A}_1^{\mathbf{O}_1^{\text{ATK}}(\cdot)}(1^\lambda, \text{pp}_\kappa)$ ;
Output  $b \wedge \overline{\text{BadQuery}}$ if $\mathcal{A}_1$ outputs $\perp$;
$(\text{K}_0, \text{CT}^*) \leftarrow \text{Encaps}(\text{pp}_\kappa, \text{cInd}^*)$ ; $\text{K}^* := \text{K}_b$;
$b' \leftarrow \mathcal{A}_2^{\mathbf{O}_2^{\text{ATK}}(\cdot)}(\text{K}^*, \text{CT}^*, St)$ ;
Output  $b = b' \wedge \overline{\text{BadQuery}}$.
```

Fig. 2. Indistinguishability experiments for PE and P-KEM.

Definition 3.2. *A* PE *scheme with public index* Π *for predicate family* $\mathcal{R}_{\Omega,\Sigma}$ *has* ***indistinguishable encryptions*** *under attack* ATK *(or* IND-ATK*-secure) if for every* $\mathrm{des} \in \Omega$ *and every ppt adversary* $\mathcal{A} \in \mathbf{A}^{\mathrm{IND}}$ *the advantage*

$$\text{Adv-IND-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) := 2 \cdot \Pr\left[\text{IND-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) = 1\right] - 1$$

is negligible.

The security definition for P-KEM is similar. We notice that by our definition $\mathcal{A}$ cannot increase its advantage using error symbol $\perp$ but this behavior is also not penalized. Indeed, as long as $\mathcal{A}$ does not cause the event BadQuery, the output of the experiment will be one with probability $1/2$ in the case that $\mathcal{A}$ outputs $\perp$. Using this modeling approach we deal with the fact that $\mathcal{A}$ must not find a valid challenge ciphertext index in order to avoid the BadQuery event. A side effect of our definition is that $\mathcal{A}_1$ can output *a guess* (in form of $\perp$) which often simplifies the description of adversaries.

3.3 Attack Scenarios and Additional Restrictions of Adversaries

Next, we specify to a certain extent the BadQuery event and the oracles for $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$. We use the syntax of P-KEM, since during the further analysis this primitive is mainly used. All results in this subsection can be translated to PE for both SS and IND. We formally show which assumptions about the behavior of adversaries can be made without loss of generality. Even though most of these assumptions are folklore, the work of [5] showed that even in the context of PKE the real effects of certain assumptions about adversarial behavior have sometimes been underestimated or even misunderstood.

Compared to PKE setting, in the case of predicate-based schemes one has to prevent so-called collusion attacks. Intuitively, this means that a set of users should not be able to combine their secret keys in order to decrypt a ciphertext, which none of the users can decrypt on its own. In the security experiments for P-KEM respectively PE this is modeled by the key generation oracle **KGen**. This oracle takes as input a key index $\mathrm{kInd} \in \mathbb{X}_\kappa$ and returns a user secret key for kInd. We call $\mathrm{kInd} \in \mathbb{X}_\kappa$ a **corrupted key index** if $\mathcal{A}$ queried **KGen** (kInd). The key generation oracle is the only one available if chosen-plaintext attacks (CPAs) are considered.

Under so-called adaptive chosen-ciphertext attack (CCA2) the adversary against a PE scheme additionally gets access to the decryption oracle **Dec** in both query phases whereas in the a priori chosen-ciphertext attacks (CCA1) this oracle is available only for $\mathcal{A}_1$. The decryption oracle takes as input a ciphertext $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}} \subset \mathbb{C}_{\mathrm{pp}_\kappa}$ and a key index $\mathrm{kInd} \in \mathbb{X}_\kappa$ and returns the decryption of CT under a secret key for kInd. In the context of P-KEM this oracle is called the decapsulation oracle and is denoted by **Decaps**. The result of an oracle query is the error symbol $\perp$ if the inputs to the oracle do not satisfy the syntactical form as described above.

Through the BadQuery event we specify two additional restrictions on $\mathcal{A}$. The first restriction is folklore and states that the adversary is not allowed to corrupt key index kInd if $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}^*) = 1$. It is important to notice that in the first query phase cInd^* is not specified, but at the end of the experiment the BadQuery event can be recognized as required. The second restriction which we consider here is for CCA2 attack scenario and disallows decryption query on CT^* in the *second* query phase.

The following lemma summarizes assumptions about the behavior of $\mathcal{A}$ which can be made w.l.o.g. in the context of P-KEM. The formal proof is presented in Appendix C on page 30.

Lemma 3.1. *Let* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$ *and* Π *be a* P-KEM *scheme with public index for predicate family* $\mathcal{R}_{\Omega,\Sigma}$*.* Π *is* ATK*-secure if and only if every ppt adversary* $\mathcal{A} \in \mathbf{A}_{\Pi,\text{P-KEM}}$ *which satisfy the following conditions has negligible advantage under attack scenario* ATK*. Let* pp_κ *be the public parameter generated during the experiment. The conditions are as follows:*

- *The output of* $\mathcal{A}_1$ *is the error symbol* $\perp$ *or a tuple* $(\mathrm{cInd}^*, \mathrm{St})$ *such that* $\mathrm{cInd}^* \in \mathbb{Y}_\kappa$*. The output of* $\mathcal{A}_2$ *is a bit.*
- *All key indices* kInd *submitted by* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *satisfy* $\mathrm{kInd} \in \mathbb{X}_\kappa$*.*
- *For every* CT *submitted to the decapsulation oracle by* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *it holds* $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}} \subset \mathbb{C}_{\mathrm{pp}_\kappa}$*. Furthermore, the corresponding key index* kInd *satisfies* $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}) = 1$*.*
- *For every corrupted key index* kInd *(in both query phases) it holds* $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}^*) = 0$*.*
- *(Only for* CCA2*)* $\mathcal{A}_2$ *never submits* CT^* *to the decapsulation oracle.*

In the proof, given an arbitrary $\mathcal{A} \in \mathbf{A}_{\text{P-KEM}}$ we construct $\mathcal{A}' \in \mathbf{A}_{\text{P-KEM}}$ which satisfies the conditions in the lemma and achieves advantage of $\mathcal{A}$. The main non-trivial step in the proof is to ensure that the condition regarding key generation queries of $\mathcal{A}'_1$ is satisfied. Similar lemmas can also be proved for our SS-security definition and for our IND-security definition when the first restriction on the form of the output of $\mathcal{A}_1$ is adopted according to the definition of $\mathbf{A}^{\text{SS}}$ and $\mathbf{A}^{\text{IND}}$, respectively.

Remark 3.1. The restrictions on adversarial queries can be extended as long as the adversary can verify the corresponding conditions previous to the query by herself. The last restriction in Lemma 3.1 is a good example for such a restriction whereas a similar restriction for $\mathcal{A}_1$ is not covered, since the challenge ciphertext is not defined in the first query phase.

3.4 Relations Between SS-Security and IND-Security for PE

For identity-based encryption (IBE), a special case of PE, different security notions and attack scenarios were considered in [2]. In turn, for more general functional encryption (FE) SS-definitions and IND-definitions were previously analyzed under CPA in [19, 11, 6, 4]. In this subsection based on the presented templates we discuss the relations between SS-security and IND-security in context of PE. We can prove the following theorem similarly to the case of PKE and IBE. The formal proof is presented in Appendix A.

Theorem 3.1. *Suppose* ATK $\in$ {CPA, CCA1, CCA2} *and* Π *is a predicate encryption scheme with public index. Then,* Π *is* SS-ATK-*secure if and only if it is* IND-ATK-*secure.*

With this theorem one might think that the relation between semantic security and indistinguishability are evident. But, as already mentioned at the beginning of this section, our SS-definition is really basic. Indeed, there are several reasonable and at least syntactically stronger definitions of semantic security. Let us first consider the probability experiments presented in Figure 3.

```
$\textbf{aSS-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$ : $(\text{msk}, \text{pp}_\kappa) \leftarrow \text{Setup}(1^\lambda, \text{des})$;
$\left(\text{cInd}^*, \left(\hat{\mathcal{M}}, h\right), St\right) \leftarrow \mathcal{A}_1^{\mathbf{O}_1^{\text{ATK}}(\cdot)}(1^\lambda, \text{pp}_\kappa)$;
Output 0 if $\mathcal{A}_1$ outputs $\perp$;
$\hat{m} \leftarrow \hat{\mathcal{M}}\left(\mathcal{U}_{\text{poly}(\lambda)}\right)$; $\text{CT}^* \leftarrow \text{Enc}(\text{cInd}^*, \hat{m})$;
$(f, \nu) \leftarrow \mathcal{A}_2^{\mathbf{O}_2^{\text{ATK}}(\cdot)}(\text{CT}^*, h(\hat{m}), St)$;
Output    $\nu = f(\hat{m}) \quad \wedge \quad \overline{\text{BadQuery}}$;
```

```
$\textbf{aSS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \text{des})$ :
$\left(\left(\hat{\mathcal{M}}, h\right), St\right) \leftarrow \mathcal{A}'_1(1^\lambda, \text{des})$;
Output 0 if $\mathcal{A}'_1$ outputs $\perp$;
$\hat{m} \leftarrow \hat{\mathcal{M}}\left(\mathcal{U}_{\text{poly}(\lambda)}\right)$;
$(f, \nu) \leftarrow \mathcal{A}'_2(h(\hat{m}), St)$;
Output    $\nu = f(\hat{m})$;
```

Fig. 3. Adaptive semantic security experiments

In these experiments the function f is specified by adversary only at the end of the experiment. In particular, this function might also depend on CT^* as well as on the computations and on the queries of $\mathcal{A}_2$. It seems like $\mathcal{A}$ has much more power in this experiment. We indeed have to weaken the restriction on the simulator and allow the distributions of the triples $(\hat{\mathcal{M}}, h, f)$ to be indistinguishable in both experiments (cf. Definition B.1 in Appendix B). But then, we can prove the following theorem (see the proof in Appendix B on page 27).

Theorem 3.2. *For every* ATK $\in$ {CPA, CCA1, CCA2} *a predicate encryption scheme is semantically secure under attack scenario* ATK *if and only if it is adaptive semantically secure under attack scenario* ATK.

The adaptive semantic security definition is only a few (non-trivial) steps shy of so-called SS2-security definition for FE presented in [6]. This definition is proved to be equivalent to the indistinguishability definition even in the more general context of functional encryptions. The required extensions toward the SS2-security are similar to the extensions for PKE and IBE and we refer to [12, 2] for extensive study of these extensions.

Based on the results in [19, 11, 6, 4] for functional encryption we deduce that, as long as key-revealing selective-opening attacks (SOA-Ks) are not considered, indistinguishability definition remains the most suitable security definition in the context of PE.

4 Handling of User Secret Keys

Whereas in the context of conventional PKE there is only a single secret key in question, in predicate encryption (PE) schemes there are many user secret keys generated from the same master secret key. Actually, several users may hold (different) keys for the same key index. In the templates from the previous sections we prescinded from details regarding this circumstance. The goal of this section is to consider different possibilities to handle user secret keys in the security experiments. Indeed, we identify three different formalizations regarding the user secret keys in the literature and name these as follows: one-key model (OK-model), one-use model (OU-model), and covered key model (CK-model). The oracles for these models under CCA2 attacks are presented in Table 1. The oracles for CPA and CCA1 are the same with the usual restrictions.

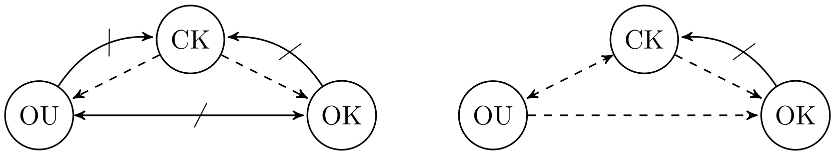
OK	OU	CK

KGen (kInd):
If (kInd, sk) $\in S_k$ return sk;
sk $\leftarrow$ KeyGen (msk, kInd);
S_k.add((kInd, sk));
Return sk;
Decaps (CT, kInd):
sk := **KGen** (kInd);
Return Decaps (sk, CT);
	KGen (kInd):
	sk $\leftarrow$ KeyGen (msk, kInd);
	Return sk;
	Decaps (CT, kInd):
	sk := **KGen** (kInd)
	Return Decaps (sk, CT);
		CKGen (kInd):
		sk $\leftarrow$ KeyGen (msk, kInd);
		i++; S_k.add((i, sk));
		Open (i):
		Return sk from (i, sk) $\in S_k$;
		Decaps (CT, i):
		Return Decaps (sk, CT),
		where (i, sk) $\in S_k$;

S_k contains all keys which have been generated by **KGen** / **CKGen**.

Table 1. Oracle specification for different models under CCA2 attacks.

In the OK-model the challenger generates and stores a unique secret key for kInd if this index is submitted by $\mathcal{A}$ for the first time. This user secret key is used to answer all oracle queries related to kInd. In particular, oracle query **KGen**(kInd) always results in the same key. The OK-model was previously used e.g. in [9, 13, 24, 25]. In the OU-model the challenger generates a new secret key for every query and the generated key will be used only once. This model was previously used e.g. in [10, 14]. In the CK-model the adversary specifies not only the key indices, but also the keys which have to be used to answer the decryption queries, which is formalized using additional covered key generation oracle. The CK-model intuitively reflects the reality, where users hold specific secret keys and use their keys for decryption. Hence, adversaries realizing chosen ciphertext attacks might not only know the access rights of the users (that is, the key indices of their keys), but could also exploit the fact that the same secret key is used several times. In [7] the authors explicitly used the CK-model and adopted the dual system encryption methodology [23] in order to deal with additional power of adversary due to the more specific decryption oracles.



A barred arrow is a separation. A dashed arrow denotes obvious implication.

Fig. 4. Relation between different security models for PE and P-KEM under CCA1 and CCA2 attacks on the left and under CPA attacks on the right.

In this section we prove that under CCA attacks the OK-model and the OU-model are weaker than the CK-model (cf. Figure 4). We notice that using the **CKGen** oracle and the **Open** oracle we can simulate the behavior of every adversary in the other two models. Hence, CK-security obviously implies OK-security and OU-security. Furthermore, under CPA the CK-model and the OU-model are equivalent due to the absence of the decryption oracle. The deduce the weakness of the OK-model under CPA from the corresponding result under CCA. All mentioned results hold for PE as well as for P-KEM. We show the separation results for P-KEM, since the constructions in the proofs are a bit more involved in this context.

For convenience, we define sets $\mathbf{A}^{\mathrm{OK}}_{\text{P-KEM}}$, $\mathbf{A}^{\mathrm{OU}}_{\text{P-KEM}}$, $\mathbf{A}^{\mathrm{CK}}_{\text{P-KEM}}$ of adversaries which are as $\mathbf{A}_{\text{P-KEM}}$, but the adversaries use the oracles as defined in the corresponding models. Obviously, the oracles in all three models can be realized using the master secret key and the public parameters as required in Section 3. Furthermore, $\text{P-KEM}^{\mathrm{ATK,mod}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$ is an instantiation of $\text{P-KEM}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$ with oracles as defined in model mod.

Definition 4.1. *Let* mod $\in \{\mathrm{OK}, \mathrm{OU}, \mathrm{CK}\}$ *and* ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$*. A* P-KEM Π *with public index for predicate family $\mathcal{R}_{\Omega,\Sigma}$ and a family of key spaces $\mathcal{K}$ is called secure in model* mod *under attack* ATK *(also* mod-ATK*-secure) if for every* des $\in \Omega$ *and every ppt adversary $\mathcal{A} \in \mathbf{A}^{\mathrm{mod}}_{\Pi,\text{P-KEM}}$, the advantage*

$$\text{Adv-P-KEM}^{\mathrm{ATK,mod}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) := 2 \cdot \Pr\left[\text{P-KEM}^{\mathrm{ATK,mod}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) = 1\right] - 1$$

with respect to des *is negligible.*

In the following subsections we usually write mod-secure instead of mod-ATK-secure, when the attack scenario is obvious from the context.

4.1 OK-security does not imply OU-security and CK-security

In this subsection we construct an OK-secure scheme which is neither OU-secure nor CK-secure. We start from an OK-secure scheme and assume existence of pseudorandom functions.

Let $\mathcal{R}_{\Omega,\Sigma}$ be an arbitrary predicate family, Π be a P-KEM for $\mathcal{R}_{\Omega,\Sigma}$. Furthermore, let $\mathcal{PRF}$ be a family of pseudorandom functions. The random choice of a pseudorandom function will be denoted by $f \leftarrow \mathcal{PRF}$ for simplicity.

Theorem 4.1. *Let* ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$ *be an attack scenario and $\mathcal{PRF}$ be a family of* PRF*s. Suppose* Π *is an* OK-ATK*-secure* P-KEM *for predicate family $\mathcal{R}_{\Omega,\Sigma}$ and family of key spaces $\mathcal{K}$. Then, there exists an* OK-ATK*-secure* P-KEM *scheme Π' for $\mathcal{R}_{\Omega,\Sigma}$ and $\mathcal{K}$ which is neither* OU-ATK*-secure nor* CK-ATK*-secure.*

Proof. Let $\Pi = (\mathrm{Setup}, \mathrm{KeyGen}, \mathrm{Encaps}, \mathrm{Decaps})$ and $\mathcal{PRF}$ be as defined in the theorem. Furthermore, let $\langle\cdot\rangle$ be any canonical representation of the master secret keys. W.l.o.g. we assume that for every des $\in \Omega$ and every $(\mathrm{msk}, \_) \in \left[\mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)\right]$ it holds $|\langle\mathrm{msk}\rangle| = \mathrm{MKLen}(\lambda)$ for some polynomial $\mathrm{MKLen}(\cdot)$. P-KEM $\Pi' = \left(\mathrm{Setup}', \mathrm{KeyGen}', \mathrm{Encaps}', \mathrm{Decaps}'\right)$ is defined as follows:

- $\mathrm{Setup}'\left(1^\lambda, \mathrm{des}\right)$: generate $(\mathrm{msk}, \mathrm{pp}_\kappa) \leftarrow \mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)$. Choose a pseudorandom function $f \leftarrow \mathcal{PRF}$ and output $\mathrm{msk}' := (\mathrm{msk}, f)$ and pp_κ.
- $\mathrm{KeyGen}'\left(\mathrm{msk}', \mathrm{kInd}\right)$ for $\mathrm{kInd} \in \mathbb{X}_\kappa$ and $\mathrm{msk}' = (\mathrm{msk}, f)$: generate $\mathrm{sk}_{\mathrm{kInd}} \leftarrow \mathrm{KeyGen}(\mathrm{msk}, \mathrm{kInd})$, choose a bit $b \leftarrow \{0,1\}$, set

$$\mathrm{rand} := \begin{cases} f(\mathrm{kInd}) & \text{if } b = 0 \\ f(\mathrm{kInd}) \oplus \langle\mathrm{msk}\rangle & \text{if } b = 1 \end{cases},$$

  and output $\mathrm{sk}'_{\mathrm{kInd}} := (\mathrm{sk}_{\mathrm{kInd}}, \mathrm{rand})$.
- $\mathrm{Encaps}'(\mathrm{cInd}) = \mathrm{Encaps}(\mathrm{cInd})$.
- $\mathrm{Decaps}'\left(\mathrm{sk}'_{\mathrm{kInd}}, \mathrm{CT}\right)$ for $\mathrm{sk}'_{\mathrm{kInd}} = (\mathrm{sk}_{\mathrm{kInd}}, \mathrm{rand})$ returns $\mathrm{Decaps}(\mathrm{sk}_{\mathrm{kInd}}, \mathrm{CT})$.

Scheme Π' is trivially broken in the OU-model and in the CK-model, where the adversary may get several keys for the same key index and hence, learns the master secret key. At the same time Π' is still OK-secure, since the adversary receives for every kInd ether $f(\mathrm{kInd})$ or $f(\mathrm{kInd}) \oplus \langle\mathrm{msk}\rangle$ and hence, due to the property of the pseudorandom function the values rand are useless for $\mathcal{A}$. To prove this formally it is sufficient to consider an imaginary scheme where f is replaced by a truly random function. For such a scheme it is clear that the additional values in the user secret keys are useless for adversary and the scheme is OK-secure due to the security property of Π. Furthermore, no ppt adversary can distinguish between this imaginary scheme and the scheme Π', since otherwise there would be a ppt distinguisher for $\mathcal{PRF}$. □

The construction in the proof reveals the weakness of the OK-model even though the scheme is artificial. Namely, the OK-model does not cover the practice where several keys for the same key index will exist. At least potentially, user secret keys for the same key index can leak more information about the master secret key than user secret keys for different key indices. Hence, already under CPA the OK-model makes unwarranted restrictions of adversarial abilities which might cause security issues.

4.2 OU-Security Does Not Imply OK-Security and CK-Security Under Chosen-Ciphertext Aattack

In this subsection we consider only chosen-ciphertext attack and construct an OU-secure scheme which is neither OK-secure nor CK-secure.

Theorem 4.2. *Let* ATK $\in$ {CCA1, CCA2} *be an attack scenario. Suppose* Π *is an* OU-ATK-*secure* P-KEM *for predicate family* $\mathcal{R}_{\Omega,\Sigma}$ *and family of key spaces* $\mathcal{K}$. *Then, there exists an* OU-ATK-*secure* P-KEM *scheme* Π' *for* $\mathcal{R}_{\Omega,\Sigma}$ *and* $\mathcal{K}$ *which is neither* OK-ATK-*secure nor* CK-ATK-*secure.*

Proof. Let $\mathbb{K}_\lambda = \{0,1\}^{\mathrm{KLen}(\lambda)}$. In the following construction of Π' we first assume that for all λ, all $(\mathrm{msk}, \mathrm{pp}_\kappa) \in \left[\mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)\right]$, all $\mathrm{kInd} \in \mathbb{X}_\kappa$ and all $\mathrm{sk} \in [\mathrm{KeyGen}(\mathrm{msk}, \mathrm{kInd})]$ it holds $|\langle \mathrm{sk} \rangle| = \mathrm{KLen}(\lambda)$, where $\langle \cdot \rangle$ is any canonical representation of the user secret keys. This enhances the perspicuity of the presented construction. Below, we explain how to drop this assumption.

P-KEM $\Pi' = \left(\mathrm{Setup}', \mathrm{KeyGen}', \mathrm{Encaps}', \mathrm{Decaps}'\right)$ as follows:

– $\mathrm{Setup}'\left(1^\lambda, \mathrm{des}\right) = \mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)$.

– $\mathrm{KeyGen}'(\mathrm{msk}, \mathrm{kInd})$: generate $\mathrm{sk} \leftarrow \mathrm{KeyGen}(\mathrm{msk}, \mathrm{kInd})$, choose a bit string $r \leftarrow \{0,1\}^{|\langle \mathrm{sk} \rangle|}$, output $\mathrm{sk}' := (\mathrm{sk}, r)$.

– $\mathrm{Encaps}'(\mathrm{cInd})$: generate $(\mathrm{CT}, \mathrm{K}) \leftarrow \mathrm{Encaps}(\mathrm{cInd})$ set $\mathrm{CT}' = 00 \| \mathrm{CT}$ and output $\left(\mathrm{CT}', \mathrm{K}\right)$.

– $\mathrm{Decaps}'\left(\mathrm{sk}', \mathrm{CT}'\right)$: parse $\mathrm{CT}' = b_1 b_2 \| \mathrm{CT}$, where $b_1, b_2 \in \{0,1\}$ and $\mathrm{sk}' = (\mathrm{sk}, r)$. Output

$$\mathrm{K} = \begin{cases} \mathrm{Decaps}(\mathrm{sk}, \mathrm{CT}) & \text{if } b_1 = b_2 = 0 \\ r & \text{if } b_1 = 1 \wedge b_2 = 0 \\ r \oplus \langle \mathrm{sk} \rangle & \text{if } b_1 = b_2 = 1 \\ \perp & \text{otherwise} \end{cases} .$$

Π' is OU-secure since using the decapsulation oracle the OU-adversary will be able to learn either r or $r \oplus \langle \mathrm{sk} \rangle$ which on their own are useless. This is due to the fact that every key is used only once. In the other two security models the adversary can get every user secret key using only two decapsulation queries and can trivially break the scheme.

Now we explain how to drop our assumption from the beginning of the proof. Indeed, it is sufficient to have a single key index kInd such that for all $\mathrm{sk} \in [\mathrm{KeyGen}(\mathrm{msk}, \mathrm{kInd})]$ it holds $|\langle \mathrm{sk} \rangle| \leq l(\lambda)$ for some polynomial $l(\lambda)$. In turn, this can be assumed w.l.o.g. If $l(\lambda) > \mathrm{KLen}(\lambda)$, we extend the encapsulation by $l = \lceil \log\left({}^{l(\lambda)}/_{\mathrm{KLen}(\lambda)}\right) \rceil + 2$ bits such that the first bit encodes if the encapsulation is correct ($b_1 = 0$) or not ($b_1 = 1$), the second bit encodes if r or $r \oplus \langle \mathrm{sk}_{\mathbf{kInd}} \rangle$ should be used, and the following bits encode the number of the block which should be returned. That is, if $b_1 = 1$, $b_2 = 1$ and $b_3 b_4 \ldots b_l = t$, the output will be the t's block of $r \oplus \langle \mathrm{sk} \rangle$, where every block is of size $\mathrm{KLen}(\lambda)$. Then, using $2 \cdot (l - 1)$ many queries one can get the key for kInd and break the scheme. □

In the case of PE schemes with message space $\{0,1\}^*$, we must not take care about the length of $\langle \mathrm{sk}_{\mathbf{kInd}} \rangle$ and can use the construction in the proof with two additional bits in the ciphertexts.

Even though the scheme Π' from the proof is artificial, it shows the main weakness of the OU-model. Namely, the model does not ensure that the decryption oracle does not leak partial information about the used user secret key if queried with an ill-formed ciphertext. This kind of partial information is difficult to exploit but might cause security issues.

4.3 Discussion

We discuss the results of this section with the security of known schemes in mind. As mentioned in the introduction to this section most known PE schemes are proved secure in the OK-security model or in the OU-security model. Hence, we examine the CK-CCA2-security of the above mentioned schemes and summarize the results in Table 2.

Construction	Type	Used model	CK-secure?
[9]	IBE	OK	YES
[10]	(H)IBE	OU	?
[13] (explicit check)	IB-KEM	OK	YES
[13] (implicit check)	IB-KEM	OK	YES
[14] (scheme I)	IBE	OU	YES
[14] (scheme II)	IBE	OU	YES
[18]	PE	OU	?
[24] (from verifyability)	ABE	OK	YES
[24] (from delegation)	ABE	OK	?
[25]	PE	OK	YES

Table 2. CK-CCA2-security for PE schemes proved to be OU-CCA2-secure or OK-CCA2-secure.

In the case of IBE, the key index is the identity of the user and hence, it is often (implicitly) assumed that for every key index there is a unique secret key. Furthermore, in the IBE scheme from [9] the keys are unique by construction and hence, all three security notions are identical for this scheme. But usually the IBE schemes do not have unique user secret keys (cf. [22, 13, 14]). The public verifiability of the first scheme in [13] ensures that the output of the decryption algorithm executed with different secret keys is the same, since ill-formed ciphertexts are explicitly rejected. The second scheme from [13] use fresh randomness during the decryption and rejects ill-formed ciphertexts with overwhelming probability independently of the used key. The schemes in [14] ensure that the ill-formed ciphertexts are rejected with overwhelming probability due to the authenticated symmetric encryption. Furthermore, the generic transformations from CPA to CCA2 secure schemes for attribute-based encryption (ABE) [24] and for predicate encryption [25] require the existence of verification algorithms which ensure that the output of the decryption algorithm is independent of the used secret key. For the remaining schemes from [10, 18, 24] it is at least not trivial to argue from the original proofs if the CK-security notion is satisfied or not. We leave this as an open question.

We notice that for most known PE schemes the correct form of the ciphertexts cannot be efficiently checked. This is due to the dual system encryption methodology used to construct most known adaptively secure PE schemes [23, 16, 17]. In the schemes from this technique there exist incorrectly formed ciphertexts which are indistinguishable from correctly generated ciphertexts. Hence, one can not simply reject ill-formed ciphertext in order to ensure CK-security.

Due to the results of this section regarding CCA2 security we encourage to use the CK-model in order to specify the security guarantees of the schemes precisely. If the CPA attacks are considered we recommend to use the simpler OU-model.

5 When and How to Restrict Challenge Decryption

In this section we consider possible restrictions of adversarial abilities regarding the decryption of the challenge under adaptive chosen-ciphertext attacks CCA2. We formally analyze the corresponding security notions in order to prevent mistakes and misunderstanding as previously made in the literature in the context of PKE, as described in [5].

5.1 Valid Adversaries and Security Notions

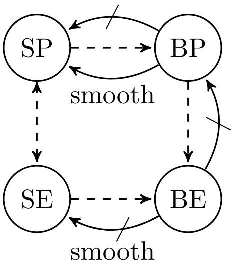
Due to the results in Section 4, in this section we only consider the CK-model. Furthermore, we mainly consider P-KEM and first of all redefine the set $\mathbf{A}_{\text{P-KEM}}$ of valid adversaries according to the syntax of CK-CCA2-model and using the results from Subsection 3.3.

$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}_{\text{P-KEM}}$ if and only if the following conditions are satisfied:

- Given public parameters pp_κ the algorithm $\mathcal{A}_1$ outputs the error symbol $\perp$ or a tuple $(\mathrm{cInd}^*, \mathrm{St})$ such that $\mathrm{cInd}^* \in \mathbb{Y}_\kappa$. The output of $\mathcal{A}_2$ is a bit.
- $\mathcal{A}_1$ and $\mathcal{A}_2$ query the oracle **CKGen** only on $\mathrm{kInd} \in \mathbb{X}_\kappa$.
- $\mathcal{A}_1$ and $\mathcal{A}_2$ submit index i to oracles **Open** and **Decaps** only after the i'th query to the **CKGen** oracle.
- For every (CT, i) submitted to the decapsulation oracle by $\mathcal{A}_1$ and $\mathcal{A}_2$ it holds $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}} \subset \mathbb{C}_{\mathrm{pp}_\kappa}$ and $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}) = 1$, where kInd is the key index submitted during the i'th covered key generation query.
- For every corrupted key index kInd it holds $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}^*) = 0$.

We dropped the last restriction of Lemma 3.1. The restriction regarding the decapsulation query on the challenge encapsulation will be considered separately in this section.

Let us shortly recall four different security notions for PKE identified and formalized in [5]. According to this work there are two dimensions in the definition of CCA2-security regarding the restrictions of adversaries to query the decryption of the challenge ciphertext. The first dimension specifies the style of the restrictions or rather how queries are disallowed. The authors differentiate between the penalty style definitions (denoted by P) and the exclusion style definition (denoted by E). So far we have used the former style in this work. That is, an adversary which violates restrictions of the security experiment is penalized at the end of the experiment. In the exclusion style definitions the set of adversaries is restricted from the beginning such that for every considered adversary the probability that the restrictions are violated is zero. Furthermore, we can disallow the adversary to query the decryption of the challenge ciphertext only in the second query phase (denoted by S) or in both query phases (denoted by B). That is either $\mathcal{A}_2$ is not allowed to query the decryption of CT^*, or $\mathcal{A}_1$ and $\mathcal{A}_2$ are not allowed to make such a query. As result, we have four different security notions denoted by SP, SE, BP, and BE.



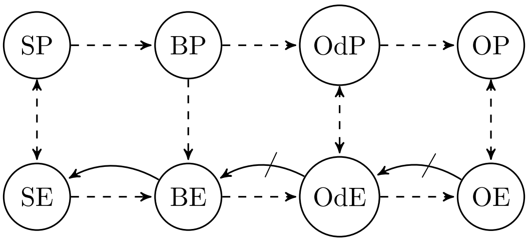
The continuous arrows denote implications and the barred arrows denote separations. The dashed arrows denote trivial implications.

Fig. 5. Relation between different security models for predicate encryptions.

For PE we can prove the same relations between mentioned notions as for PKE (cf. Figure 5), which is not surprising and was stated without a proof in [5] for IBE schemes. The proof ideas are the same as for PKE even though some extensions are required.

In the context of conventional KEM two additional notions have been considered in [5]. Namely, the first query phase can be completely dropped mainly due to the fact that the adversary cannot influence the generated challenge. This results in two additional security notions denoted by OP and OE, where "O" stays for "One phase". Indeed, for conventional KEM all six security notions were proved to be equivalent [5].

P-KEM substantially differ from conventional KEM, since the adversary might be able to influence the ciphertext index. We indeed prove that the corresponding security notions are not all equal due to this property. First of all, because of the key generation oracle we consider two additional security notions, where the adversary has access to this oracle in both phases, but the decapsulation oracle is available only in the second query phase. The corresponding penalty style security notion is denoted by OdP and the exclusion style notion is be denoted by OdE. Here "Od" stays for "One decapsulation" phase.



The continuous arrows denote implications and the barred arrows denote separations. The dashed arrows denote trivial implications.

Fig. 6. Relation between different security notions for P-KEM.

We prove that the OdE and the OdP security notions are weaker than the BE security notion (cf. Figure 6). The one-phase security notions OE and OP are even weaker. We also prove that the other four security notions (SP, SE, BP, and BE) are equivalent for P-KEM. Nevertheless, the reductions are not all tight and we advice against both BP and BE security models. The main difference between KEM and P-KEM is in the fact that whereas security of conventional KEM implies smoothness [5], this is not the case for P-KEM. Rather, BE-security implies that every ppt algorithm has only a negligible advantage in finding a ciphertext index with only few possible encapsulations, which we call a *weak ciphertext* index.

In order to formalize the exclusion style definitions we define restricted sets of adversaries. Furthermore, we keep a uniform security experiment and formalize the required restrictions in the first query phase by appropriate restrictions of the adversaries. Let S_1 and S_2 be the sets of encapsulations submitted to the decapsulation oracle by $\mathcal{A}_1$ and by $\mathcal{A}_2$, respectively. Furthermore, let SK_1 be the set of key indices corrupted by $\mathcal{A}_1$. For every security model $\mathrm{mod} \in \{\mathrm{SP}, \mathrm{BP}, \mathrm{OdP}, \mathrm{OP}, \mathrm{SE}, \mathrm{BE}, \mathrm{OdE}, \mathrm{OE}\}$ we define a set of adversaries denoted by $\mathbf{A}^{\mathrm{mod}}_{\text{P-KEM}}$. Namely, for every adversary $\mathcal{A} \in \mathbf{A}_{\text{P-KEM}}$ we define:

$$\mathcal{A} \in \mathbf{A}^{\mathrm{SE}}_{\text{P-KEM}} \text{ iff } \Pr\left[\mathrm{CT}^* \notin S_2\right] = 1 \ , \mathcal{A} \in \mathbf{A}^{\mathrm{BE}}_{\text{P-KEM}} \text{ iff } \Pr\left[\mathrm{CT}^* \notin S_1 \cup S_2\right] = 1 \ ,$$

$$\mathcal{A} \in \mathbf{A}^{\mathrm{OdP}}_{\text{P-KEM}} \text{ iff } \Pr\left[S_1 = \emptyset\right] = 1 \ , \mathcal{A} \in \mathbf{A}^{\mathrm{OdE}}_{\text{P-KEM}} \text{ iff } \Pr\left[S_1 = \emptyset \wedge \mathrm{CT}^* \notin S_2\right] = 1 \ ,$$

$$\mathcal{A} \in \mathbf{A}^{\mathrm{OP}}_{\text{P-KEM}} \text{ iff } \Pr\left[SK_1 = \emptyset \wedge S_1 = \emptyset\right] = 1 \ , \text{and}$$

$$\mathcal{A} \in \mathbf{A}^{\mathrm{OE}}_{\text{P-KEM}} \text{ iff } \Pr\left[SK_1 = \emptyset \wedge S_1 = \emptyset \wedge \mathrm{CT}^* \notin S_2\right] = 1 \ .$$

Furthermore, we define $\mathbf{A}^{\mathrm{SP}}_{\text{P-KEM}} = \mathbf{A}^{\mathrm{BP}}_{\text{P-KEM}} = \mathbf{A}_{\text{P-KEM}}$.

The exclusion/penalty style of the definitions only refer to the restriction regarding the decapsulation queries on the challenge encapsulation. The additional restrictions regarding the first query phase are all in exclusion style. This is only due to our goal of uniform templates. Alternatively we could define extra probability experiments with restricted first query phase or without the first query phase at all.

5.2 CCA2-Security Template for P-KEM

Let $\mathrm{mod} \in \{\mathrm{SP}, \mathrm{BP}, \mathrm{OdP}, \mathrm{OP}, \mathrm{SE}, \mathrm{BE}, \mathrm{OdE}, \mathrm{OE}\}$, $\mathcal{R}_{\Omega,\Sigma}$ be a predicate family, $\mathcal{K} = \{\mathbb{K}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of key spaces, and Π be a P-KEM for $\mathcal{R}_{\Omega,\Sigma}$ and $\mathcal{K}$. CCA2-security experiment $\text{P-KEM}^{\mathrm{CCA2,mod}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$ is defined in Figure 7.

The oracles are as defined in Table 1 except for decapsulation oracle which additionally stores all queried ciphertexts in S_1 or in S_2 in the first phase or in the second phase, respectively.

Definition 5.1. *Let* Π *be a* P-KEM *with public index for predicate family* $\mathcal{R}_{\Omega,\Sigma}$ *and* $\mathrm{mod} \in \{\mathrm{SP}, \mathrm{BP}, \mathrm{OdP}, \mathrm{OP}, \mathrm{SE}, \mathrm{BE}, \mathrm{OdE}, \mathrm{OE}\}$ *be a security model.* Π *is called* ***secure under adaptive chosen ciphertext attacks in model*** mod *(or* mod*-secure) if for every* $\mathrm{des} \in \Omega$ *the advantage of* $\mathcal{A} \in \mathbf{A}^{\mathrm{mod}}_{\text{P-KEM}}$ *with respect to* des*, defined by*

$$\text{Adv-P-KEM}^{\mathrm{CCA2,mod}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) := 2 \cdot \Pr\left[\text{P-KEM}^{\mathrm{CCA2,mod}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) = 1\right] - 1$$

is negligible.

```
$\textbf{P-KEM}^{\text{CCA2,mod}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) :$
$b \leftarrow \{0,1\}\,; S_1, S_2, S_{ck}, S_k \leftarrow \emptyset; (\text{msk}, \text{pp}_\kappa) \leftarrow \text{Setup}\left(1^\lambda, \text{des}\right);$
$(\text{cInd}^*, St) \leftarrow \mathcal{A}_1^{\textbf{CKGen}(\cdot),\textbf{Open}(\cdot),\textbf{Decaps}_1(\cdot,\cdot)}\left(1^\lambda, \text{pp}_\kappa\right);$
Output $b$ if the output of $\mathcal{A}_1$ is $\perp$;
$(\text{K}_0, \text{CT}^*) \leftarrow \text{Enc}\left(\text{pp}_\kappa, \text{cInd}^*\right); \text{K}_1 \leftarrow \mathbb{K}_\lambda; \text{K}^* := \text{K}_b;$
$b' \leftarrow \mathcal{A}_2^{\textbf{CKGen}(\cdot),\textbf{Open}(\cdot),\textbf{Decaps}_2(\cdot,\cdot)}\left(\text{K}^*, \text{CT}^*, St\right);$
Output :
  SE, BE, OdE, OE : return $b' = b$;
  SP, OdP, OP : return $b' = b \wedge (\text{CT}^* \notin S_2)$;
  BP : return $b' = b \wedge (\text{CT}^* \notin S_1 \cup S_2)$.
```

Fig. 7. CCA2-security experiment for different security notions of P-KEM.

5.3 Separation Results and Implication Results

In this subsection we prove the implication results and the separation results mentioned in Figure 6. It is easy to check that the four top down and the six rightwards relations hold mainly by the definition of the experiments and by the definitions of the corresponding sets of adversaries. The three implications between the exclusion and the corresponding penalty notions are covered by Remark 3.1 from Subsection 3.3.

OE-security does not imply OdE-security. In this subsection we show that OE-security notion is weaker then the OdE-security notion. The difference between the OE-security notion and the OdE-security notion is in the first query phase. Whereas in the OE-model this phase does not exist at all, in the OdE-model the adversary is allowed to corrupt user secret keys. To prove the stated separation, we present a counterexample based on a natural predicate family, which point up the weakness of the one phase notions for P-KEMs.

Consider a predicate family $\mathcal{R}$ for equality predicate with $\mathbb{X}_\kappa = \mathbb{Y}_\kappa = \{0,1\}^n$ and $n = \lambda$, that is $\text{R}_\kappa(\text{kInd}, \text{cInd}) = 1$ if and only if kInd = cInd. Notice that equality predicate corresponds to the identity-based schemes. Let $\mathcal{F}$ be a family of injective functions, $\text{Adv-Invert}_{\mathcal{F},\mathcal{I}}(\lambda)$ be the advantage of algorithm $\mathcal{I}$ in computing the preimage of y defined through by $f \leftarrow \mathcal{F}\left(1^\lambda\right)$, $x \leftarrow \{0,1\}^\lambda$, $y := f(x)$. $\mathcal{F}$ is called one-way if $\text{Adv-Invert}_{\mathcal{F},\mathcal{I}}(\lambda)$ is negligible for all ppt $\mathcal{I}$.

Theorem 5.1. *Suppose $\mathcal{F}$ is a family of injective one-way functions and Π is an* OE*-secure* P-KEM *for predicate family $\mathcal{R}$ and a family of key spaces $\mathcal{K}$. Then, there exists a* P-KEM Π' *for $\mathcal{R}$ and $\mathcal{K}$ which is not* OdE*-secure but is* OE*-secure. In particular, for every* $\text{des} \in \Omega$ *and every ppt* $\mathcal{A} \in \mathbf{A}^{\text{OE}}_{\Pi',\text{P-KEM}}$ *there is a ppt* $\mathcal{B} \in \mathbf{A}^{\text{OE}}_{\Pi,\text{P-KEM}}$ *and a ppt inverter $\mathcal{I}$ such that it holds*

$$\text{Adv-P-KEM}^{\text{CCA2,OE}}_{\Pi',\mathcal{A}}(\lambda, \text{des}) \leq \text{Adv-P-KEM}^{\text{CCA2,OE}}_{\Pi,\mathcal{B}}(\lambda, \text{des}) + \text{Adv-Invert}_{\mathcal{F},\mathcal{I}}(\lambda) \quad .$$

Proof. Let $\Pi = (\text{Setup}, \text{KeyGen}, \text{Encaps}, \text{Decaps})$ and $\mathcal{K} = \{\mathbb{K}_\lambda\}_{\lambda \in \mathbb{N}}$.

$\Pi' = \left(\text{Setup}', \text{KeyGen}', \text{Encaps}', \text{Decaps}'\right)$ as follows:

Setup$'\left(1^\lambda, \text{des}\right)$: Generate $(\text{pp}_\kappa, \text{msk}) \leftarrow \text{Setup}\left(1^\lambda, \text{des}\right)$. Choose $f \leftarrow \mathcal{F}\left(1^\lambda\right)$, $\text{cInd}_w \leftarrow \{0,1\}^n$, and compute $Y := f(\text{cInd}_w)$. Choose $\text{K}_w \leftarrow \mathbb{K}_\lambda$, set $\text{pp}'_\kappa := (\text{pp}_\kappa, f, \text{K}_w, Y)$, and $\text{msk}' = (\text{msk}, \text{cInd}_w)$. Output $\left(\text{pp}'_\kappa, \text{msk}'\right)$.

KeyGen$'\left(\text{pp}'_\kappa, \text{msk}', \text{kInd}\right)$ for $\text{pp}'_\kappa = (\text{pp}_\kappa, f_i, \text{K}_w, Y)$, and $\text{msk}' = (\text{msk}, \text{cInd}_w)$: Generate a secret key $\text{sk} \leftarrow \text{KeyGen}(\text{pp}_\kappa, \text{msk}, \text{kInd})$ and output $\text{sk}' := (\text{sk}, \text{cInd}_w)$.

Encaps$'(\text{pp}'_\kappa, \text{cInd})$ for $\text{pp}'_\kappa := (\text{pp}_\kappa, f, \text{K}_w, Y)$: If $f(\text{cInd}) = Y$ output key K_w and encapsulation $\text{CT}' = 0\|\left(\text{cInd}, 1^\lambda\right)$. Otherwise compute $(\text{K}, \text{CT}) \leftarrow \text{Encaps}\left(\text{pp}_{p,n}, \text{cInd}\right)$, set $\text{CT}' := 1\|\text{CT}$ and output $\left(\text{K}, \text{CT}'\right)$.

Decaps$'\left(\mathrm{pp}'_\kappa, \mathrm{CT}', \mathrm{sk}'\right)$ for $\mathrm{pp}'_\kappa := (\mathrm{pp}_\kappa, f, \mathrm{K}_w, Y)$, $\mathrm{CT}' = c\|\mathrm{CT}$, and $\mathrm{sk}' = (\mathrm{sk}, \mathrm{cInd}_w)$: Output K_w if $\mathrm{CT}' = 0\|\left(\mathrm{cInd}_w, 1^\lambda\right)$. If $c = 0$, output $\perp$. Otherwise output $\mathrm{Decaps}\left(\mathrm{pp}_\kappa, \mathrm{sk}, \mathrm{CT}\right)$.

Obviously, Π' is not OdE-secure since every secret key reveals cInd_w and the challenge for this weak ciphertext index can be easily solved.

Next we argue that Π' is OE-secure. Intuitively, the adversary in this model cannot exploit the modification of the scheme, since it has to commit to the challenge ciphertext index without querying the key generation oracle. Formally, from every $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}^{\mathrm{OE}}_{\Pi',\text{P-KEM}}$ we construct $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{A}^{\mathrm{OE}}_{\Pi,\text{P-KEM}}$ which simulates $\mathcal{A}$ and exploits its success probability except for the case that $\mathcal{A}_1$ outputs $\mathrm{cInd}^* = \mathrm{cInd}_w$. In turn, we can construct an inverter $\mathcal{I}$ for $\mathcal{F}$, which given the challenge (f, y) (where $f \leftarrow \mathcal{F}\left(1^\lambda\right)$, $x \leftarrow \{0,1\}^\lambda$, y:=f(x)) simulates $\mathcal{A}$ using f and $Y := y$, and exploits the event $\mathrm{cInd}^* = \mathrm{cInd}_w$ which corresponds to the event $f(\mathrm{cInd}^*) = y$. □

OdE-security does not imply BE-security. By definition of adversaries it holds $\mathbf{A}^{\mathrm{OdE}}_{\text{P-KEM}} \subset \mathbf{A}^{\mathrm{BE}}_{\text{P-KEM}}$. The adversaries in $\mathbf{A}^{\mathrm{OdE}}_{\text{P-KEM}}$ have the additional restriction that they do not make decapsulation queries in the first query phase. Consider a predicate family $\mathcal{R}$ for prefix predicates with $\mathbb{X}_\kappa = \mathbb{Y}_\kappa = \{0,1\}^{\leq n}$, where $n = 2 \cdot \lambda$, that is $\mathrm{R}_\kappa\left(\mathrm{kInd}, \mathrm{cInd}\right) = 1$ if and only if kInd is a prefix of cInd. Note that CCA2-secure P-KEM for $\mathcal{R}$ can be realized from hierarchical IBE [16].

Theorem 5.2. *Suppose $\mathcal{F}$ is a family of injective one-way functions and Π is an* OdE*-secure* P-KEM *for predicate family $\mathcal{R}$ and a family of key spaces $\mathcal{K}$. Then, there exists a* P-KEM *Π' for $\mathcal{R}$ and $\mathcal{K}$ which is not* BE*-secure but is* OdE*-secure. In particular, for every* des $\in \Omega$ *and every ppt $\mathcal{A} \in \mathbf{A}^{\mathrm{OdE}}_{\Pi',\text{P-KEM}}$ there exist ppt adversaries $\mathcal{B}, \mathcal{B}' \in \mathbf{A}^{\mathrm{OdE}}_{\Pi,\text{P-KEM}}$ and a ppt inverter $\mathcal{I}$ such that*

$$\begin{aligned}\text{Adv-P-KEM}^{\mathrm{CCA2,OdE}}_{\Pi',\mathcal{A}}\left(\lambda, \mathrm{des}\right) \leq \text{Adv-P-KEM}^{\mathrm{CCA2,OdE}}_{\Pi,\mathcal{B}}\left(\lambda, \mathrm{des}\right) + \text{Adv-Invert}_{\mathcal{F},\mathcal{I}}\left(\lambda\right) \\ +2 \cdot \lambda \cdot \text{Adv-P-KEM}^{\mathrm{CCA2,OdE}}_{\Pi,\mathcal{B}'}\left(\lambda, \mathrm{des}\right) \quad .\end{aligned}$$

Proof. Let $\Pi = (\mathrm{Setup}, \mathrm{KeyGen}, \mathrm{Encaps}, \mathrm{Decaps})$. Assume w.l.o.g. that $\mathcal{K} = \{\mathbb{K}_\lambda\}_{\lambda\in\mathbb{N}}$, $\mathbb{K}_\lambda = \{0,1\}^\lambda$ and that the encapsulations under cInd are of the form (cInd, ct).

$\Pi' = \left(\mathrm{Setup}', \mathrm{KeyGen}', \mathrm{Encaps}', \mathrm{Decaps}'\right)$ as follows:

Setup$'\left(1^\lambda, \mathrm{des}\right)$: Choose $(\mathrm{pp}_\kappa, \mathrm{msk}) \leftarrow \mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)$, $r = (r_1, \ldots, r_\lambda) \leftarrow \{0,1\}^\lambda$. Set $\mathrm{cInd}_w := 1^\lambda\|r$ and for all $i \in [\lambda]$ denote $\mathrm{kInd}_i = \mathrm{cInd}_i = 1^i$. For every $i \in [\lambda]$ generate $(\mathrm{CT}_i, \mathrm{K}_i) \leftarrow \mathrm{Encaps}\left(1^\lambda, \mathrm{cInd}_i\right)$ until the i'th bit of K_i is equal r_i. Choose $\mathrm{K}_w \leftarrow \mathbb{K}_\lambda$, $f \leftarrow \mathcal{F}$, and compute $Y := f\left(r\right)$. Output msk and $\mathrm{pp}'_\kappa := (\mathrm{pp}_\kappa, f, \mathrm{K}_w, Y, \mathrm{CT}_1, \ldots, \mathrm{CT}_\lambda)$.

KeyGen$'\left(\mathrm{pp}'_\kappa, \mathrm{msk}, \mathrm{kInd}\right)$: Output $\mathrm{sk} \leftarrow \mathrm{KeyGen}\left(\mathrm{pp}_\kappa, \mathrm{msk}, \mathrm{kInd}\right)$.

Encaps$'\left(\mathrm{pp}'_\kappa, \mathrm{cInd}\right)$: If $\mathrm{cInd} = 1^\lambda\|r'$ and $f\left(r'\right) = Y$ output K_w and $\mathrm{CT}' = 1\|\left(\mathrm{cInd}, 1^\lambda\right)$. Otherwise compute $(\mathrm{K}, \mathrm{CT}) \leftarrow \mathrm{Encaps}\left(\mathrm{pp}_\kappa, \mathrm{cInd}\right)$ and output K and $\mathrm{CT}' := 0\|\mathrm{CT}$.

Decaps$'\left(\mathrm{pp}'_\kappa, \mathrm{CT}', \mathrm{sk}\right)$: Parse $\mathrm{CT}' = b\|\left(\mathrm{cInd}, ct\right)$. Output K_w if $b = 1$, $ct = 1^\lambda$, $\mathrm{cInd} = 1^\lambda\|r'$, and $f\left(r'\right) = Y$. If $b = 1$ output $\perp$. Otherwise output $\mathrm{Decaps}\left(\mathrm{pp}_\kappa, \left(\mathrm{cInd}, ct\right), \mathrm{sk}\right)$.

Obviously Π' is not BE-secure, since cInd_w can be revealed using decapsulation oracle on CT_i's. The idea of the presented construction is as follows. In order to decapsulate just one $\mathrm{CT}_i \in \mathrm{pp}'_\kappa$ in the first query phase, an OdE-adversary needs a key for any prefix of cInd_λ. If $\mathcal{A}$ queries the corresponding key, it cannot use the weak ciphertext index $\mathrm{cInd}_w = 1^\lambda\|r$ for the challenge anymore.

Next we give a sketch of the proof that Π' is OdE-secure. Formally, from every ppt $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}^{\mathrm{OdE}}_{\Pi',\text{P-KEM}}$ we construct $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{A}^{\mathrm{OdE}}_{\Pi,\text{P-KEM}}$ which extends the public parameters, simulates $\mathcal{A}$ and exploits its success probability except for the case that $\mathcal{A}_1$ outputs $\mathrm{cInd}^* = \mathrm{cInd}_w$. Then, we prove that the probability for $\mathrm{cInd}^* = \mathrm{cInd}_w$ is negligible. Namely, we construct an inverter $\mathcal{I}$ for $\mathcal{F}$, which given the challenge (f, y) generates the public parameters and thereby computes all CT_i by $\mathrm{Encaps}\left(1^\lambda, \mathrm{cInd}_i\right)$. $\mathcal{I}$ wins if $\mathcal{A}$ outputs $\mathrm{cInd}^* = \mathrm{cInd}_w = 1^\lambda\|r'$, $f\left(r'\right) = y$. The last step is to prove that the probability for $\mathrm{cInd}^* = \mathrm{cInd}_w$ in the real experiment and in the experiment with public parameters as generated by $\mathcal{I}$ is the same except for negligible probability. For this last step it is sufficient to consider hybrid distributions, one for the modification of a single CT_i which results in the reduction algorithm $\mathcal{B}'$. □

BE-security implies SE-security. Next we prove that BE-security implies SE-security for P-KEMs. Even though we are able to prove this result, the corresponding reduction is not tight even for smooth schemes. Namely, the security guaranties linearly decrease in the number of decapsulation oracles of adversary in the first query phase. Contrary, the corresponding reduction for conventional KEMs from [5] is tight for smooth schemes.

The difference between SE-security and BE-security is in the definition of $\mathbf{A}^{\mathrm{SE}}_{\text{P-KEM}}$ and $\mathbf{A}^{\mathrm{BE}}_{\text{P-KEM}}$, since the experiments are completely equal. By definition, for every $\mathcal{A} \in \mathbf{A}_{\text{P-KEM}}$ it holds $\mathcal{A} \in \mathbf{A}^{\mathrm{SE}}_{\text{P-KEM}}$ if and only if $\Pr\left[\mathrm{CT}^* \in S_2\right] = 0$, and $\mathcal{A} \in \mathbf{A}^{\mathrm{BE}}_{\text{P-KEM}}$ if and only if $\Pr\left[\mathrm{CT}^* \in S_1 \cup S_2\right] = 0$. Every adversary against the scheme in SE-model with non-negligible advantage achieves a non-negligible advantage also in BE model as long as $p = \Pr\left[\mathrm{CT}^* \in S_1\right] = 0$. The main difficulty in the proof is to deal with adversaries with $p > 0$. The idea is to guess the number of the decapsulation query, where the challenge index cInd^* (or rather $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}^*}$) is used for the first time.

Theorem 5.3. *Suppose* Π *is a* BE*-secure* P-KEM *for predicate family* $\mathcal{R}_{\Omega,\Sigma}$*. Then,* Π *is* SE*-secure. In particular, for every* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}^{\mathrm{SE}}_{\text{P-KEM}}$ *and every* $\mathrm{des} \in \Omega$ *there exist* $\mathcal{A}', \mathcal{A}'' \in \mathbf{A}^{\mathrm{BE}}_{\text{P-KEM}}$ *such that*

$$\begin{aligned}\text{Adv-P-KEM}^{\mathrm{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) \le{}& (l+1) \cdot \text{Adv-P-KEM}^{\mathrm{CCA2,BE}}_{\Pi,\mathcal{A}'}(\lambda, \mathrm{des}) \\ &+ l \cdot \sqrt{2 \cdot (l+1) \cdot \text{Adv-P-KEM}^{\mathrm{CCA2,BE}}_{\Pi,\mathcal{A}''}(\lambda, \mathrm{des})}\ ,\end{aligned}$$

where $l = l(\lambda, \mathrm{des})$ *is the upper bound for the maximum number of decapsulation queries of* $\mathcal{A}_1$ *in* $\text{P-KEM}^{\mathrm{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$*. For smooth* Π *it holds*

$$\text{Adv-P-KEM}^{\mathrm{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) \le (l+1) \cdot \text{Adv-P-KEM}^{\mathrm{CCA2,BE}}_{\Pi,\mathcal{A}'}(\lambda, \mathrm{des}) + l \cdot \mathrm{Smth}_{\Pi}(\lambda, \mathrm{des})\ .$$

Proof. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}^{\mathrm{SE}}_{\text{P-KEM}}$ and $\mathrm{des} \in \Omega$ are arbitrary, but fixed. Let $l = l(\lambda, \mathrm{des})$ be the upper bound for the number of decapsulation queries of $\mathcal{A}_1$ in $\text{P-KEM}^{\mathrm{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$. W.l.o.g. we can assume that $\mathcal{A}_1$ makes exactly l decapsulation queries in this experiment.[1] Let us first estimate the advantage of $\mathcal{A}$ according to the event $\mathrm{CT}^* \in S_1$, which we denote by BD:

$$\text{Adv-P-KEM}^{\mathrm{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) \le \left(2 \cdot \Pr\left[\text{P-KEM}^{\mathrm{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) = 1 \wedge \overline{\mathrm{BD}}\right] - \Pr\left[\overline{\mathrm{BD}}\right]\right) + \Pr\left[\mathrm{BD}\right]\ . \quad (1)$$

For smooth schemes we can immediately estimate the probability for the event BD by union bound, since it holds $\Pr\left[\mathrm{BD}\right] \le l \cdot \mathrm{Smth}_{\Pi}(\lambda, \mathrm{des})$. But, whereas secure conventional KEMs are smooth as proved in [5], we can not prove this for P-KEMs. What we prove is that for every $\mathcal{A} \in \mathbf{A}^{\mathrm{SE}}_{\text{P-KEM}}$ the probability $\Pr\left[\mathrm{BD}\right]$ is negligible due to the BE-security of Π.

Let us first prove that the first summand in (1) is negligible. We construct an adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2) \in \mathbf{A}^{\mathrm{BE}}_{\text{P-KEM}}$ which exploits the success probability of $\mathcal{A} \in \mathbf{A}^{\mathrm{SE}}_{\text{P-KEM}}$ in the case that the event BD does not occur in $\text{P-KEM}^{\mathrm{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$. We have to ensure that the probability that $\mathcal{A}'$ (and especially $\mathcal{A}'_1$) submits CT^* to the decapsulation oracle is equal zero. Whereas in the second query phase this is trivial, the main difficulty is to prevent such a query in the first query phase, where the challenge ciphertext index cInd^* is not known. The main observation is that the probability for the event $\mathrm{CT}^* \in S_1$ is equal zero if and only if $\mathcal{A}_1$ does not query the decapsulation oracle on correctly generated encapsulations under cInd^*. Hence, $\mathcal{A}'_1$ will avoid to submit $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}^*}$ and this way we ensure the required property.

In order to prevent the event $\mathrm{CT}^* \in S_1$ in the first query phase, $\mathcal{A}'$ guesses by $j \leftarrow [l+1]$ the step where $\mathcal{A}_1$ will use the challenge index cInd^* *for the first time.* This can be a decapsulation query ($j \in [l]$) or the challenge ciphertext index itself ($j = l+1$). Let us assume at this point, that j was correctly guessed and $j \in [l]$. That is, the ciphertext index cInd_j of the submitted ciphertext CT_j is equal to the challenge ciphertext index and has not been used before. When the j'th decapsulation query is asked by $\mathcal{A}$, then $\mathcal{A}'$ will directly ask for the challenge on cInd_j and after that it will proceed to simulate $\mathcal{A}_1$. If the guess was correct, $\mathcal{A}'$ receives CT^* already in the j'th query and can avoid decapsulation queries on $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}^*}$ in the first query phase. Furthermore, if j was guessed correctly, $\mathcal{A}$ can be perfectly

[1] This can not be assumed if $\mathcal{A}_1$ outputs $\perp$ without querying the decapsulation oracle, but this event will be not relevant for the analysis.

simulated as long as it does not cause BD event. In turn, if j was guessed incorrectly, $\mathcal{A}'$ will output $\perp$ and hence, also avoid the BD event. Hence, the advantage of $\mathcal{A}'$ compared to the advantage of $\mathcal{A}$ decreases only due to the BD event $\mathrm{CT}^* \in S_1$ in $\text{P-KEM}^{\text{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$ and by factor $1/l+1$ due to the guess. We can prove that it holds

$$\text{Adv-P-KEM}^{\text{CCA2,BE}}_{\Pi,\mathcal{A}'}(\lambda, \text{des}) \geq \frac{1}{l+1} \cdot \Big(\text{Adv-P-KEM}^{\text{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) - \Pr[\text{BD}]\Big).$$

For smooth schemes we immediately get the statement in the theorem. For the general case we construct $\mathcal{A}'' = (\mathcal{A}''_1, \mathcal{A}''_2) \in \mathbf{A}^{\text{BE}}_{\text{P-KEM}}$ which exploits the event BD in $\text{P-KEM}^{\text{CCA2,SE}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$. Again we use similar ideas in order to ensure that $\mathcal{A}''$ never queries the decapsulation of CT^* in the first query phase. But there is an additional challenge. Namely, $\mathcal{A}''$ can not directly exploit the BD event $\mathrm{CT}^* \in S_1$ in order to break the challenge, since the corresponding encapsulated key is not given in the query. Rather, the main observation is that if this event occurs with non-negligible probability, we know that the probability $\Pr_{(\mathrm{K},\mathrm{CT})\leftarrow \mathrm{Encaps}(\mathrm{pp}_\kappa, \mathrm{cInd}^*)}[\mathrm{CT} = \mathrm{CT}^*]$ is non-negligible for the given ciphertext index cInd^*. Hence, $\mathcal{A}''$ just generates an additional encapsulation $\big(\mathrm{K}', \mathrm{CT}'\big)$ and then solves the challenge given the correct encapsulated key K' for CT^* if $\mathrm{CT}' = \mathrm{CT}^*$. Note, that $\mathcal{A}''$ will not even use $\mathcal{A}_2$. We can prove that it holds

$$\text{Adv-P-KEM}^{\text{CCA2,BE}}_{\Pi,\mathcal{A}''}(\lambda, \text{des}) \geq \frac{1}{2 \cdot (l+1) \cdot l^2} \cdot (\Pr[\text{BD}])^2 \quad .$$

From this we finally deduce the statement of the theorem. □

We deduce that the security notions SP, SE, BP, and BE are equivalent for P-KEMs. But the reductions are not all tight due to the security guarantees in Theorem 5.3. In particular, the reductions for BE ⇒ SE and BE ⇒ BP are not tight even for smooth schemes. For the implication BP ⇒ SP a tight reduction can be presented for smooth schemes, mainly due to the fact that we do not have to avoid the BD event as in the proof of Theorem 5.3. To the best of our knowledge all practical predicate encryption schemes are smooth and hence, we could also use the BP-model for these schemes. But since the probability for the event $\mathrm{CT}^* \in S_1$ can always be estimated by $l \cdot \mathrm{Smth}_\Pi(\lambda, \text{des})$, we do not really get any advantage from this model.

References

[1] Attrapadung, N.: Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In: Advances in Cryptology - EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577. Springer (2014)

[2] Attrapadung, N., Cui, Y., Galindo, D., Hanaoka, G., Hasuo, I., Imai, H., Matsuura, K., Yang, P., Zhang, R.: Relations among notions of security for identity based encryption schemes. In: LATIN 2006: Theoretical Informatics. LNCS, vol. 3887, pp. 130–141. Springer (2006)

[3] Attrapadung, N., Imai, H.: Dual-policy attribute based encryption: Simultaneous access control with ciphertext and key policies. IEICE Transactions 93-A(1), 116–125 (2010)

[4] Barbosa, M., Farshim, P.: On the semantic security of functional encryption schemes. In: Public-Key Cryptography - PKC 2013. LNCS, vol. 7778, pp. 143–161. Springer (2013)

[5] Bellare, M., Hofheinz, D., Kiltz, E.: Subtleties in the definition of IND-CCA: when and how should challenge decryption be disallowed? Journal of Cryptology 28(1), 29–48 (2015)

[6] Bellare, M., O'Neill, A.: Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. In: Cryptology and Network Security - CANS 2013. Lecture Notes in Computer Science, vol. 8257, pp. 218–234. Springer (2013)

[7] Blömer, J., Liske, G.: Construction of fully cca-secure predicate encryptions from pair encoding schemes. In: Topics in Cryptology - CT-RSA 2016. LNCS, vol. 9610, pp. 431–447. Springer (2016)

[8] Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM Journal on Computing 36(5), 1301–1328 (2007)

[9] Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM J. Comput. 32(3), 586–615 (2003)

[10] Boneh, D., Hamburg, M.: Generalized identity based and broadcast encryption schemes. In: Advances in Cryptology - ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer (2008)

[11] Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer (2011)

[12] Goldreich, O.: The Foundations of Cryptography - Volume II, Basic Applications. Cambridge University Press (2004)

[13] Kiltz, E., Galindo, D.: Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. Theoretical Computer Science 410(47-49), 5093–5111 (2009)

[14] Kiltz, E., Vahlis, Y.: CCA2 secure IBE: standard model efficiency through authenticated symmetric encryption. In: Topics in Cryptology - CT-RSA 2008. LNCS, vol. 4964, pp. 221–238. Springer (2008)

[15] Koblitz, N., Menezes, A.: Another look at security definitions. Advances in Mathematics of Communications (AMC) 7(1), 1–38 (2013)

[16] Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer (2010)

[17] Lewko, A.B., Waters, B.: New proof methods for attribute-based encryption: Achieving full security through selective techniques. In: Advances in Cryptology - CRYPTO 2012. LNCS, vol. 7417, pp. 180–198. Springer (2012)

[18] Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Advances in Cryptology - CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer (2010)

[19] O'Neill, A.: Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556 (2010), `https://eprint.iacr.org/2010/556`

[20] Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Advances in Cryptology - EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer (2005), full

[21] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Advances in Cryptology, Proceedings of CRYPTO '84. LNCS, vol. 196, pp. 47–53. Springer (1984)

[22] Waters, B.: Efficient identity-based encryption without random oracles. In: Advances in Cryptology - EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer (2005)

[23] Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Advances in Cryptology - CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer (2009)

[24] Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N.: Generic constructions for chosen-ciphertext secure attribute based encryption. In: Public Key Cryptography - PKC 2011. LNCS, vol. 6571, pp. 71–89. Springer (2011)

[25] Yamada, S., Attrapadung, N., Santoso, B., Schuldt, J.C.N., Hanaoka, G., Kunihiro, N.: Verifiable predicate encryption and applications to CCA security and anonymous predicate authentication. In: Public Key Cryptography - PKC 2012. LNCS, vol. 7293, pp. 243–261. Springer (2012)

A Equivalence of SS and IND for PE

In order to prevent additional work regarding the ability of the adversary to abort in the first query phase let us consider here how it influence the advantage of the adversary. These results will be used in the following proofs.

Let us consider an arbitrary but fixed adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}^{\mathrm{SS}}$ which satisfies the conditions of the conditions of Lemma 3.1 adopted to PE and SS-definition. Let des $\in \Omega$ be arbitrary but fixed. We denote by E the event that $\mathcal{A}_1$ outputs $\perp$ in the experiment $\text{SS-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$. Furthermore, let $p_{\text{des}}(\lambda)$ be the probability for the event $\overline{\text{E}}$ in this experiment, that is $p_{\text{des}}(\lambda) = \Pr\left[\overline{\text{E}}\right] = 1 - \Pr[\text{E}]$. $\mathcal{A}$ never cause the event BadQuery and hence, for every simulator $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ of $\mathcal{A}$ with respect to des the challenge templates are identically distributed in $\text{SS-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$ and $\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \text{des})$. In particular, the probability that $\mathcal{A}'_1$ outputs $\perp$ (event E$'$) is equal to $1 - p_{\text{des}}(\lambda)$. Since $\mathcal{A}$ as well as $\mathcal{A}'$ loose in their experiments if the event E respectively E$'$ occur, it holds

$$\begin{aligned}
&\text{Adv-SS-PE}^{\text{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}(\lambda, \text{des}) \\
\overset{\text{by def.}}{=}\ & \Pr\left[\text{SS-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) = 1\right] - \Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \text{des}) = 1\right] \\
=\ & p_{\text{des}}(\lambda) \cdot \Big(\Pr\left[\text{SS-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) = 1 \mid \overline{\text{E}}\right] \\
& \qquad - \Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \text{des}) = 1 \mid \overline{\text{E}'}\right]\Big). \qquad (2)
\end{aligned}$$

By the analysis of adversarial advantage, if $p_{\text{des}}(\lambda)$ is negligible, the second factor does not matter or rather such an adversary, which outputs $\perp$ with all except negligible probability, can be ignored. Contrary, if $p_{\text{des}}(\lambda)$ is not negligible, it will be sufficient to consider the success probability of $\mathcal{A}$ conditioned on the fact that $\mathcal{A}_1$ did not output $\perp$.

Let us also consider the advantage of an arbitrary adversary $\mathcal{A} = (A_1, A_2) \in \mathbf{A}^{\mathrm{IND}}_{\Pi}$ which satisfies the conditions of Lemma 3.1 adopted to PE and SS-definition. By E we denote the event that $\mathcal{A}_1$ outputs $\perp$ in the experiment $\text{IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$. Furthermore, let $p_{\text{des}}(\lambda)$ be the probability for the event $\overline{\text{E}}$ in this experiment, that is $p_{\text{des}}(\lambda) = \Pr\left[\overline{\text{E}}\right] = 1 - \Pr[\text{E}]$. Then, it holds by the definition of the experiment:

$$\begin{aligned}
\text{Adv-IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) \overset{\text{by def.}}{=}\ & 2 \cdot \Pr\left[\text{IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) = 1\right] - 1 \\
=\ & 2 \cdot \left(\frac{1}{2} \cdot \Pr[\text{E}] + \Pr\left[\text{IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) = 1 \wedge \overline{\text{E}}\right]\right) - 1 \\
=\ & 2 \cdot \Pr\left[\text{IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) = 1 \mid \overline{\text{E}}\right] \cdot \Pr\left[\overline{\text{E}}\right] - (1 - \Pr[\text{E}]) \\
=\ & p_{\text{des}}(\lambda) \cdot \left(2 \cdot \Pr\left[\text{IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) = 1 \mid \overline{\text{E}}\right] - 1\right) \\
=\ & p_{\text{des}}(\lambda) \cdot \Big(\Pr\left[\text{IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) = 1 \mid \overline{\text{E}} \wedge b = 0\right] \\
& \qquad + \Pr\left[\text{IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) = 1 \mid \overline{\text{E}} \wedge b = 1\right] - 1\Big) \\
=\ & p_{\text{des}}(\lambda) \cdot \Big(\Pr\left[b' = 0 \mid \overline{\text{E}} \wedge b = 0\right] \\
& \qquad + \Pr\left[b' = 1 \mid \overline{\text{E}} \wedge b = 1\right] - 1\Big)\ , \qquad (3)
\end{aligned}$$

where in the penultimate equation we used the fact the the choice of b and the event E are independent. All except the last equation hold also if $\mathcal{A}$ cause the event BadQuery. In particular, the second equation holds due to the fact that the event BadQuery cannot be caused if $\mathcal{A}_1$ outputs $\perp$.

A.1 Indistinguishability implies semantic security

The following lemma shows that it is sufficient to prove the indistinguishability of encryptions in order to prove semantic security of the scheme.

Lemma A.1. *Suppose* Π *is a predicate encryption scheme for a predicate family* $\mathcal{R}_{\Omega,\Sigma}$ *which is* IND-ATK-*secure, where* ATK $\in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$*. Then,* Π *is* SS-ATK-*secure. In particular, for every ppt* *adversary* $\mathcal{A} \in \mathbf{A}^{\mathrm{SS}}$ *there exists a ppt simulator* $\mathcal{A}'$ *for* $\mathcal{A}$ *and a ppt adversary* $\mathcal{B} \in \mathbf{A}^{\mathrm{IND}}$ *such that for every* $\mathrm{des} \in \Omega$, $\lambda \in \mathbb{N}$ *it holds*

$$\text{Adv-IND-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B}}(\lambda, \mathrm{des}) \geq \text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}(\lambda, \mathrm{des}) \quad .$$

Proof. Note that the statement in the lemma is even stronger than required by the definition of semantic security, since $\mathcal{A}'$ will not depend on des.

Define a modification $\text{SS-PE-Mod}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$ of the experiment $\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$, where CT^* is computed by $\mathrm{Enc}\left(\mathrm{pp}_\kappa, \mathrm{cInd}^*, 1^{|\hat{m}|}\right)$ instead of $\mathrm{Enc}\left(\mathrm{pp}_\kappa, \mathrm{cInd}^*, \hat{m}\right)$. It is important to notice that the experiments are identical until the generation of the challenge ciphertext.

Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}^{\mathrm{SS}}$ be an arbitrary but fixed adversary against Π which does not cause the BadQuery. In order to prove the statement of the lemma it is sufficient to show that the advantage of $\mathcal{A}$ is negligible (due to Lemma 3.1 adopted to PE and SS-definition). Note that even though $\mathcal{A}$ never cause the event BadQuery in $\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$, the probability $\Pr[\mathrm{BadQuery}]$ in the modified experiment $\text{SS-PE-Mod}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$ is not necessarily zero. The same holds also for the other conditions defined in Lemma 3.1. Nevertheless, due to the special modification of the experiment $\text{SS-PE-Mod}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$ all these properties hold for $\mathcal{A}_1$. This will be explicitly used in our construction of $\mathcal{B} \in \mathbf{A}^{\mathrm{IND}}$ below.

Let additionally $\mathrm{des} \in \Omega$, $\lambda \in \mathbb{N}$ be arbitrary, but fixed. First, we construct a simulation algorithm $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ which uses $\mathcal{A}$ as a subroutine. Note, that simulator $\mathcal{A}'_2$ does not receive the encryption of $\hat{m}$ and hence, cannot simulate $\mathcal{A}_2$ as in $\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$. But $\mathcal{A}'_2$ can properly simulate the view of $\mathcal{A}$ in $\text{SS-PE-Mod}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$. Intuitively, due to the indistinguishability of ciphertexts $\mathcal{A}$ can not notice the difference. This will be formally proved in the second part of the proof.

```
$\mathcal{A}'_1\left(1^\lambda, \mathrm{des}\right)$:
  – Generate $(\mathrm{msk}, \mathrm{pp}_\kappa) \leftarrow \mathrm{Setup}\left(1^\lambda, \mathrm{des}\right)$.
  – Compute $\left(\mathrm{cInd}^*, \left(\hat{\mathcal{M}}, h, f\right), St\right) \leftarrow \mathcal{A}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa, \mathrm{msk}, \cdot)}\left(1^\lambda, \mathrm{pp}_\kappa\right)$.
  – Output $\perp$ if $\mathcal{A}_1$ outputs $\perp$.
  – Set $St' := (St, \mathrm{msk}, \mathrm{pp}_\kappa, \mathrm{cInd}^*)$ and output $\left(\mathrm{cInd}^*, \left(\hat{\mathcal{M}}, h, f\right), St'\right)$.

$\mathcal{A}'_2\left(1^{|\hat{m}|}, h(\hat{m}), St'\right)$ with $St' = (St, \mathrm{msk}, \mathrm{pp}_\kappa, \mathrm{cInd}^*)$:
  – Generate $\mathrm{CT}' \leftarrow \mathrm{Enc}\left(\mathrm{pp}_\kappa, \mathrm{cInd}^*, 1^{|\hat{m}|}\right)$.
  – Compute $\nu \leftarrow \mathcal{A}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa, \mathrm{msk}, \cdot)}\left(\mathrm{CT}', |\hat{m}|, h(\hat{m}), St\right)$ and output the output of $\mathcal{A}_2$.
```

The distribution of the challenge template generated by $\mathcal{A}'_1$ is as required in the Definition 3.1, since $\mathcal{A}'_1$ uses correctly generated $(\mathrm{msk}, \mathrm{pp}_\kappa)$ and $\mathcal{A}_1$ in order to generate $\left(\hat{\mathcal{M}}, h, f\right)$. Hence, $\mathcal{A}'$ is a simulator of $\mathcal{A}$. Furthermore, by construction of $\mathcal{A}'$ the view of $\mathcal{A}$ in this experiment is the same as in the experiment $\text{SS-PE-Mod}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$. Hence, if $\mathcal{A}$ wins in $\text{SS-PE-Mod}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$, then $\mathcal{A}'$ wins in $\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \mathrm{des})$. The opposite direction does not necessarily hold, since $\mathcal{A}$ might get a penalty (caused by $\mathcal{A}_2$) whereas $\mathcal{A}'$ could still win in this case. We deduce that it holds

$$\Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \mathrm{des}) = 1\right] \geq \Pr\left[\text{SS-PE-Mod}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) = 1\right] \quad . \tag{4}$$

Now we are ready to construct an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{A}^{\mathrm{IND}}$ for the indistinguishability experiment $\text{IND-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B}}(\lambda, \mathrm{des})$ and to relate its success probability to $\text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}(\lambda, \mathrm{des})$:

$\mathcal{B}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(1^\lambda,\mathrm{pp}_\kappa\right)$:

- Compute $\left(\mathrm{cInd}^*,\left(\hat{\mathcal{M}},h,f\right),St\right)\leftarrow\mathcal{A}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(1^\lambda,\mathrm{pp}_\kappa\right)$ using own oracles in order to answer the queries of $\mathcal{A}_1$. Output $\perp$ if the output of $\mathcal{A}_1$ is $\perp$.
- Choose $\hat{m}\leftarrow\hat{\mathcal{M}}\left(\mathcal{U}_{\mathrm{poly}(\lambda)}\right)$, set $m_0:=1^{|\hat{m}|}$ and $m_1:=\hat{m}$. Set $St':=\left(St,|\hat{m}|,h\left(\hat{m}\right),f\left(\hat{m}\right)\right)$.
- Output $(\mathrm{cInd}^*,m_0,m_1,St')$.

$\mathcal{B}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(\mathrm{CT}^*,St'\right)$ with $St'=\left(St,n,\hat{h},\hat{f}\right)$:

- Simulate $\nu\leftarrow\mathcal{A}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(\mathrm{CT}^*,n,\hat{h},St\right)$ using own oracles in order to answer the queries of $\mathcal{A}_2$. Thereby abort the simulation of $\mathcal{A}_2$ and output $b'=0$ if $\mathcal{A}_2$ cause the event BadQuery.
- Output $b'=1$ if $\nu=\hat{f}$. Otherwise output 0.

By construction it obviously holds $\mathcal{B}\in\mathbf{A}^{\mathrm{IND}}$. $\mathcal{B}$ (in particular $\mathcal{B}_1$) redirects all queries of $\mathcal{A}$ and does not make additional queries. Furthermore, in the first query phase $\mathcal{B}_1$ perfectly simulates the view of $\mathcal{A}_1$ in the experiment $\text{SS-PE}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda,\mathrm{des})$ (respectively in the experiment $\text{SS-PE-Mod}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda,\mathrm{des})$). Hence, $\mathcal{B}_1$ never cause the event BadQuery. In turn, $\mathcal{B}_2$ explicitly prevents the event BadQuery. We deduce that $\mathcal{B}$ never cause a penalty and wins if $b'=b$.

Let E be the event that an adversary outputs $\perp$ after the first query phase of the corresponding experiment (we will make the adversary explicit in the subindex). Furthermore, let $p_{\mathrm{des}}(\lambda)=\Pr\left[\overline{\mathrm{E}_{\mathcal{B}}}\right]=\Pr\left[\overline{\mathrm{E}_{\mathcal{A}}}\right]$. Let b be the challenge bit of $\mathcal{B}$. We will analyze the view of $\mathcal{A}_2$ and the success probability of $\mathcal{B}$ for both values of b conditioned on the event $\overline{\mathrm{E}_{\mathcal{B}}}$.

- $\overline{\mathrm{E}_{\mathcal{B}}}\wedge b=0$: It holds $\mathrm{CT}^*=\mathrm{Enc}\left(\mathrm{pp}_\kappa,\mathrm{cInd}^*,1^{|\hat{m}|}\right)$, where $\hat{m}\leftarrow\hat{\mathcal{M}}\left(\mathcal{U}_{\mathrm{poly}(\lambda)}\right)$. Hence, the view of $\mathcal{A}_2$ is as in the experiment $\text{SS-PE-Mod}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda,\mathrm{des})$. In particular, $\mathcal{A}_2$ might still cause the event BadQuery. In this case $\mathcal{B}_2$ outputs $b'=0$ without causing the penalty and wins. We deduce

$$\begin{aligned}\Pr\left[b'=0\;\middle|\;\overline{\mathrm{E}_{\mathcal{B}}}\wedge b=0\right] &\overset{(*)}{=} \Pr\left[\mathrm{BadQuery}\;\middle|\;\overline{\mathrm{E}_{\mathcal{A}}}\right]\\ &\quad+\Pr\left[\nu\neq f\left(\hat{m}\right)\wedge\overline{\mathrm{BadQuery}}\;\middle|\;\overline{\mathrm{E}_{\mathcal{A}}}\right]\\ &= 1-\Pr\left[\nu=f\left(\hat{m}\right)\wedge\overline{\mathrm{BadQuery}}\;\middle|\;\overline{\mathrm{E}_{\mathcal{A}}}\right]\\ &= 1-\frac{1}{p_{\mathrm{des}}(\lambda)}\cdot\Pr\left[\nu=f\left(\hat{m}\right)\wedge\overline{\mathrm{BadQuery}}\wedge\overline{\mathrm{E}_{\mathcal{A}}}\right]\\ &\overset{\text{by def.}}{=} 1-\frac{1}{p_{\mathrm{des}}(\lambda)}\cdot\Pr\left[\text{SS-PE-Mod}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda,\mathrm{des})=1\right]\\ &\overset{4}{\geq} 1-\frac{1}{p_{\mathrm{des}}(\lambda)}\cdot\Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda,\mathrm{des})=1\right]\quad,\end{aligned}$$

where in $(*)$ we switch from probability distribution defined by $\text{IND-PE}_{\Pi,\mathcal{B}}^{\mathrm{ATK}}(\lambda,\mathrm{des})$ to the probability distribution defined by $\text{SS-PE-Mod}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda,\mathrm{des})$.

- $\overline{\mathrm{E}_{\mathcal{B}}}\wedge b=1$: It holds $\mathrm{CT}^*=\mathrm{Enc}\left(\mathrm{pp}_\kappa,\mathrm{cInd}^*,\hat{m}\right)$, where $\hat{m}\leftarrow\hat{\mathcal{M}}\left(\mathcal{U}_{\mathrm{poly}(\lambda)}\right)$. Hence, the view of $\mathcal{A}_2$ is as in the experiment $\text{SS-PE}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda,\mathrm{des})$. In particular, $\mathcal{A}_2$ never cause the event BadQuery. We deduce

$$\begin{aligned}\Pr\left[b'=1\;\middle|\;\overline{\mathrm{E}_{\mathcal{B}}}\wedge b=1\right] &\overset{(*)}{=} \Pr\left[\nu=f\left(\hat{m}\right)\;\middle|\;\overline{\mathrm{E}_{\mathcal{A}}}\right]\\ &= \frac{1}{p_{\mathrm{des}}(\lambda)}\cdot\Pr\left[\nu=f\left(\hat{m}\right)\wedge\overline{\mathrm{E}_{\mathcal{A}}}\right]\\ &\overset{\text{by def.}}{=} \frac{1}{p_{\mathrm{des}}(\lambda)}\cdot\Pr\left[\text{SS-PE}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda,\mathrm{des})=1\right]\quad,\end{aligned}$$

where in $(*)$ we switch from probability distribution defined by $\text{IND-PE}_{\Pi,\mathcal{B}}^{\mathrm{ATK}}(\lambda,\mathrm{des})$ to the probability definition defined by $\text{SS-PE}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda,\mathrm{des})$.

$\mathcal{B}$ does not cause the event BadQuery and hence, the advantage of $\mathcal{B}$ is as follows

$$\begin{aligned}
& \text{Adv-IND-PE}^{\text{ATK}}_{\Pi,\mathcal{B}}(\lambda, \text{des}) \\
\overset{(3)}{=} \;& p_{\text{des}}(\lambda) \cdot \Big(\Pr\left[b' = 0 \;\middle|\; \overline{\text{E}} \wedge b = 0\right] + \Pr\left[b' = 1 \;\middle|\; \overline{\text{E}} \wedge b = 1\right] - 1\Big) \\
\geq \;& \Pr\left[\text{SS-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) = 1\right] - \Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \text{des}) = 1\right] \\
\overset{\text{by def.}}{=} \;& \text{Adv-SS-PE}^{\text{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}(\lambda, \text{des}) \quad .
\end{aligned}$$

But for every $\mathcal{B} \in \mathbf{A}^{\text{IND}}$ and every $\text{des} \in \Omega$ the advantage $\text{Adv-IND-PE}^{\text{ATK}}_{\Pi,\mathcal{B}}(\lambda, \text{des})$ is negligible due to the IND-ATK-security of Π.

Due to this lemma, in order to prove that a predicate encryption scheme is semantically secure it is sufficient to prove that this scheme has indistinguishable encryptions.

A.2 Semantic security implies indistinguishability of encryptions

The following lemma shows that the notion of indistinguishable encryptions is not to strong.

Lemma A.2. *Let* $\text{ATK} \in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$ *and* Π *be a predicate encryption scheme for predicate family* $\mathcal{R}_{\Omega,\Sigma}$ *which is* SS-ATK-*secure. Then,* Π *is* IND-ATK-*secure. In particular, for every ppt adversary* $\mathcal{A} \in \mathbf{A}^{\text{IND}}$ *there exists a ppt adversary* $\mathcal{B} \in \mathbf{A}^{\text{SS}}$ *such that for every* $\text{des} \in \Omega$ *and every ppt simulator* $\mathcal{B}'$ *of* $\mathcal{B}$ *with respect to* des *it holds*

$$2 \cdot \text{Adv-SS-PE}^{\text{ATK}}_{\Pi,\mathcal{B},\mathcal{B}'}(\lambda, \text{des}) \geq \text{Adv-IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des}) \quad .$$

Proof. Let $\lambda \in \mathbb{N}$, $\text{des} \in \Omega$, $\text{ATK} \in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$, and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}^{\text{IND}}$ be arbitrary but fixed such that $\mathcal{A}$ satisfies the restrictions defined in Lemma 3.1 (adopted to PE). In particular, $\mathcal{A}$ never cause the event BadQuery.

We construct $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{A}^{\text{SS}}$ which exploits the advantage of $\mathcal{A}$ as follows:

$\mathcal{B}_1^{\mathbf{O}_1(\text{pp}_\kappa, \text{msk}, \cdot)}\left(1^\lambda, \text{pp}_\kappa\right)$:

- Simulate $(\text{cInd}^*, m_0, m_1, St) \leftarrow \mathcal{A}_1^{\mathbf{O}_1(\text{pp}_\kappa, \text{msk}, \cdot)}\left(1^\lambda, \text{pp}_\kappa\right)$ using the own oracles in order to answer the queries of $\mathcal{A}_1$.
- Output $\perp$ if the output of $\mathcal{A}_1$ is $\perp$.
- Set $\hat{\mathcal{M}}$ to a circuit corresponding to the uniform distribution on $\{m_0, m_1\}$, set h to an arbitrary function such that $h(m_0) = h(m_1)$, and set f to an arbitrary function such that $f(m_0) = 0$ and $f(m_1) = 1$.[a]
- Output $\left(\text{cInd}^*, \left(\hat{\mathcal{M}}, h, f\right), St\right)$.

$\mathcal{B}_2^{\mathbf{O}_2(\text{pp}_\kappa, \text{msk}, \cdot)}\left(\text{CT}^*, |\hat{m}|, h(\hat{m}), St\right)$:

- Simulate $b' \leftarrow \mathcal{A}_2^{\mathbf{O}_2(\text{pp}_\kappa, \text{msk}, \cdot)}(\text{CT}^*, St)$ using the own oracles in order to answer the queries of $\mathcal{A}_1$.
- Output $\nu := b'$.

[a] For simplicity we assumed w.l.o.g. that $m_0 \neq m_1$. If $m_0 = m_1$, $\mathcal{A}$ cannot have any advantage and $\mathcal{B}_1$ can output $\perp$.

$\mathcal{B} \in \mathbf{A}^{\text{SS}}$ by construction. Furthermore, the view of $\mathcal{A}$ is as defined in $\text{IND-PE}^{\text{ATK}}_{\Pi,\mathcal{A}}(\lambda, \text{des})$. In particular, $\mathcal{A}_1$ receives correctly generated public parameters and all queries of $\mathcal{A}_1$ are correctly answered. In turn, $\mathcal{A}_2$ receives the encryption of m_0 or the encryption of m_1, both with probability $1/2$ and all queries of $\mathcal{A}_2$ are also correctly answered. Hence, $\mathcal{B}$ never cause the event BadQuery since $\mathcal{A}$ never cause the corresponding event. Consider an arbitrary simulator $\mathcal{B}' = (\mathcal{B}'_1, \mathcal{B}'_2)$ of $\mathcal{B}$ with respect to an arbitrary but fixed $\text{des} \in \Omega$. Due to the requirement on $\mathcal{B}'$ the distribution of $\left(\hat{\mathcal{M}}, h, f\right)$ generated by $\mathcal{B}'_1$ must be the same as above in the experiment with $\mathcal{B}$.

By construction, $\mathcal{B}_1$ outputs $\perp$ if and only if $\mathcal{A}_1$ outputs $\perp$ and $\mathcal{B}'_1$ outputs $\perp$ with the same probability by definition. Let E be the event that an adversary (or a simulator) outputs $\perp$ after the first query phase (we will make the adversary explicit in the subindex). Furthermore, let $p_{\mathrm{des}}(\lambda) = \Pr\left[\overline{\mathrm{E}_{\mathcal{B}}}\right] = \Pr\left[\overline{\mathrm{E}_{\mathcal{B}'}}\right] = \Pr\left[\overline{\mathrm{E}_{\mathcal{A}}}\right]$. Due to $h(m_0) = h(m_1)$ and $|m_0| = |m_1|$ the input of $\mathcal{B}'_2$ is independent of $\hat{m}$ and it holds

$$\Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{B}'}(\lambda, \mathrm{des}) = 1 \mid \overline{\mathrm{E}_{\mathcal{B}'}}\right] = \frac{1}{2} \ ,$$

since $\hat{\mathcal{M}}$ is the uniform distribution on $\{m_0, m_1\}$, $f(m_0) = 0$, and $f(m_1) = 1$.

Consider the sucess probability of $\mathcal{B}$. By construction it holds

$$\begin{aligned}
\Pr\left[\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B}}(\lambda, \mathrm{des}) = 1 \mid \overline{\mathrm{E}_{\mathcal{B}}}\right] &= \Pr\left[\hat{m} = m_0 \wedge b' = f(\hat{m}) \mid \overline{\mathrm{E}_{\mathcal{B}}}\right] + \Pr\left[\hat{m} = m_1 \wedge b' = f(\hat{m}) \mid \overline{\mathrm{E}_{\mathcal{B}}}\right] \\
&= \Pr\left[b' = f(\hat{m}) \mid \overline{\mathrm{E}_{\mathcal{B}}} \wedge \hat{m} = m_0\right] \cdot \Pr\left[\hat{m} = m_0 \mid \overline{\mathrm{E}_{\mathcal{B}}}\right] \\
&\quad + \Pr\left[b' = f(\hat{m}) \mid \overline{\mathrm{E}_{\mathcal{B}}} \wedge \hat{m} = m_1\right] \cdot \Pr\left[\hat{m} = m_1 \mid \overline{\mathrm{E}_{\mathcal{B}}}\right] \\
&\overset{f}{=} \frac{1}{2} \cdot \left(\Pr\left[b' = 0 \mid \overline{\mathrm{E}_{\mathcal{B}}} \wedge b = 0\right] + \Pr\left[b' = 1 \mid \overline{\mathrm{E}_{\mathcal{B}}} \wedge b = 1\right]\right) \\
&\overset{(3)}{=} \frac{1}{2 \cdot p_{\mathrm{des}}(\lambda)} \cdot \text{Adv-IND-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B}}(\lambda, \mathrm{des}) + \frac{1}{2} \ .
\end{aligned}$$

We deduce that for every simulator $\mathcal{B}'$ of $\mathcal{B}$ with respect to des it holds (we explicitly use the fact that $\Pr[\mathrm{BadQuery}] = 0$ in $\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B}}(\lambda, \mathrm{des})$):

$$\begin{aligned}
&2 \cdot \text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B},\mathcal{B}'}(\lambda, \mathrm{des}) \\
&\overset{(2)}{=} 2 \cdot p_{\mathrm{des}}(\lambda) \cdot \Big(\Pr\left[\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B}}(\lambda, \mathrm{des}) = 1 \mid \overline{\mathrm{E}_{\mathcal{B}}}\right] \\
&\qquad - \Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{B}'}(\lambda, \mathrm{des}) = 1 \mid \overline{\mathrm{E}_{\mathcal{B}'}}\right]\Big) \\
&= \text{Adv-IND-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B}}(\lambda, \mathrm{des}) \ .
\end{aligned}$$

Using this result we argue by contradiction. Assume that Π is not IND-ATK-secure. That is, there exists $\widehat{\mathrm{des}} \in \Omega$ and a ppt adversary $\hat{\mathcal{A}} = \left(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2\right) \in \mathbf{A}^{\mathrm{IND}}$ which satisfies the restrictions defined in Corollary 3.1 such that the advantage $\text{Adv-IND-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}}}\left(\lambda, \widehat{\mathrm{des}}\right)$ is non-negligible. Then, $\hat{\mathcal{B}} \in \mathbf{A}^{\mathrm{SS}}$ constructed as above from $\hat{\mathcal{A}}$ has non-negligible advantage under attack scenario ATK. In particular, for every simulator $\hat{\mathcal{B}}'$ of $\hat{\mathcal{B}}$ with respect to $\widehat{\mathrm{des}}$ it holds

$$\text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{B}},\hat{\mathcal{B}}'}\left(\lambda, \widehat{\mathrm{des}}\right) \geq \frac{1}{2} \cdot \text{Adv-IND-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}}}\left(\lambda, \widehat{\mathrm{des}}\right) \ .$$

Hence, $\text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{B}},\hat{\mathcal{B}}'}\left(\lambda, \widehat{\mathrm{des}}\right)$ is non-negligible and this finally proves the lemma. □

B Adaptive Semantic Security Template

Definition B.1. *Let* ATK $\in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$. *A* PE *scheme* Π *for predicate family* $\mathcal{R}_{\Omega,\Sigma}$ *is called **(adaptively) semantically secure** under attack scenario* ATK *(also* aSS-ATK*-secure) if for every* $\mathrm{des} \in \Omega$ *and for every ppt algorithm* $\mathcal{A} \in \mathbf{A}^{\mathrm{aSS}}$ *there exists a ppt algorithm* $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ *such that it holds:*

1. *It holds* $\Pr[\mathcal{A}'_1 \text{ outputs } \perp] = \Pr[\mathcal{A}_1 \text{ outputs } \perp]$ *and the distributions of triples* $\left(\hat{\mathcal{M}}, h, f\right)$ *in* $\text{aSS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des})$ *and in* $\text{aSS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \mathrm{des})$ *are computationally indistinguishable.*
2. *The advantage of* $\mathcal{A}$ *defined by*

$$\text{Adv-aSS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}(\lambda, \mathrm{des}) := \Pr\left[\text{aSS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}(\lambda, \mathrm{des}) = 1\right] - \Pr\left[\text{aSS-PE-Sim}_{\Pi,\mathcal{A}'}(\lambda, \mathrm{des}) = 1\right]$$

is a negligible (in λ*) function.*

Proof. (Proof of Theorem 3.2) The adaptive semantic security trivially implies the semantic security, since every adversary in the adaptive security experiment can easily be adopted to a valid adversary for the semantic security experiment and would have the same success probability. The target function f can be just passed from $\mathcal{A}_1$ to $\mathcal{A}_2$ using St. The other direction will be formally proven next.

Let $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$ be arbitrary. Assume that there exists a predicate encryption scheme Π which is SS-ATK-secure, but is not adaptively SS-ATK-secure. From the latter we deduce that there exists an adversary $\hat{\mathcal{A}} \in \mathbf{A}^{\mathrm{aSS}}$ with non-negligible advantage. Hence, there exists $\widehat{\mathrm{des}} \in \Omega$ such that for every simulator $\hat{\mathcal{A}}'$ of $\hat{\mathcal{A}}$ with respect to $\widehat{\mathrm{des}}$ the advantage $\text{Adv-aSS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}},\hat{\mathcal{A}}'}\left(\lambda, \widehat{\mathrm{des}}\right)$ is not-negligible. Given such an $\hat{\mathcal{A}}$, we construct an adversary $\mathcal{A} \in \mathbf{A}^{\mathrm{SS}}$ with non-negligible advantage. In particular, $\mathcal{A}$ will be such that for every simulator $\mathcal{A}'$ of $\mathcal{A}$ with respect to $\widehat{\mathrm{des}}$ the advantage $\text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}\left(\lambda, \widehat{\mathrm{des}}\right)$ is non-negligible.

Let $\hat{\mathcal{A}}$ and $\widehat{\mathrm{des}}$ be as described above. Let E be the event that $\hat{\mathcal{A}}_1$ outputs $\perp$ and $p_{\widehat{\mathrm{des}}}(\lambda) = \Pr\left[\overline{\mathrm{E}}\right]$ be the probability of $\overline{\mathrm{E}}$ in $\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}}}\left(\lambda, \widehat{\mathrm{des}}\right)$. We deduce that $p_{\widehat{\mathrm{des}}}(\lambda)$ is non-negligible, which will be later used in the proof. First of all, consider the following simulator $\hat{\mathcal{A}}'$ of $\hat{\mathcal{A}}$ with respect to $\widehat{\mathrm{des}}$:

```
$\hat{\mathcal{A}}'_1\left(1^\lambda, \widehat{\mathrm{des}}\right)$:

  – Compute $(\mathrm{msk}, \mathrm{pp}_\kappa) \leftarrow \mathrm{Setup}\left(1^\lambda, \widehat{\mathrm{des}}\right)$.
  – Simulate $\left(\mathrm{cInd}^*, \left(\hat{\mathcal{M}}, h\right), St\right) \leftarrow \hat{\mathcal{A}}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa, \mathrm{msk}, \cdot)}\left(1^\lambda, \mathrm{pp}_\kappa\right)$ using $(\mathrm{msk}, \mathrm{pp}_\kappa)$.
  – Output $\perp$ if $\mathcal{A}_1$ outputs $\perp$.
  – Set $St' := (St, \mathrm{msk}, \mathrm{pp}_\kappa, \mathrm{cInd}^*)$ and output $\left(\mathrm{cInd}^*, \left(\hat{\mathcal{M}}, h\right), St'\right)$.

$\hat{\mathcal{A}}'_2\left(1^{|\hat{m}|}, h(\hat{m}), St'\right)$ with $St' = (St, \mathrm{msk}, \mathrm{pp}_\kappa, \mathrm{cInd}^*)$:

  – Compute $\mathrm{CT}' := \mathrm{Enc}\left(\mathrm{pp}_\kappa, \mathrm{cInd}^*, 1^{|\hat{m}|}\right)$.
  – Simulate $(f, \nu) \leftarrow \hat{\mathcal{A}}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa, \mathrm{msk}, \cdot)}\left(\mathrm{CT}', |\hat{m}|, h(\hat{m}), St\right)$ using $(\mathrm{msk}, \mathrm{pp}_\kappa)$.
  – Output the output of $\hat{\mathcal{A}}_2$.
```

We have to show that $\hat{\mathcal{A}}'$ is indeed a simulator of $\hat{\mathcal{A}}$ with respect to $\widehat{\mathrm{des}}$. By construction $\hat{\mathcal{A}}'_1$ outputs $\perp$ if and only if $\hat{\mathcal{A}}'$ outputs $\perp$. It remains to show that $\left(\hat{\mathcal{M}}, h, f\right)$ in the real experiment with $\hat{\mathcal{A}}$ is indistinguishable from $\left(\hat{\mathcal{M}}, h, f\right)$ in the simulated experiment with $\hat{\mathcal{A}}'$. It is important to note that the distribution of the first two elements is identical in both experiments, since $\hat{\mathcal{A}}'$ simulates $\hat{\mathcal{A}}$ perfectly using correctly generated public parameter and the master secret key. Furthermore, $\hat{\mathcal{A}}'_2$ simulates $\hat{\mathcal{A}}$ using CT' generated by $\mathrm{Enc}\left(\mathrm{pp}_\kappa, \mathrm{cInd}^*, 1^{|\hat{m}|}\right)$ instead of $\mathrm{CT}^* = \mathrm{Enc}(\mathrm{pp}_\kappa, \mathrm{cInd}^*, \hat{m})$.

Assume that there is a ppt distinguisher $\mathcal{D}$ for the triples $\left(\hat{\mathcal{M}}, h, f\right)$ generated in the real experiment $\text{aSS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}}}\left(\lambda, \widehat{\mathrm{des}}\right)$ and in the experiment $\text{aSS-PE-Sim}_{\Pi,\hat{\mathcal{A}}'}\left(\lambda, \widehat{\mathrm{des}}\right)$. More specifically, let $\varepsilon(\lambda)$ be the non-negligible advantage of $\mathcal{D}$: [2]

$$\begin{aligned}\varepsilon(\lambda) := \Pr&\left[\mathcal{D}\left(\hat{\mathcal{M}}, h, f\right) = 1 \,\middle|\, \overline{\mathrm{E}_{\hat{\mathcal{A}}}} \wedge \mathrm{CT}^* = \mathrm{Enc}\left(\mathrm{pp}_\kappa, \mathrm{cInd}^*, 1^{|\hat{m}|}\right)\right] \\ &- \Pr\left[\mathcal{D}\left(\hat{\mathcal{M}}, h, f\right) = 1 \,\middle|\, \overline{\mathrm{E}_{\hat{\mathcal{A}}}} \wedge \mathrm{CT}^* = \mathrm{Enc}(\mathrm{pp}_\kappa, \mathrm{cInd}^*, \hat{m})\right] \quad .\end{aligned}$$

We will show that this contradicts the security guaranties of Π. Namely, we construct an adversary $\mathcal{B} \in \mathbf{A}^{\mathrm{SS}}$ with non-negligible advantage under the assumption that $\varepsilon(\lambda)$ is non-negligible. $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{A}^{\mathrm{SS}}$ is as follows:

[2] If $-\varepsilon(\lambda)$ is non-negligible, just invert the output of $\mathcal{D}$ in the contraction of $\mathcal{B}$.

$\mathcal{B}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(1^\lambda,\mathrm{pp}_\kappa\right)$:

- Simulate $\left(\mathrm{cInd}^*,\left(\hat{\mathcal{M}},h\right),St\right)\leftarrow\hat{\mathcal{A}}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(1^\lambda,\mathrm{pp}_\kappa\right)$ using own oracles.
- Output $\perp$ if the output of $\hat{\mathcal{A}}_1$ is $\perp$.
- Choose $\hat{m}\leftarrow\hat{\mathcal{M}}$, set $m_0:=\hat{m}$ and $m_1:=1^{|\hat{m}|}$. Construct a circuit $\hat{\mathcal{M}}'$ which corresponds to the uniform distribution on $\{m_0,m_1\}$. Compute $\hat{h}:=h\left(\hat{m}\right)$. Choose an arbitrary h' such that $h'\left(m_0\right)=h'\left(m_1\right)$. Choose an arbitrary f' such that $f'\left(m_0\right)=0$ and $f'\left(m_1\right)=1$.[a] Set $St':=\left(St,\hat{\mathcal{M}},h,\hat{h}\right)$ and output $\left(\mathrm{cInd}^*,\left(\hat{\mathcal{M}}',f',h'\right),St'\right)$.

$\mathcal{B}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(\mathrm{CT}^*,\left|\hat{m}'\right|,h\left(\hat{m}'\right),St'\right)$ with $St'=\left(St,\hat{\mathcal{M}},h,\hat{h}\right)$.

- Simulate $(f,\nu)\leftarrow\hat{\mathcal{A}}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(\mathrm{CT}^*,\left|\hat{m}'\right|,\hat{h},St\right)$ using the own oracles.[b]
- Simulate $\mathcal{D}$ on input $\left(\hat{\mathcal{M}},h,f\right)$ and output the output of $\mathcal{D}$.

[a] We assume w.l.o.g. that $m_0\neq m_1$, which implies $\hat{m}\neq 1^{|\hat{m}|}$. In the case $\hat{m}=1^{|\hat{m}|}$ we could set $f':=h'$ and output the correct value. No simulator can do better in this case. At the same time both distributions in question are the same in this case.

[b] W.l.o.g. we assume that $\hat{\mathcal{A}}_2$ does not cause the BadQuery, since this can happen only in the case m_1 is encrypted and $\mathcal{B}$ an directly win. In the proof of Lemma A.1 we already considered similar case formally.

By construction of $\mathcal{B}$ the view of any simulator $\mathcal{B}'=\left(\mathcal{B}_1',\mathcal{B}_2'\right)$ of $\mathcal{B}$ with respect to $\widehat{\mathrm{des}}$ is independent of the challenge message and hence, it holds

$$\Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{B}'}\left(\lambda,\widehat{\mathrm{des}}\right)=1\;\middle|\;\overline{\mathrm{E}_{\mathcal{B}'}}\right]=\frac{1}{2}\quad.$$

Now, let us consider the success probability of $\mathcal{B}$:

$$\begin{aligned}\Pr\left[\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B}}\left(\lambda,\widehat{\mathrm{des}}\right)=1\;\middle|\;\overline{\mathrm{E}_{\mathcal{B}}}\right] &= \Pr\left[\hat{m}'=m_1\wedge\mathcal{D}\left(\hat{\mathcal{M}},h,f\right)=f'\left(\hat{m}'\right)\;\middle|\;\overline{\mathrm{E}_{\mathcal{B}}}\right]\\ &\quad+\Pr\left[\hat{m}'=m_0\wedge\mathcal{D}\left(\hat{\mathcal{M}},h,f\right)=f'\left(\hat{m}'\right)\;\middle|\;\overline{\mathrm{E}_{\mathcal{B}}}\right]\\ &\overset{\hat{\mathcal{M}}',f'}{=} \frac{1}{2}\cdot\left(\Pr\left[\mathcal{D}\left(\hat{\mathcal{M}},h,f\right)=1\;\middle|\;\overline{\mathrm{E}'}\wedge\hat{m}'=1^{|\hat{m}|}\right]\right.\\ &\qquad\left.+\Pr\left[\mathcal{D}\left(\hat{\mathcal{M}},h,f\right)=0\;\middle|\;\overline{\mathrm{E}'}\wedge\hat{m}'=\hat{m}\right]\right)\\ &= \frac{1}{2}\cdot\left(\Pr\left[\mathcal{D}\left(\hat{\mathcal{M}},h,f\right)=1\;\middle|\;\overline{\mathrm{E}'}\wedge\hat{m}'=1^{|\hat{m}|}\right]\right.\\ &\qquad\left.1-\Pr\left[\mathcal{D}\left(\hat{\mathcal{M}},h,f\right)=1\;\middle|\;\overline{\mathrm{E}'}\wedge\hat{m}'=\hat{m}\right]\right)\\ &= \frac{1}{2}+\frac{1}{2}\cdot\varepsilon\left(\lambda\right)\quad.\end{aligned}$$

By construction it holds $\Pr\left[\overline{\mathrm{E}_{\mathcal{B}}}\right]=\Pr\left[\overline{\mathrm{E}_{\mathcal{B}}}\right]=\Pr\left[\overline{\mathrm{E}_{\hat{\mathcal{A}}}}\right]=p_{\widehat{\mathrm{des}}}\left(\lambda\right)$. Hence, all together we get

$$\begin{aligned}\text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B},\mathcal{B}'}\left(\lambda,\mathrm{des}\right) &\overset{(2)}{=} p_{\widehat{\mathrm{des}}}\left(\lambda\right)\cdot\left(\Pr\left[\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{B}}\left(\lambda,\mathrm{des}\right)=1\;\middle|\;\overline{\mathrm{E}_{\mathcal{B}}}\right]\right.\\ &\qquad\left.-\Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{B}'}\left(\lambda,\mathrm{des}\right)=1\;\middle|\;\overline{\mathrm{E}_{\mathcal{B}'}}\right]\right)\\ &= \frac{1}{2}\cdot p_{\widehat{\mathrm{des}}}\left(\lambda\right)\cdot\varepsilon\left(\lambda\right)\quad,\end{aligned}$$

where $p_{\widehat{\mathrm{des}}}\left(\lambda\right)$ and $\varepsilon\left(\lambda\right)$ are non-negligible, which contradicts the SS-ATK-security of Π. We deduce that $\hat{\mathcal{A}}'$ is a simulator of $\hat{\mathcal{A}}$ with respect to $\widehat{\mathrm{des}}$, which finalize the first part of the proof. Hence, by our assumption the advantage $\text{Adv-aSS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}},\hat{\mathcal{A}}'}\left(\lambda,\widehat{\mathrm{des}}\right)$ is not negligible.

Next, we construct $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}^{\mathrm{SS}}$ such that for every simulator $\mathcal{A}'$ of $\mathcal{A}$ with respect to $\widehat{\mathrm{des}}$ the advantage $\text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}\left(\lambda, \widehat{\mathrm{des}}\right)$ is not negligible. $\mathcal{A}$ is as follows:

$\mathcal{A}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa, \mathrm{msk}, \cdot)}\left(1^\lambda, \mathrm{pp}_\kappa\right)$:

- Simulate $\left(\mathrm{cInd}^*, \left(\hat{\mathcal{M}}, h\right), St\right) \leftarrow \hat{\mathcal{A}}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa, \mathrm{msk}, \cdot)}\left(1^\lambda, \mathrm{pp}_\kappa\right)$ using own oracles.
- Output $\perp$ if the output of $\hat{\mathcal{A}}_1$ is $\perp$.
- Choose $\hat{m} \leftarrow \hat{\mathcal{M}}$, set $m_0 := \hat{m}$ and $m_1 := 1^{|\hat{m}|}$. Construct a circuit $\hat{\mathcal{M}}'$ which corresponds to the uniform distribution on $\{m_0, m_1\}$. Choose an arbitrary h' such that $h'(m_0) = h'(m_1)$. Choose an arbitrary f' such that $f'(m_0) = 0$ and $f'(m_1) = 1$.[a] Compute $\hat{h} := h(\hat{m})$, set $St' := \left(St, \hat{h}, \hat{m}\right)$ and output $\left(\mathrm{cInd}^*, \left(\hat{\mathcal{M}}', f', h'\right), St'\right)$.

$\mathcal{A}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa, \mathrm{msk}, \cdot)}\left(\mathrm{CT}^*, |\hat{m}'|, h(\hat{m}'), St'\right)$ with $St' = \left(St, \hat{h}, \hat{m}\right)$

- Simulate $(f, \nu) \leftarrow \hat{\mathcal{A}}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa, \mathrm{msk}, \cdot)}\left(\mathrm{CT}^*, |\hat{m}'|, \hat{h}, St\right)$ using own oracles.
- Output $b' := 0$ if $f(\hat{m}) = \nu$ and $b' := 1$ otherwise.

[a] For simplicity we ignore the case $m_0 = m_1$, where $\hat{\mathcal{A}}$ cannot have any advantage. In this case $\mathcal{A}_1$ just sets $f(m_0) = f(m_1) = 0$ and $\mathcal{A}_2$ outputs 0. Hence, no simulator of $\mathcal{A}$ can do better in this case.

It is easy to verify that $\mathcal{A} \in \mathbf{A}^{\mathrm{SS}}$. By construction, also for every simulator $\mathcal{A}'$ of $\mathcal{A}$ with respect to $\widehat{\mathrm{des}}$ it holds

$$\Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}\left(\lambda, \widehat{\mathrm{des}}\right) = 1 \,\middle|\, \overline{\mathrm{E}_{\mathcal{A}'}}\right] = \frac{1}{2} \ .$$

Consider the success probability of $\mathcal{A}$. Let $\mathrm{E}_{\mathcal{A}}$ be the event that $\mathcal{A}_1$ outputs $\perp$, which implies $\Pr\left[\overline{\mathrm{E}_{\mathcal{A}}}\right] = p_{\widehat{\mathrm{des}}}(\lambda)$. Analogously to the previous analyzes it holds

$$\begin{aligned}
\Pr\left[\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}\left(\lambda, \widehat{\mathrm{des}}\right) = 1 \,\middle|\, \overline{\mathrm{E}_{\mathcal{A}}}\right] &= \Pr\left[\hat{m}' = m_0 \wedge b' = f'(\hat{m}') \,\middle|\, \overline{\mathrm{E}_{\mathcal{A}}}\right] + \Pr\left[\hat{m}' = m_1 \wedge b' = f'(\hat{m}') \,\middle|\, \overline{\mathrm{E}_{\mathcal{A}}}\right] \\
&\overset{f,\hat{M}}{=} \frac{1}{2} \cdot \left(\Pr\left[b' = 0 \,\middle|\, \hat{m}' = m_0 \wedge \overline{\mathrm{E}_{\mathcal{A}}}\right] + \Pr\left[b' = 1 \,\middle|\, \hat{m}' = m_1 \wedge \overline{\mathrm{E}_{\mathcal{A}}}\right]\right) \\
&= \frac{1}{2} \cdot \Big(\Pr\left[f(\hat{m}) = \nu \,\middle|\, \mathrm{CT}^* = \mathrm{Enc}\left(\mathrm{pp}_\kappa, \mathrm{cInd}^*, \hat{m}\right) \wedge \overline{\mathrm{E}_{\mathcal{A}}}\right] \\
&\qquad + 1 - \Pr\left[f(\hat{m}) = \nu \,\middle|\, \mathrm{CT}^* = \mathrm{Enc}\left(\mathrm{pp}_\kappa, \mathrm{cInd}^*, 1^{|\hat{m}|}\right) \wedge \overline{\mathrm{E}_{\mathcal{A}}}\right]\Big) \\
&\overset{(*)}{=} \frac{1}{2} + \frac{1}{2} \cdot \Big(\Pr\left[\text{aSS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}}}\left(\lambda, \widehat{\mathrm{des}}\right) = 1 \,\middle|\, \overline{\mathrm{E}_{\hat{\mathcal{A}}}}\right] \\
&\qquad - \Pr\left[\text{aSS-PE-Sim}_{\Pi,\hat{\mathcal{A}}'}\left(\lambda, \widehat{\mathrm{des}}\right) = 1 \,\middle|\, \overline{\mathrm{E}_{\hat{\mathcal{A}}'}}\right]\Big) \\
&= \frac{1}{2} + \frac{1}{2 \cdot p_{\widehat{\mathrm{des}}}(\lambda)} \cdot \text{Adv-aSS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}},\hat{\mathcal{A}}'}\left(\lambda, \widehat{\mathrm{des}}\right) \ ,
\end{aligned}$$

where in equation $(*)$ we switch from the probability distribution defines by $\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}\left(\lambda, \widehat{\mathrm{des}}\right)$ to the probability distribution defined by $\text{aSS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}}}\left(\lambda, \widehat{\mathrm{des}}\right)$ conditioned on different computations of the challenge ciphertext CT^*. The last equation can be proved similarly to (2).

All together, for every simulator $\mathcal{A}'$ of $\mathcal{A}$ with respect to $\widehat{\mathrm{des}}$ it holds:

$$\begin{aligned}
\text{Adv-SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A},\mathcal{A}'}\left(\lambda, \widehat{\mathrm{des}}\right) &\overset{(2)}{=} p_{\widehat{\mathrm{des}}}(\lambda) \cdot \Big(\Pr\left[\text{SS-PE}^{\mathrm{ATK}}_{\Pi,\mathcal{A}}\left(\lambda, \widehat{\mathrm{des}}\right) = 1 \,\middle|\, \overline{\mathrm{E}_{\mathcal{A}}}\right] \\
&\qquad - \Pr\left[\text{SS-PE-Sim}_{\Pi,\mathcal{A}'}\left(\lambda, \widehat{\mathrm{des}}\right) = 1 \,\middle|\, \overline{\mathrm{E}_{\mathcal{A}'}}\right]\Big) \\
&= \frac{1}{2} \cdot \text{Adv-aSS-PE}^{\mathrm{ATK}}_{\Pi,\hat{\mathcal{A}},\hat{\mathcal{A}}'}\left(\lambda, \widehat{\mathrm{des}}\right)
\end{aligned}$$

This contradicts our precondition, that the advantage of $\mathcal{A}$ is negligible. Hence, the statement of the theorem holds. □

C Further proofs

Proof. (Proof of Lemma 3.1) Let Π, ATK be as in the lemma and $\mathcal{B} \in \mathbf{A}_{\text{P-KEM}}$ be arbitrary. We construct an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \mathbf{A}_{\text{P-KEM}}$ which simulates $\mathcal{B}$, has the same advantage but never cause the event BadQuery. $\mathcal{A}$ is as follows:

$\mathcal{A}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(1^\lambda, \mathrm{pp}_\kappa\right)$:
Simulate $(\mathrm{cInd}^*, St) \leftarrow \mathcal{B}_1^{\mathbf{O}_1(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}\left(1^\lambda, \mathrm{pp}_\kappa\right)$ using own oracles in order to answer the queries of $\mathcal{B}_1$. Store all corrupted key indices in the set S_{ck}. Furthermore, answer the queries with respect to the following rules:

- Return $\perp$ without querying the own oracle if the input of the queries in not syntactically correct:
  - kInd submitted to the key generation oracle must satisfy $\mathrm{kInd} \in \mathbb{X}_\kappa$.
  - CT submitted to the decapsulation oracle must satisfy $\mathrm{CT} \in \mathbb{C}_{\mathrm{pp}_\kappa}$. That is, there must be $\mathrm{cInd} \in \mathbb{Y}_\kappa$ such that $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}}$.
  - If $\mathrm{CT} \in \mathbb{C}_{\mathrm{cInd}}$ is submitted to the decapsulation oracle, the key index kInd related to this query must satisfy $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}) = 1$.
  - Any further syntactical restrictions which can be efficiently verified.

Output $\perp$ if the output of $\mathcal{B}_1$ is $\perp$.
Output $\perp$ if there is $\mathrm{kInd} \in S_{ck}$ such that $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}^*) = 1$. Otherwise output (cInd^*, St).

$\mathcal{A}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}(\mathrm{K}^*, \mathrm{CT}^*, St)$
Simulate $b' \leftarrow \mathcal{B}_2^{\mathbf{O}_2(\mathrm{pp}_\kappa,\mathrm{msk},\cdot)}(\mathrm{K}^*, \mathrm{CT}^*, St)$ using the own oracles in order to answer the queries of $\mathcal{B}_2$ with respect to the following rules:

- If the input of the queries in not syntactically correct perform as $\mathcal{A}_1$.
- If $\mathcal{B}_2$ queries the key generation oracle for $\mathrm{kInd} \in \mathbb{X}_\kappa$ such that $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}^*) = 1$ abort the simulation and output a guess $b' \leftarrow \{0,1\}$.
- (Only for ATK = CCA2) If $\mathcal{B}_2$ queries the decryption of CT^* (and the key index kInd specified for this query satisfies $\mathrm{R}_\kappa(\mathrm{kInd}, \mathrm{cInd}^*) = 1$ [a]) abort the simulation and output a guess $b' \leftarrow \{0,1\}$.

Output b'.

[a] Already ensured by syntactical checks.

By construction it holds $\mathcal{A} \in \mathbf{A}_{\text{P-KEM}}$, all queries of $\mathcal{A}$ are syntactically correct, and for every $\mathrm{des} \in \Omega$ it holds $\Pr[\mathrm{BadQuery}] = 0$ in $\text{P-KEM}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda, \mathrm{des})$. Note that $\mathcal{A}$ correctly answers all queries of $\mathcal{B}$, since it returns $\perp$ as answer to the queries if and only if the inputs are syntactically incorrect and the algorithms would return the same output. Next we will analyze the advantage of $\mathcal{A}$.

Let BD be the event in the experiment $\text{P-KEM}_{\Pi,\mathcal{A}}^{\mathrm{ATK}}(\lambda, \mathrm{des})$ that $\mathcal{A}_1$ outputs $\perp$ due to the illegal challenge index or $\mathcal{A}_2$ aborts the simulation of $\mathcal{B}_2$ and outputs a guess. By construction, event BD occurs if and only if $\mathcal{B}$ cause the event BadQuery in the corresponding experiment $\text{P-KEM}_{\Pi,\mathcal{B}}^{\mathrm{ATK}}(\lambda, \mathrm{des})$. Hence, it holds

$$\begin{aligned}
\text{Adv-P-KEM}_{\Pi,\mathcal{A}}^{\text{ATK}}(\lambda, \text{des}) &\overset{\text{by def.}}{=} 2 \cdot \Pr\left[\text{P-KEM}_{\Pi,\mathcal{A}}^{\text{ATK}}(\lambda, \text{des}) = 1\right] - 1 \\
&= 2 \cdot \Big(\Pr\left[\text{P-KEM}_{\Pi,\mathcal{A}}^{\text{ATK}}(\lambda, \text{des}) = 1 \wedge \overline{\text{BD}}\right] \\
&\quad + \Pr\left[\text{P-KEM}_{\Pi,\mathcal{A}}^{\text{ATK}}(\lambda, \text{des}) = 1 \wedge \text{BD}\right]\Big) - 1 \\
&\overset{(*)}{\geq} 2 \cdot \left(\Pr\left[\text{P-KEM}_{\Pi,\mathcal{B}}^{\text{ATK}}(\lambda, \text{des}) = 1 \wedge \overline{\text{BadQuery}}\right] + 0\right) - 1 \\
&= 2 \cdot \Pr\left[\text{P-KEM}_{\Pi,\mathcal{B}}^{\text{ATK}}(\lambda, \text{des}) = 1\right] - 1 \\
&\overset{\text{by def.}}{=} \text{Adv-P-KEM}_{\Pi,\mathcal{B}}^{\text{ATK}}(\lambda, \text{des}) \quad ,
\end{aligned}$$

where in the inequality $(*)$ we ignore the success probability in the case of BD. Furthermore, the event BD occurs if and only if the event BadQuery occurs and for all des $\in \mathit{\Omega}$ the view of $\mathcal{B}$ is as defined in the experiment $\text{P-KEM}_{\Pi,\mathcal{B}}^{\text{ATK}}(\lambda, \text{des})$ if $\mathcal{A}$ does not abort the simulation. The penultimate equation holds since $\text{P-KEM}_{\Pi,\mathcal{B}}^{\text{ATK}}(\lambda, \text{des}) = 1$ implies that the event BadQuery does not occur. □