Constructing and Understanding Chosen Ciphertext Security via Puncturable Key Encapsulation Mechanisms

Takahiro Matsuda and Goichiro Hanaoka

Research Institute for Secure Systems (RISEC),
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{t-matsuda,hanaoka-goichiro}@aist.go.jp

Abstract. In this paper, we introduce and study a new cryptographic primitive that we call *puncturable key encapsulation mechanism* (PKEM), which is a special class of KEMs that satisfy some functional and security requirements that, combined together, imply chosen ciphertext security (CCA security). The purpose of introducing this primitive is to capture certain common patterns in the security proofs of the several existing CCA secure public key encryption (PKE) schemes and KEMs based on general cryptographic primitives which (explicitly or implicitly) use the ideas and techniques of the Dolev-Dwork-Naor (DDN) construction (STOC'91), and "break down" the proofs into smaller steps, so that each small step is easier to work with/verify/understand than directly tackling CCA security.

To see the usefulness of PKEM, we show (1) how several existing constructions of CCA secure PKE/KEM constructed based on general cryptographic primitives can be captured as a PKEM, which enables us to understand these constructions via a unified framework, (2) their connection to detectable CCA security (Hohenberger et al. EUROCRYPT'12), and (3) a new security proof for a KEM-analogue of the DDN construction from a set of assumptions: *sender non-committing encryption* (SNCE) and non-interactive witness indistinguishable proofs.

Then, as our main technical result, we show how to construct a PKEM satisfying our requirements (and thus a CCA secure KEM) from a new set of general cryptographic primitives: *SNCE* and *symmetric key encryption secure for key-dependent messages* (KDM secure SKE). Our construction realizes the "decrypt-then-re-encrypt"-style validity check of a ciphertext which is powerful but in general has a problem of the circularity between a plaintext and a randomness. We show how SNCE and KDM secure SKE can be used together to overcome the circularity. We believe that the connection among three seemingly unrelated notions of encryption primitives, i.e. CCA security, the sender non-committing property, and KDM security, to be of theoretical interest.

Keywords: public key encryption, puncturable key encapsulation mechanism, chosen ciphertext security, sender non-committing encryption, key-dependent message secure symmetric-key encryption.

1 Introduction

In this paper, we continue a long line of work studying the constructions of public key encryption (PKE) schemes and its closely related primitive called *key encapsulation*

Y. Dodis and J.B. Nielsen (Eds.): TCC 2015, Part I, LNCS 9014, pp. 561–590, 2015.

mechanism (KEM) that are secure against chosen ciphertext attacks (CCA) [53,57,24] from general cryptographic primitives. CCA secure PKE/KEM is one of the most important cryptographic primitives that has been intensively studied in the literature, due to not only its implication to strong and useful security notions such as non-malleability [24] and universal composability [16], but also its resilience and robustness against practical attacks such as Bleichenbacher's attack [12].

There have been a number of works that show CCA secure PKE/KEMs from general cryptographic primitives: These include trapdoor permutations [24,30,31] (with some enhanced property [32]), identity-based encryption [19] and a weaker primitive called tag-based encryption [43,40], lossy trapdoor function [56] and trapdoor functions with weaker functionality/security properties [59,49,41,61], PKE with weaker than but close to CCA security [38,42,21], a combination of chosen plaintext secure (CPA secure) PKE and a hash function with some strong security [48], and techniques from program obfuscation [60,47].

One of the ultimate goals of this line of researches is to clarify whether one can construct CCA secure PKE only from CPA secure one (and in fact, a partial negative result is known [29]). This problem is important from both theoretical and practical points of view. To obtain insights into this problem, clarifying new classes of primitives that serve as building blocks is considered to be important, because those new class of primitives can be a new target that we can try constructing from CPA secure PKE schemes (or other standard primitives such as one-way injective trapdoor functions and permutations).

Our Motivation. Although differing in details, the existing constructions of CCA secure PKE schemes and KEMs from general cryptographic primitives [24,56,59,61,47,48,21] often employ the ideas and techniques of the Dolev-Dwork-Naor (DDN) construction [24], which is the first construction of CCA secure PKE from general primitives. The security proofs of these constructions are thus similar in a large sense, and it is highly likely that not a few future attempts to constructing CCA secure PKE/KEMs from general cryptographic primitives will also follow the DDN-style construction and security proof. Therefore, it will be useful and helpful for future research and also for understanding the existing works of this research direction if we can extract and abstract the common ideas and techniques behind the security proofs of the original DDN and the existing DDN-like constructions, and formalize them as a cryptographic primitive with a few formal functionality and security requirements (rather than heuristic ideas and techniques), so that most of the existing DDN-style constructions as well as potential future constructions are captured/explained/understood in a unified way, and in particular these are more accessible and easier-to-understand.

Our Contributions. Based on the motivation mentioned above, in this paper, we introduce and study a new cryptographic primitive that we call *puncturable key encapsulation mechanism* (PKEM). This is a class of KEMs that has two kinds of decryption procedures, and it is required to satisfy three simple security requirements, *decapsulation soundness*, *punctured decapsulation soundness*, and *extended CPA security* which we show in Section 3.3 that, combined together, implies CCA security. The intuition of these security notions as well as their formal definitions are explained in Section 3.2. The purpose of introducing this primitive is to capture certain common patterns in the

security proofs of the several existing CCA secure PKE schemes and KEMs based on general cryptographic primitives which (explicitly or implicitly) use the ideas and techniques of the DDN construction [24], and "break down" the proofs into smaller steps, so that each small step is easier to work with/verify/understand than directly tackling CCA security. Our formalization of PKEM is inspired (and in some sense can be seen an extension of) the notion of *puncturable tag-based encryption* [48] (which is in turn inspired by the notion of *puncturable pseudorandom function* [60]), and we explain the difference from [48] in the paragraph "*Related Work*" below.

To see the usefulness of our framework of PKEM, we show (1) how the KEM-analogue of the original DDN [24] and several existing DDN-like constructions (e.g. [56,59,61,47,48]) can be understood as a PKEM in Section 3.4, (2) its connection to detectable CCA security which is a weaker security notion than CCA security introduced by Hohenberger et al. [38] in Section 3.5, and (3) a new security proof for a KEM-analogue of the DDN construction from a set of assumptions that are different from the one used in its known security proof: *sender non-committing encryption* (SNCE, see below) and non-interactive witness indistinguishable proofs. (For the purpose of exposition, this last result is shown in Section 5.)

Then, as our main technical result, in Section 4 we show how to construct a PKEM satisfying our requirements (and thus a CCA secure KEM) from a new set of general cryptographic primitives: *SNCE* and *symmetric key encryption secure for key-dependent messages* (KDM secure SKE) [11]. Roughly speaking, a SNCE scheme is a special case of non-committing encryption [18] and is a PKE scheme which is secure even if the sender's randomness used to generate the challenge ciphertext is corrupted by an adversary. See Section 2.1 where we define SNCE formally, explain the difference among related primitives, and how it can be realized from the standard cryptographic assumptions such as the decisional Diffie-Hellman (DDH), quadratic residuosity (QR), and decisional composite residuosity (DCR). The function class with respect to which we require the building block SKE scheme to be KDM secure, is a class of efficiently computable functions whose running time is a-priori fixed. Due to Applebaum's result [1,3] (and its efficient variant [6, §7.2]) we can realize a KDM secure SKE scheme satisfying our requirement from standard assumptions such as DDH, QR, DCR. For more details on KDM secure SKE, see Section 2.2.

Our proposed PKEM has a similalriy with the "double-layered" construction of Myers and Shelat [51] and its variants [38,45,21], in which a plaintext is encrypted twice: firstly by the "inner" scheme, and secondly by "outer" scheme. Strictly speaking, however, our construction is not purely double-layered, but in some sense is closer to "hybrid encryption" of a PKE (seen as a KEM) and a SKE schemes, much similarly to the recent constructions by Matsuda and Hanaoka [47,48]. Furthermore, our construction realizes the "decrypt-then-re-encrypt"-style validity check of a ciphertext, which is a powerful approach that has been adopted in several existing constructions that construct CCA secure PKE/KEM from general cryptographic primitives [27,56,59,51,41,38,47,48,21]. In general, however, this approach has a problem of the circularity between a plaintext and a randomness, and previous works avoid such a circularity using a random oracle [27], a trapdoor function [56,59,41], a PKE scheme which achieves some security which is (weaker than but) close to CCA security [51,38,21], or a power of additional building

blocks with (seemingly very strong) security properties [47,48]. We show how SNCE and KDM secure SKE can be used together to overcome the circularity. Compared with the structurally similar constructions [38,47,48,21], the assumptions on which our construction is based could be seen weak, in the sense that the building blocks are known to be realizable from fairly standard computational assumptions such as the DDH, QR, and DCR assumptions. We believe that the connection among three seemingly unrelated notions of encryption primitives, i.e. CCA security, the sender non-committing property, and KDM security, to be of theoretical interest.

Open Problems. We believe that our framework of PKEM is useful for constructing and understanding the current and the potential future constructions of CCA secure PKE/KEMs based on the DDN-like approach, and motivates further studies on it. Our work leaves several open problems. Firstly, our framework of PKEM actually does not capture the recent construction by Dachman-Soled [21] who constructs a CCA secure PKE scheme from a PKE scheme that satisfies (standard model) plaintext awareness and some simulatability property. The construction in [21] is similar to our proposed (P)KEM in Section 4 and the recent similar constructions [47,48]. (Technically, to capture it in the language of PKEM, slight relaxations of some of the security requirements will be necessary, due to its double-layered use of PKE schemes similarly to [51].)

Secondly and perhaps more importantly, it will be worth clarifying whether it is possible to construct a PKEM satisfying our requirements only from CPA secure PKE or (an enhanced variant of) trapdoor permutations in a black-box manner. Note that a negative answer to this question will also give us interesting insights, as it shows that to construct a CCA secure PKE/KEM from these standard primitives, we have to essentially avoid the DDN-like construction.

Finally, it would also be interesting to find applications of a PKEM other than CCA secure PKE/KEMs.

Related Work. The notion of CCA security for PKE was formalized by Naor and Yung [53] and Rackoff and Simon [57]. We have already listed several existing constructions of CCA secure PKE/KEMs from general primitives in the second paragraph of Introduction. In our understanding, the works [24,56,59,61,47,48,21] are based on the ideas and techniques from the DDN construction [24].

As mentioned above, our notion of PKEM is inspired by the notion of *puncturable tag-based encryption* (PTBE) that was recently introduced by Matsuda and Hanaoka [48]. Similarly to PKEM, PTBE is a special kind of tag-based encryption [43,40] with two modes of decryption. (Roughly, in PKEM, a secret key can be punctured by a ciphertext, but in PTBE, a secret key is punctured by a tag.) Matsuda and Hanaoka [48] introduced PTBE as an abstraction of the "core" structure that appears in the original DDN construction (informally, it is the original DDN construction without a one-time signature scheme and a non-interactive zero-knowledge proof), and they use it to mainly reduce the "description complexity" of their proposed construction [48] and make it easier to understand the construction. However, they did not study it as a framework for capturing and understanding the existing DDN-style constructions (as well as potential future constructions) in a unified manner as we do in this paper. We note that Matsuda and Hanaoka [48] also formalized the security requirement called `eCPA` *security* whose formalization is a

PTBE-analogue of eCPA security for a PKEM (and thus we borrow the name). However, they did not formalize the security notions for PTBE that correspond to *decapsulation soundness* and *punctured decapsulation soundness* for a PKEM.

Paper Organization. The rest of the paper is organized as follows: In Section 2 and (in Appendix A), we review the notation and definitions of cryptographic primitives. In Section 3, we introduce and study PKEM, where in particular we show its implication to CCA security and how some of the existing constructions of KEMs can be interpreted and explained as a PKEM. In Section 4, we show our main technical result: a PKEM from SNCE and KDM secure SKE, which by the result in Section 3 yields a new CCA secure KEM from general assumptions. In Section 5, we show the CCA security of the DDN-KEM based on SNCE and non-interactive witness indistinguishable arguments.

2 Preliminaries

In this section, we give the definitions for sender non-committing encryption (SNCE) and symmetric key encryption (SKE) and its key-dependent message (KDM) security that are used in our main result in Section 4. The definitions for standard cryptographic primitives are given in Appendix A, which include PKE, KEMs, signature schemes, non-interactive argument systems, and universal one-way hash functions (UOWHFs). (The reader familiar with them need not check Appendix A at the first read, and can do so when he/she wants to check the details of the definitions.)

Basic Notation. $\mathbb{N}$ denotes the set of all natural numbers, and for $n \in \mathbb{N}$, we define $[n] := \{1, \ldots, n\}$. "$x \leftarrow y$" denotes that x is chosen uniformly at random from y if y is a finite set, x is output from y if y is a function or an algorithm, or y is assigned to x otherwise. If x and y are strings, then "$|x|$" denotes the bit-length of x, "$x\|y$" denotes the concatenation x and y, and "$(x \stackrel{?}{=} y)$" is the operation which returns 1 if $x = y$ and 0 otherwise. "PPTA" stands for a *probabilistic polynomial time algorithm*. For a finite set S, "$|S|$" denotes its size. If $\mathcal{A}$ is a probabilistic algorithm then "$y \leftarrow \mathcal{A}(x; r)$" denotes that $\mathcal{A}$ computes y as output by taking x as input and using r as randomness. $\mathcal{A}^{\mathcal{O}}$ denotes an algorithm $\mathcal{A}$ with oracle access to $\mathcal{O}$. A function $\epsilon(\cdot) : \mathbb{N} \to [0, 1]$ is said to be *negligible* if for all positive polynomials $p(\cdot)$ and all sufficiently large $k \in \mathbb{N}$, we have $\epsilon(k) < 1/p(k)$. Throughout this paper, we use the character "k" to denote a security parameter.

2.1 Sender Non-committing Public Key Encryption

Roughly, a SNCE scheme is a PKE scheme that remains secure even against an adversary who may obtain sender's randomness used to generate the challenge ciphertext. This security is ensured by requiring that there be an algorithm that generates a "fake transcript" pk and c that denote a public key and a ciphertext, respectively, so that the pair (pk, c) can be later explained as a transcript of an arbitrary message m. Our syntax of SNCE loosely follows that of sender-equivocable encryption [26,39], but departs from it because we need perfect correctness (or at least almost-all-keys-perfect correctness [25]) so that error-less decryption is guaranteed, which cannot be achieved

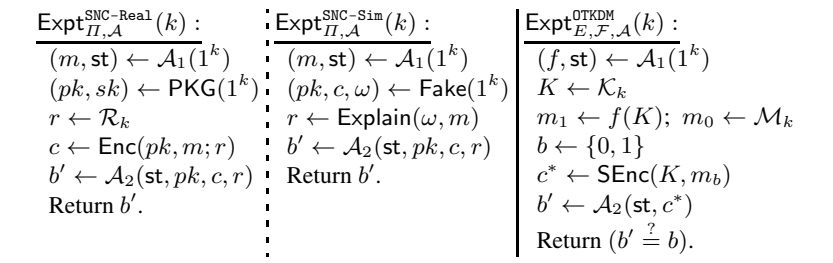

Fig. 1. Security experiments for defining the SNC security of a SNCE scheme (left and center) and that for the $\mathcal{F}$-OTKDM security of a SKE scheme (right)

by sender-equivocable encryption. We also note that recently, Hazay and Patra [35] introduced (among other notions) the notion that they call *NCE for the Sender* (NCES), which is a notion very close to SNCE we consider here. We will discuss the correctness and the difference between our definition and that of [35] later in this subsection.

Formally, a sender non-committing (public key) encryption (SNCE) scheme Π consists of the five PPTAs $(\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Explain})$ where $(\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ constitutes a PKE scheme (where definitions for ordinary PKE can be found in Appendix A), and Fake and Explain are the simulation algorithms with the following syntax:

Fake**:** This is the "fake transcript" generation algorithm that takes 1^k as input, and outputs a "fake" public key/ciphertext pair (pk, c) and a corresponding state information ω (that will be used in the next algorithm).

Explain**:** This is the (deterministic) "explanation" algorithm that takes a state information ω (where ω is computed by $(pk, c, \omega) \leftarrow \mathsf{Fake}(1^k)$) and a plaintext m as input, and outputs a randomness r that "explains" the transcript (pk, c) corresponding to ω. Namely, it is required that $\mathsf{Enc}(pk, m; r) = c$ hold.

SNC *Security.* For a SNCE scheme $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Explain})$ (where the randomness space of Enc is $\mathcal{R} = (\mathcal{R}_k)_{k\in\mathbb{N}}$) and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define the SNC-Real experiment $\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\Pi,\mathcal{A}}(k)$ and the SNC-Sim experiment $\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\Pi,\mathcal{A}}(k)$ as in Fig. 1 (left and center, respectively).

Definition 1. *We say that a SNCE scheme Π is* SNC *secure if for all PPTAs $\mathcal{A}$,* $\mathsf{Adv}^{\mathtt{SNC}}_{\Pi,\mathcal{A}}(k) := |\Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\Pi,\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\Pi,\mathcal{A}}(k) = 1]|$ *is negligible.*

The Difference among Non-committing Encryption and Related Primitives. The original definition of non-committing encryption by Canetti et al. [18] ensures security under both the sender and receiver's corruption. This is ensured by requiring that the "explaining" algorithm output not only the sender's randomness but also receiver's (i.e. randomness used to generate public/secret keys). The original definition in [18] (and several works [23,28]) allows multi-round interaction between a sender and a receiver (and even the multi-party case), but in this paper we only consider the public-key case (equivalently, the one-round two-party protocol case). A SNCE scheme is a non-committing encryption scheme that only takes care of the sender's side corruption.

Sender-equivocable encryption [26,39] is a special case of a SNCE scheme in which a sender can, under an honestly generated public key, generate a fake ciphertext that

can be later explained as an encryption of an arbitrary message (while a SNCE scheme allows that even a public key is a fake one).

Deniable encryption [17,54,10,60] has an even stronger property in which an honestly generated ciphertext under an honestly generated public key can be later explained as an encryption of an arbitrary message. For details on deniable encryption, we refer the reader to the papers [54,10].

The difference among these primitives is very important in our paper, as we explain below.

On Correctness of SNCE Schemes. In this paper, unlike most of the papers that treat (sender) non-committing encryption schemes and related primitives such as sender-equivocable encryption and deniable encryption, we require a SNCE scheme satisfy perfect correctness or at least almost-all-keys perfect correctness [25]. This is because our proposed constructions follow the Dolev-Dwork-Naor-style construction [24] which requires error-less decryption (under all but negligible fraction of key pairs) for a building block PKE scheme. Here, the non-committing property and (perfect or almost-all-keys perfect) correctness might sound contradicting. This is indeed the case for ordinary (i.e. bi-) and "receiver" non-committing encryption, sender-equivocable encryption, and deniable encryption, and thus we cannot use these primitives in our proposed constructions. However, "sender" non-committing encryption can avoid such an incompatibility, because the fake transcript generation algorithm Fake can generate (pk, c) such that pk is *not* in the range of the normal key generation algorithm PKG. Moreover, as we will see below, $\mathtt{SNC}$ secure SNCE schemes with perfect correctness (and even practical efficiency) can be realized from standard assumptions.

Concrete Instantiations of SNCE Schemes. Bellare et al. [8] formalized the notion of *lossy encryption* [8], which is a PKE scheme that has the "lossy key generation" algorithm. It outputs a "lossy public key" which is indistinguishable from a public key generated by the ordinary key generation algorithm, and an encryption under a lossy public key statistically hides the information of a plaintext. Bellare et al. [8] also introduced an additional property for lossy encryption called *efficient openability*, in which the lossy key generation algorithm outputs a trapdoor in addition to a lossy public key, and by using the trapdoor, an encryption under the lossy public key can be efficiently "explained" as a ciphertext of any plaintext.

We note that any lossy encryption with efficient openability yields a $\mathtt{SNC}$ secure SNCE scheme: the algorithm Fake generates a lossy public key pk as well as an encryption c of some plaintext, and keeps the trapdoor corresponding to pk as ω.; the algorithm $\mathsf{Explain}$ on input ω and a plaintext m outputs a randomness r that explains that $c = \mathsf{Enc}(pk, m; r)$ holds. Hence, we can use the existing lossy encryption schemes with efficient openability that are based on standard assumptions. These include the scheme based on the quadratic residuosity (QR) assumption [8, § 4.4] (which is essentially the multi-bit version of the Goldwasser-Micali scheme [33]), the scheme based on the decisional Diffie-Hellman (DDH) assumption [9, § 5.4] (which is the "bit-wise" encryption version of the DDH-based lossy encryption scheme [8, § 4.1]), and the scheme based on the decisional composite residuosity (DCR) assumption [36] (which shows that the original Paillier scheme [55] and the Damgård-Jurik scheme [22] can be extended to lossy encryption with ef-

ficient openability). In particular, the DCR-based schemes [55,22,36] have a compact ciphertext whose size does not grow linearly in the length of plaintexts.

On the Difference from the Formalization of "NCE for the Sender" in [35]. The definition of NCE for the Sender in [35] explicitly requires that the scheme have the "fake" key generation algorithm that outputs a "fake" public key together with a trapdoor, with which one can "equivocate" (or in our terminology, "explain") any ciphertext as an encryption of arbitrary plaintext m. Therefore, it seems to us that their formalization is close to lossy encryption with efficient openability [8]. On the other hand, our formalization requires that only a pair (pk, c) of public key and a ciphertext (or a "transcript" in a one-round message transmission protocol between two parties) be explained. We can construct a SNCE scheme in our formalization from NCE for the Sender of [35] (in essentially the same manner as we do so from lossy encryption with efficient openability), while we currently do not know if the converse implication can be established. Therefore, in the sense that currently an implication of only one direction is known, our formalization is weaker.

Some Useful Facts. For our result in Section 4, it is convenient to consider the so-called "repetition construction," in which a plaintext is encrypted multiple times by independently generated public keys.

More specifically, given a SNCE scheme $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Explain})$, the n-wise repetition construction $\Pi^n = (\mathsf{PKG}^n, \mathsf{Enc}^n, \mathsf{Dec}^n, \mathsf{Fake}^n, \mathsf{Explain}^n)$ is defined as follows: The key generation algorithm PKG^n runs $(pk_i, sk_i) \leftarrow \mathsf{PKG}$ for $i \in [n]$ and returns public key $PK = (pk_i)_{i\in[n]}$ and secret key $SK = (sk_i)_{i\in[n]}$.; The encryption algorithm Enc^n, on input PK and a plaintext m, runs $c_i \leftarrow \mathsf{Enc}(pk_i, m; r_i)$ for $i \in [n]$ (where each r_i is an independently chosen randomness), and outputs a ciphertext $C = (c_i)_{i\in[n]}$.; The decryption algorithm Dec^n, on input SK and C, runs $m_i \leftarrow \mathsf{Dec}(sk_i, c_i)$ for $i \in [n]$, and returns m_1 if every m_i is equal or $\perp$ otherwise.; The fake transcript generation algorithm Fake^n runs $(pk_i, c_i, \omega_i) \leftarrow \mathsf{Fake}(1^k)$ for $n \in [n]$, and returns $PK = (pk_i)_{i\in[n]}$, $C = (c_i)_{i\in[n]}$, and a state information $W = (\omega_i)_{i\in[n]}$.; The explanation algorithm $\mathsf{Explain}^n$, on input W and m, runs $r_i \leftarrow \mathsf{Explain}(\omega_i, m)$ for $i \in [n]$, and returns the randomness $R = (r_i)_{i\in[n]}$ that explains that $C = \mathsf{Enc}^n(PK, m; R)$.

By a straightforward hybrid argument, we can show that for any polynomial $n = n(k) > 0$, if the underlying scheme Π is $\mathtt{SNC}$ secure, then so is the n-wise repetition construction Π^n. (It is also a well-known fact that if Π is $\mathtt{CPA}$ secure, then so is Π^n.)

We also note that the plaintext space of an SNCE scheme can be easily extended by considering the straightforward "concatenation construction," in which plaintext $m = (m_1, \ldots, m_n)$ is encrypted block-wise by independently generated public keys.

More formal statements regarding the repetition construction and the concatenation constructions are given in the full version.

2.2 Symmetric Key Encryption

A symmetric key encryption (SKE) scheme E with key space $\mathcal{K} = \{\mathcal{K}_k\}_{k\in\mathbb{N}}$ and plaintext space $\mathcal{M} = \{\mathcal{M}_k\}_{k\in\mathbb{N}}$[1] consists of the following two PPTAs $(\mathsf{SEnc}, \mathsf{SDec})$:

[1] In this paper, for simplicity, we assume that the key space $\mathcal{K}$ and plaintext space $\mathcal{M}$ of a SKE scheme satisfy the following conditions: For each $k \in \mathbb{N}$, (1) every element in $\mathcal{K}_k$ has

SEnc**:** The encryption algorithm that takes a key $K \in \mathcal{K}_k$ and a plaintext $m \in \mathcal{M}_k$ as input, and outputs a ciphertext c.

SDec**:** The (deterministic) decryption algorithm that takes $K \in \mathcal{K}_k$ and c as input, and outputs a plaintext m which could be the special symbol $\perp$ (which indicates that c is an invalid ciphertext under K).

Correctness. We require for all $k \in \mathbb{N}$, all keys $K \in \mathcal{K}_k$, and all plaintexts $m \in \mathcal{M}_k$, it holds that $\mathsf{SDec}(K, \mathsf{SEnc}(K, m)) = m$.

One-Time Key-Dependent Message Security. Let $E = (\mathsf{SEnc}, \mathsf{SDec})$ be a SKE scheme with key space $\mathcal{K} = \{\mathcal{K}_k\}_{k\in\mathbb{N}}$ and plaintext space $\mathcal{M} = \{\mathcal{M}_k\}_{k\in\mathbb{N}}$. Let $\mathcal{F} = \{\mathcal{F}_k\}_{k\in\mathbb{N}}$ be an ensemble (which we call *function ensemble*) where for each k, $\mathcal{F}_k$ is a set of efficiently computable functions with their domain $\mathcal{K}_k$ and range $\mathcal{M}_k$.

For the SKE scheme E, the function ensemble $\mathcal{F}$, and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define the $\mathcal{F}$-$\mathtt{OTKDM}$ experiment $\mathsf{Expt}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{A}}(k)$ as in Fig. 1 (right). In the experiment, it is required that $f \in \mathcal{F}_k$.

Definition 2. *We say that a SKE scheme E is* $\mathtt{OTKDM}$ *secure with respect to $\mathcal{F}$ ($\mathcal{F}$-*$\mathtt{OTKDM}$ *secure, for short) if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{A}}(k) = 1] - 1/2|$ is negligible.*

We would like to remark that our definition of $\mathtt{OTKDM}$ security is considerably weak: it is a single instance definition that need not take into account the existence of other keys, and an adversary is allowed to make a KDM encryption query (which is captured by f) only once.

Concrete Instantiations of $\mathtt{OTKDM}$ *Secure SKE Schemes.* In our proposed construction in Section 4, the class of functions with respect to which a SKE scheme is $\mathtt{OTKDM}$ secure needs to be rich enough to be able to compute the algorithm $\mathsf{Explain}$ in a SNCE scheme multiple (an a-priori bounded number of) times. Fortunately, Applebaum [1] showed how to generically convert any SKE scheme which is many-time KDM secure (i.e. secure for many KDM encryption queries) with respect to "projections" (i.e. functions each of whose output bit depends on at most one bit of inputs) into a SKE scheme which is many-time KDM secure (and thus $\mathtt{OTKDM}$ secure), with respect to a family of functions computable in a-priori fixed polynomial time. (We can also use a more efficient construction shown by Bellare et al. [6, §7.2].) This notion is sufficient for our proposed construction. Since most SKE and PKE schemes KDM secure with respect to the class of affine functions can be interpreted as (or easily converted to) "projection"-KDM secure SKE schemes [3, §A], we can use the existing (many-time) "affine"-KDM secure SKE schemes as a building block, and apply Applebaum's conversion (or that of [6, §7.2]). Therefore, for example, one can realize a $\mathtt{OTKDM}$ secure SKE scheme with respect to fixed poly-time computable functions, based on the DDH assumption [13], the QR assumption [15], the DCR assumption [15,44], the learning with errors (LWE) assumption [4], and the learning parity with noise (LPN) assumption [4,2]. Very recently,

the same length, (2) every element in $\mathcal{M}_k$ has the same length, (3) both $\mathcal{K}_k$ and $\mathcal{M}_k$ are efficiently recognizable, and (4) we can efficiently sample a uniformly random element from both $\mathcal{K}_k$ and $\mathcal{M}_k$.

Bellare et al. [5] introduced a notion of a family of hash function called *universal computational extractor* (UCE) which is seemingly quite strong (almost random oracle-like) but a standard model assumption, and then they showed (among many other things) how to construct a SKE scheme which is non-adaptively KDM secure (in which encryption queries have to be made in parallel) with respect to any efficiently computable functions. OTKDM security is the special case of non-adaptive KDM security, and hence we can also use the result of [5] in our proposed construction.

3 Chosen Ciphertext Security from Puncturable KEMs

In this section, we introduce the notion of a *puncturable KEM* (PKEM) and show several results on it.

This section is organized as follows: In Sections 3.1 and 3.2, we define the syntax and the security requirements of a PKEM, respectively. Then in Sections 3.3 and 3.5, we show the implication of a PKEM to a CCA secure KEM and a DCCA secure detectable KEM, respectively. We also explain how a wide class of the existing constructions of CCA secure KEMs can be understood via a PKEM in Section 3.4.

3.1 Syntax

Informally, a PKEM is a KEM that has additional procedures for "puncturing secret keys according to a ciphertext" and "punctured decapsulation." In a PKEM, one can generate a "punctured" secret key $\widehat{sk}_{c^*}$ from an ordinary sk and a ciphertext c^* via the "puncturing" algorithm Punc. Intuitively, although an ordinary secret key sk defines a map (via Decap) whose domain is the whole of the ciphertext space, $\widehat{sk}_{c^*}$ only defines a map whose domain is the ciphertext space that has a "hole" produced by the puncture of the ciphertext c^*. This "punctured" secret key $\widehat{sk}_{c^*}$ can be used in the "punctured" decapsulation algorithm PDecap to decapsulate all ciphertexts that are "far" from c^* (or, those that are not in the "hole" produced by c^*), while $\widehat{sk}_{c^*}$ is useless for decapsulating ciphertexts that are "close" to c^* (or, those that are in the "hole" including c^* itself), where what it means for a ciphertext to be close to/far from c^* is decided according to a publicly computable predicate F, which is also a part of a PKEM.

Formally, a puncturable KEM consists of the six PPTAs $(\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{F}, \mathsf{Punc}, \mathsf{PDecap})$, where $(\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ constitute a KEM, and the latter three algorithms are deterministic algorithms with the following interface:

F**:** The predicate that takes a public key pk (output by $\mathsf{KKG}(1^k)$) and two ciphertexts c and c' as input, where c has to be in the range of $\mathsf{Encap}(pk)$ (but c' need not), and outputs 0 or 1.

Punc**:** The "puncturing" algorithm that takes a secret key sk (output by $\mathsf{KKG}(1^k)$) and a ciphertext c^* (output by $\mathsf{Encap}(pk)$) as input, and outputs a punctured secret key $\widehat{sk}_{c^*}$.

PDecap**:** The "punctured" decapsulation algorithm that takes a punctured secret key $\widehat{sk}_{c^*}$ (output by $\mathsf{Punc}(sk, c^*)$) and a ciphertext c as input, and outputs a session-key K which could be the special symbol $\perp$ (meaning that "c cannot be decapsulated by $\widehat{sk}_{c^*}$").

$\underline{\mathsf{Expt}^{\mathtt{DSND}}_{\Gamma,\mathcal{A}}(k)}$:
$(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
$(c^*, K^*) \leftarrow \mathsf{Encap}(pk)$
$c' \leftarrow \mathcal{A}^{\mathsf{Decap}(sk,\cdot)}(pk, c^*, K^*)$
Return 1 iff **(a)** $\wedge$ **(b)** $\wedge$ **(c)**:
(a) $\mathsf{F}(pk, c^*, c') = 1$
(b) $c' \neq c^*$
(c) $\mathsf{Decap}(sk, c') \neq \bot$

$\underline{\mathsf{Expt}^{\mathtt{PDSND}}_{\Gamma,\mathcal{A}}(k)}$:
$(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
$(c^*, K^*) \leftarrow \mathsf{Encap}(pk)$
$\widehat{sk}_{c^*} \leftarrow \mathsf{Punc}(sk, c^*)$
$c' \leftarrow \mathcal{A}^{\mathsf{PDecap}(\widehat{sk}_{c^*},\cdot)}(pk, c^*, K^*)$
Return 1 iff **(a)** $\wedge$ **(b)**:
(a) $\mathsf{F}(pk, c^*, c') = 0$
(b) $\mathsf{Decap}(sk, c') \neq \mathsf{PDecap}(\widehat{sk}_{c^*}, c')$

$\underline{\mathsf{Expt}^{\mathtt{eCPA}}_{\Gamma,\mathcal{A}}(k)}$:
$(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
$(c^*, K^*_1) \leftarrow \mathsf{Encap}(pk)$
$\widehat{sk}_{c^*} \leftarrow \mathsf{Punc}(sk, c^*)$
$K^*_0 \leftarrow \{0,1\}^k$
$b \leftarrow \{0,1\}$
$b' \leftarrow \mathcal{A}(pk, \widehat{sk}_{c^*}, c^*, K^*_b)$
Return $(b' \stackrel{?}{=} b)$.

$\underline{\mathsf{Expt}^{\mathtt{sDSND}}_{\Gamma,\mathcal{A}}(k)}$:
$(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
$(c^*, K^*) \leftarrow \mathsf{Encap}(pk)$
$c' \leftarrow \mathcal{A}(pk, sk, c^*, K^*)$
Return 1 iff **(a)** $\wedge$ **(b)** $\wedge$ **(c)**:
(a) $\mathsf{F}(pk, c^*, c') = 1$
(b) $c' \neq c^*$
(c) $\mathsf{Decap}(sk, c') \neq \bot$

$\underline{\mathsf{Expt}^{\mathtt{sPDSND}}_{\Gamma,\mathcal{A}}(k)}$:
$(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
$(c^*, K^*) \leftarrow \mathsf{Encap}(pk)$
$c' \leftarrow \mathcal{A}(pk, sk, c^*, K^*)$
Return 1 iff **(a)** $\wedge$ **(b)**:
(a) $\mathsf{F}(pk, c^*, c') = 0$
(b) $\mathsf{Decap}(sk, c') \neq \mathsf{PDecap}(\mathsf{Punc}(sk, c^*), c')$

Definitions of Advantages:
For $\mathtt{XXX} \in \{\mathtt{DSND}, \mathtt{sDSND}, \mathtt{PDSND}, \mathtt{sPDSND}\}$:
$\mathsf{Adv}^{\mathtt{XXX}}_{\Gamma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{XXX}}_{\Gamma,\mathcal{A}}(k) = 1]$
$\mathtt{eCPA}$ security:
$\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma,\mathcal{A}}(k) := 2\times |\Pr[\mathsf{Expt}^{\mathtt{eCPA}}_{\Gamma,\mathcal{A}}(k) = 1] - \frac{1}{2}|$

Fig. 2. Security experiments for a PKEM and the definition of an adversary's advantage in each experiment

The predicate F is used to define *decapsulation soundness* and *punctured decapsulation soundness*, which we explain in the next subsection. Its role is very similar to the predicate used to define $\mathtt{DCCA}$ security and unpredictability of detectable PKE in [38]. As mentioned above, intuitively, the predicate $\mathsf{F}(pk, c^*, \cdot)$ divides the ciphertext space into two classes: ciphertexts that are "close" to c^* and those that are "far" from c^*, and for each of the classes, we expect the decapsulation algorithms Decap and PDecap to work "appropriately," as we will see below.

3.2 Security Requirements

For a PKEM, we consider the three kinds of security notions: *decapsulation soundness*, *punctured decapsulation soundness*, and *extended CPA security*. The intuition for each of the security notions as well as formal definitions are explained below. Furthermore, for the first two notions, we consider two flavors: the ordinary version and the strong version (where the latter formally implies the former). We only need the ordinary notions for showing the $\mathtt{CCA}$ security of a PKEM, while the strong notions are usually easier to test/prove.

Decapsulation Soundness. This security notion is intended to capture the intuition that the only valid ciphertext which is "close" to c^* is c^* itself: It requires that given the challenge ciphertext/session-key pair (c^*, K^*), it is hard to come up with another ciphertext $c' \neq c^*$ that is (1) "close" to c^* (i.e. $\mathsf{F}(pk, c^*, c') = 1$), and (2) valid (i.e. $\mathsf{Decap}(sk, c') \neq \bot$).

Formally, for a PKEM Γ and an adversary $\mathcal{A}$, consider the decapsulation soundness ($\mathtt{DSND}$) experiment $\mathsf{Expt}^{\mathtt{DSND}}_{\Gamma,\mathcal{A}}(k)$ and the strong decapsulation soundness ($\mathtt{sDSND}$) experiment $\mathsf{Expt}^{\mathtt{sDSND}}_{\Gamma,\mathcal{A}}(k)$ defined as in Fig. 2 (left-top/bottom). The adversary $\mathcal{A}$'s advantage

in each experiment is defined as in Fig. 2 (right-bottom). Note that in the "strong" version (sDSND), an adversary is even given a secret key (which makes achieving the notion harder, but makes the interface of the adversary simpler).

Definition 3. *We say that a PKEM Γ satisfies* decapsulation soundness *(resp.* strong decapsulation soundness*) if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{DSND}}_{\Gamma,\mathcal{A}}(k)$ (resp. $\mathsf{Adv}^{\mathtt{sDSND}}_{\Gamma,\mathcal{A}}(k)$) is negligible.*

Punctured Decapsulation Soundness. This security notion is intended to capture the intuition that the "punctured" decapsulation by $\mathsf{PDecap}(\widehat{sk}_{c^*}, \cdot)$ works as good as the normal decapsulation by $\mathsf{Decap}(sk, \cdot)$ for all "far" ciphertexts c': It requires that given the challenge ciphertext/session-key pair (c^*, K^*), it is hard to come up with another ciphertext c' that is (1) "far" from c^* (i.e. $\mathsf{F}(pk, c^*, c') = 0$), and (2) the decapsulations under two algorithms $\mathsf{Decap}(sk, c')$ and $\mathsf{PDecap}(\widehat{sk}_{c^*}, c')$ disagree.

Formally, for a PKEM Γ and an adversary $\mathcal{A}$, consider the punctured decapsulation soundness (PDSND) experiment $\mathsf{Expt}^{\mathtt{PDSND}}_{\Gamma,\mathcal{A}}(k)$ and the strong punctured decapsulation soundness (sPDSND) experiment $\mathsf{Expt}^{\mathtt{sPDSND}}_{\Gamma,\mathcal{A}}(k)$ defined as in Fig. 2 (center-top/bottom). The adversary $\mathcal{A}$'s advantage in each experiment is defined as in Fig. 2 (right-bottom). Note that as in the sDSND experiment, in the "strong" version (sPDSND), an adversary is even given a secret key (which makes achieving the notion harder, but makes the interface of the adversary simpler).

Definition 4. *We say that a PKEM Γ satisfies* punctured decapsulation soundness *(resp.* strong punctured decapsulation soundness*) if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{PDSND}}_{\Gamma,\mathcal{A}}(k)$ (resp. $\mathsf{Adv}^{\mathtt{sPDSND}}_{\Gamma,\mathcal{A}}(k)$) is negligible.*

Extended CPA Security: CPA security in the presence of a punctured secret key. Extended CPA security (eCPA security, for short) requires that the CPA security hold even in the presence of the punctured secret key $\widehat{sk}_{c^*}$ corresponding to the challenge ciphertext c^*.

Formally, for a PKEM Γ and an adversary $\mathcal{A}$, consider the eCPA experiment $\mathsf{Expt}^{\mathtt{eCPA}}_{\Gamma,\mathcal{A}}(k)$ defined as in Fig. 2 (right-top). We define the advantage of an adversary as in Fig. 2 (right-bottom).

Definition 5. *We say that a PKEM Γ is* eCPA *secure if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma,\mathcal{A}}(k)$ is negligible.*

3.3 CCA Secure KEM from a Puncturable KEM

Here, we show that a PKEM satisfying all security notions introduced in Section 3.2 yields a CCA secure KEM. (The formal proof is given in the full version.)

Theorem 1. *Let $\Gamma = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{F}, \mathsf{Punc}, \mathsf{PDecap})$ be a PKEM satisfying decapsulation soundness, punctured decapsulation soundness, and* eCPA *security. Then, $\Gamma^* = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ is a* CCA *secure KEM.*

Specifically, for any PPTA $\mathcal{A}$ that attacks the CCA *security of Γ^* and makes in total $Q = Q(k) > 0$ decapsulation queries, there exist PPTAs $\mathcal{B}_{\mathsf{d}}$, $\mathcal{B}_{\mathsf{a}}$, and $\mathcal{B}_{\mathsf{e}}$ such that*

$$\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma^*,\mathcal{A}}(k) \leq 2 \cdot \mathsf{Adv}^{\mathtt{DSND}}_{\Gamma,\mathcal{B}_{\mathsf{d}}}(k) + 2Q \cdot \mathsf{Adv}^{\mathtt{PDSND}}_{\Gamma,\mathcal{B}_{\mathsf{a}}}(k) + \mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma,\mathcal{B}_{\mathsf{e}}}(k). \tag{1}$$

Proof Sketch of Theorem 1. Let $\mathcal{A}$ be any PPTA adversary that attacks the KEM Γ^* in the sense of $\mathtt{CCA}$ security. Consider the following sequence of games:

Game 1: This is the $\mathtt{CCA}$ experiment $\mathsf{Expt}^{\mathtt{CCA}}_{\Gamma^*,\mathcal{A}}(k)$ itself.

Game 2: Same as Game 1, except that all decapsulation queries c satisfying $\mathsf{F}(pk, c^*, c) = 1$ are answered with $\perp$.

Game 3: Same as Game 2, except that all decapsulation queries c satisfying $\mathsf{F}(pk, c^*, c) = 0$ are answered with $\mathsf{PDecap}(\widehat{sk}_{c^*}, c)$, where $\widehat{sk}_{c^*} = \mathsf{Punc}(sk, c^*)$.

For $i \in [3]$, let Succ_i denote the event that in Game i, $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs). We will show that $|\Pr[\mathsf{Succ}_i] - \Pr[\mathsf{Succ}_{i+1}]|$ is negligible for each $i \in [2]$ and that $|\Pr[\mathsf{Succ}_3] - 1/2|$ is negligible, which proves the theorem.

Firstly, note that Game 1 and Game 2 proceed identically unless $\mathcal{A}$ makes a decapsulation query c satisfying $\mathsf{F}(pk, c^*, c') = 1$ and $\mathsf{Decap}(sk, c) \neq \perp$, and hence $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ is upperbounded by the probability of $\mathcal{A}$ making such a query in Game 1 or Game 2. Recall that by the rule of the $\mathtt{CCA}$ experiment, $\mathcal{A}$'s queries c must satisfy $c \neq c^*$. But $\mathsf{F}(pk, c^*, c') = 1$, $c \neq c^*$, and $\mathsf{Decap}(sk, c) \neq \perp$ are exactly the conditions of violating the decapsulation soundness, and the probability of $\mathcal{A}$ making a query satisfying these conditions is negligible.

Secondly, note that Game 2 and Game 3 proceed identically unless $\mathcal{A}$ makes a decapsulation query c satisfying $\mathsf{F}(pk, c^*, c) = 0$ and $\mathsf{Decap}(sk, c) \neq \mathsf{PDecap}(\widehat{sk}_{c^*}, c)$, where $\widehat{sk}_{c^*} = \mathsf{Punc}(sk, c^*)$. Hence $|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$ is upperbounded by the probability of $\mathcal{A}$ making such a query in Game 2 or Game 3. However, since these conditions are exactly those of violating the punctured decapsulation soundness, the probability of $\mathcal{A}$ making a query satisfying the above conditions is negligible.

Finally, we can upperbound $|\Pr[\mathsf{Succ}_3] - 1/2|$ to be negligible directly by the $\mathtt{eCPA}$ security of the PKEM Γ. More specifically, any $\mathtt{eCPA}$ adversary $\mathcal{B}_\mathsf{e}$, which receives $(pk, \widehat{sk}_{c^*}, c^*, K^*_b)$ as input, can simulate Game 3 for $\mathcal{A}$, where $\mathcal{A}$'s decapsulation oracle in Game 3 is simulated perfectly by using $\widehat{sk}_{c^*}$, so that $\mathcal{B}_\mathsf{e}$'s $\mathtt{eCPA}$ advantage is exactly $2 \cdot |\Pr[\mathsf{Succ}_3] - 1/2|$. This shows that $|\Pr[\mathsf{Succ}_3] - 1/2|$ is negligible. □

On the Tightness of the Reduction. In the equation (1) of the above proof, the reason why we have the factor Q (the number of a $\mathtt{CCA}$ adversary $\mathcal{A}$'s decapsulation queries) in front of the advantage $\mathsf{Adv}^{\mathtt{PDSND}}_{\Gamma,\mathcal{B}_\mathsf{a}}(k)$ of the reduction algorithm $\mathcal{B}_\mathsf{a}$ attacking punctured decapsulation soundness, is that the reduction algorithm $\mathcal{B}_\mathsf{a}$ cannot check whether a ciphertext c' satisfies the condition (b) of violating punctured decapsulation soundness, i.e. $\mathsf{Decap}(sk, c') \neq \mathsf{PDecap}(\widehat{sk}_{c^*}, c')$, and thus $\mathcal{B}_\mathsf{a}$ picks one of $\mathcal{A}$'s decapsulation queries randomly. However, if we instead use a PKEM with *strong* punctured decapsulation soundness, then, when proving security, a reduction algorithm attacking *strong* punctured decapsulation soundness is given the secret key sk as input, which enables it to check whether the condition $\mathsf{Decap}(sk, c') \neq \mathsf{PDecap}(\widehat{sk}_{c^*}, c')$ is satisfied. Therefore, the reduction algorithm need not pick one of the decapsulation queries randomly, but can find a ciphertext c' that violates the conditions of strong punctured decapsulation soundness whenever the adversary $\mathcal{A}$ asks such a ciphertext as a decapsulation query, which leads to a tight security reduction. We will explain this in more details in the full version.

3.4 Understanding the Existing Constructions of CCA Secure KEMs via Puncturable KEM

To see the usefulness of a PKEM and the result in Section 3.3, here we demonstrate how the existing constructions of CCA secure KEMs can be understood via a PKEM.

The Dolev-Dwork-Naor KEM. We first show how a security proof of the KEM version of the DDN construction [24], which we call the *DDN-KEM*, can be understood via a PKEM. This is the KEM obtained from the original DDN construction (which is a PKE scheme) in which we encrypt a random value and regard it as a session-key.

Let $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme whose plaintext space is $\{0,1\}^k$ and whose randomness space (for security parameter k) is $\mathcal{R}_k$. Consider the NP language $L = \{L_k\}_{k\in\mathbb{N}}$ where each L_k is defined as follows:

$$L_k := \left\{ ((pk_i)_{i\in[k]}, (c_i)_{i\in[k]}) \;\middle|\; \begin{array}{r} \exists((r_i)_{i\in[k]}, K) \in (\mathcal{R}_k)^k \times \{0,1\}^k \text{ s.t.} \\ \forall i \in [k] : \mathsf{Enc}(pk_i, K;\, r_i) = c_i \end{array} \right\}.$$

Let $\mathcal{P} = (\mathsf{CRSG}, \mathsf{Prove}, \mathsf{PVer})$ be a non-interactive argument system for the language L. Moreover, let $\Sigma = (\mathsf{SKG}, \mathsf{Sign}, \mathsf{SVer})$ and $\mathcal{H} = (\mathsf{HKG}, \mathsf{H})$ be a signature scheme and a UOWHF, respectively. (The definitions of an ordinary PKE scheme, a signature scheme, a UOWHF, and a non-interactive argument system can be found in Appendix A.) Then we construct the PKEM $\Gamma_{\mathtt{DDN}} = (\mathsf{KKG}_{\mathtt{DDN}}, \mathsf{Encap}_{\mathtt{DDN}}, \mathsf{Decap}_{\mathtt{DDN}}, \mathsf{F}_{\mathtt{DDN}}, \mathsf{Punc}_{\mathtt{DDN}}, \mathsf{PDecap}_{\mathtt{DDN}})$, which is based on the DDN-KEM, as in Fig. 3. The original DDN-KEM $\Gamma^*_{\mathtt{DDN}}$ is $(\mathsf{KKG}_{\mathtt{DDN}}, \mathsf{Encap}_{\mathtt{DDN}}, \mathsf{Decap}_{\mathtt{DDN}})$.

For the PKEM $\Gamma_{\mathtt{DDN}}$, the three security requirements are shown as follows:

Lemma 1. *If $\mathcal{H}$ is a UOWHF and Σ is a* SOT *secure signature scheme, then the PKEM $\Gamma_{\mathtt{DDN}}$ satisfies strong decapsulation soundness.*

Lemma 2. *If the non-interactive argument system $\mathcal{P}$ satisfies adaptive soundness, then the PKEM $\Gamma_{\mathtt{DDN}}$ satisfies strong punctured decapsulation soundness.*

Lemma 3. *If the PKE scheme Π is* CPA *secure and the non-interactive argument system $\mathcal{P}$ is* ZK *secure, then the PKEM $\Gamma_{\mathtt{DDN}}$ is* eCPA *secure.*

The formal proofs of these lemmas are given in the full version, and here we give some intuitions below.

The first two lemmas are almost trivial. Specifically, let $C^* = (vk^*, (c^*_i)_i, \pi^*, \sigma^*)$ be the challenge ciphertext, and let $C' = (vk', (c'_i)_i, \pi', \sigma')$ be a ciphertext output by an adversary in the sDSND experiment or the sPDSND experiment (recall that the interface of an adversary in these experiments is the same). Then, a simple observation shows that if C' is a successful ciphertext that violates strong decapsulation soundness, then C' must satisfy one of the following two conditions: (1) $\mathsf{H}_\kappa(vk^*) = \mathsf{H}_\kappa(vk')$ and $vk^* \neq vk'$, or (2) $\mathsf{SVer}(vk', ((c'_i)_i, \pi'), \sigma') = \top$, $((c^*_i)_i, \pi^*, \sigma^*) \neq ((c'_i)_i, \pi', \sigma')$, and $vk^* = vk'$. However, a ciphertext with the first condition is hard to find due to the security of the UOWHF $\mathcal{H}$, and a ciphertext with the second condition is hard to find due to the SOT security of the signature scheme Σ. Similarly, again a simple observation shows that in order for C' to be a successful ciphertext that violates strong punctured

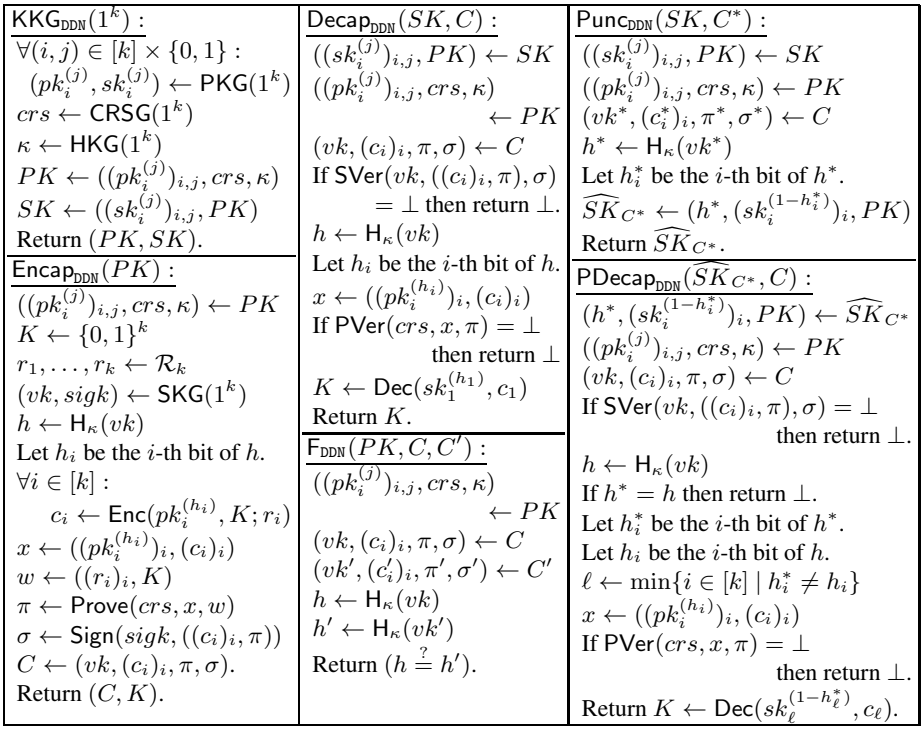

Fig. 3. The PKEM $\Gamma_{\mathtt{DDN}}$ based on a PKE scheme Π and a non-interactive argument system $\mathcal{P}$. In the figure, "$(r_i)_i$" and "$(pk_i^{(j)})_{i,j}$" are the abbreviations of "$(r_i)_{i\in[k]}$" and "$(pk_i^{(j)})_{i\in[k],j\in\{0,1\}}$", respectively, and we use a similar notation for other values.

decapsulation soundness, C' has to satisfy $\mathsf{PVer}(crs,x',\pi')=\top$ and $x'\notin L_k$ where $x'=((pk_i^{(h_i')})_i,(c_i')_i)$, and hence the adaptive soundness of the non-interactive argument system $\mathcal{P}$ guarantees that the probability that an adversary coming up with such a ciphertext in the sPDSND experiment is negligible. The eCPA security is also easy to see. Specifically, we can first consider a modified experiment in which crs and π are respectively generated by using the simulation algorithms SimCRS and SimPrv which exist by the ZK security of $\mathcal{P}$. By the ZK security, an eCPA adversary cannot notice this change. Then, the CPA security of the underlying PKE scheme directly shows that the information of a session-key does not leak, leading to the eCPA security.

Capturing Other Existing Constructions. Our framework with a PKEM can explain other existing constructions that, explicitly or implicitly, follow a similar security proof to the DDN construction. For example, the Rosen-Segev construction based on an injective trapdoor function (TDF) secure under correlated inputs [59], the Peikert-Waters construction [56] based on a lossy TDF and an all-but-one lossy TDF (ABO-TDF) in which the ABO-TDF is instantiated from a lossy TDF (see this construction in [56, §2.3]). Moreover, the construction based on CPA secure PKE and an obfuscator for point functions (with multi-bit output) by Matsuda and Hanaoka [47] and one based on

CPA secure PKE and a hash function family satisfying the strong notion (called UCE security [5]) from the same authors [48] can also be captured as a PKEM.

Furthermore, our framework with a PKEM can also capture KEMs based on *all-but-one extractable hash proof systems* (ABO-XHPS) by Wee [61] (and its extension by Matsuda and Hanaoka [46]), by introducing some additional property for underlying ABO-XHPS. Although the additional property that we need is quite subtle, it is satisfied by most existing ABO-XHPS explained in [61,46]. Since a number of recent practical CCA secure KEMs (e.g. [14,20,34,37]) are captured by the framework of ABO-XHPS, our result is also useful for understanding practical KEMs. We expand the explanation for capturing ABO-XHPS-based KEMs in the full version.

3.5 DCCA Secure Detectable KEM from a Puncturable KEM

Here, we show that even if a PKEM does not have decapsulation soundness, it still yields a DCCA secure detectable KEM [38,45]. Therefore, if a PKEM satisfying punctured decapsulation soundness and eCPA security additionally satisfies the property called *unpredictability* [38,45] it can still be used as a building block in the constructions [38,45] to obtain fully CCA secure PKE/KEM.[2]

Theorem 2. *Let* $\Gamma = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{F}, \mathsf{Punc}, \mathsf{PDecap})$ *be a PKEM satisfying punctured decapsulation soundness and* eCPA *security. Then,* $\Gamma^{\dagger} = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{F})$ *is a* DCCA *secure detectable KEM.*

Proof Sketch of Theorem 2. The proof of this theorem is straightforward given the proof of Theorem 1 (it is only simpler), and thus we omit a formal proof. The reason why we do not need decapsulation soundness is that an adversary in the DCCA experiment is not allowed to ask a decapsulation query c with $\mathsf{F}(pk, c^*, c) = 1$, and we need not care the behavior of Decap for "close" ciphertexts. Thus, as in the proof of Theorem 1, the punctured decapsulation soundness guarantees that $\mathsf{PDecap}(\widehat{sk}_{c^*}, \cdot)$ works as good as $\mathsf{Decap}(sk, \cdot)$ for all "far" ciphertexts c with $\mathsf{F}(pk, c^*, c) = 0$, and then the eCPA security guarantees the indistinguishability of a real session-key K_1^* and a random K_0^*. □

4 Puncturable KEM from Sender Non-committing Encryption and KDM Secure SKE

In this section, we show our main technical result: a PKEM that uses a SNCE scheme and a OTKDM secure SKE scheme (with respect to efficiently computable functions). By Theorem 1, this yields a CCA secure KEM. Therefore, this result clarifies a new set of general cryptographic primitives that implies CCA secure PKE/KEM.

The construction of the proposed PKEM is as follows: Let $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Explain})$ be a SNCE scheme such that the plaintext space is $\{0,1\}^n$ (for some polynomial $n = n(k) > 0$) and the randomness space of Enc is $\mathcal{R}_k$. Let $E = (\mathsf{SEnc},$

[2] We note that the DDN-KEM reviewed in Section 3.4 and our proposed KEM in Section 4 achieve strong unpredictability (based on the security of the building blocks), which we show in the full version.

SDec) be a SKE scheme whose key space and plaintext space (for security parameter k) are $\mathcal{K}_k$ and $\mathcal{M}_k$, respectively. We require $\mathcal{K}_k \subseteq \{0,1\}^n$ and $(\mathcal{R}_k)^{k+1} \times \{0,1\}^k \subseteq \mathcal{M}_k$. Furthermore, let $\mathcal{H} = (\mathsf{HKG}, \mathsf{H})$ be a hash function family (which is going to be assumed to be a UOWHF). Then we construct a PKEM $\widehat{\Gamma} = (\widehat{\mathsf{KKG}}, \widehat{\mathsf{Encap}}, \widehat{\mathsf{Decap}}, \widehat{\mathsf{F}}, \widehat{\mathsf{Punc}}, \widehat{\mathsf{PDecap}})$ as in Fig. 4.

Function Ensemble for OTKDM *Security.* For showing the eCPA security of $\widehat{\Gamma}$, we need to specify a function ensemble $\mathcal{F} = \{\mathcal{F}_k\}_{k\in\mathbb{N}}$ with respect to which E is OTKDM secure. For each $k \in \mathbb{N}$, define a set $\mathcal{F}_k$ of efficiently computable functions as follows:

$$\mathcal{F}_k := \left\{ \begin{array}{l} f_z : \mathcal{K}_k \to \mathcal{M}_k \text{ given by} \\ \quad f_z(\alpha) := ((\mathsf{Explain}(\omega_i, \alpha))_{i\in[k+1]}, K) \end{array} \middle| \begin{array}{l} z = ((\omega_i)_{i\in[k+1]}, K) \text{ where } K \in \{0,1\}^k \\ \quad \text{and each } \omega_i \text{ is output from } \mathsf{Fake}(1^k) \end{array} \right\}$$

Note that each function in $\mathcal{F}_k$ is parameterized by z, and is efficiently computable.

Security of $\widehat{\Gamma}$. The three security requirements of the PKEM $\widehat{\Gamma}$ can be shown as follows: (The formal proofs of Lemmas 4, 5, and 6 are given in Appendices B.1, B.2, and B.3, respectively.)

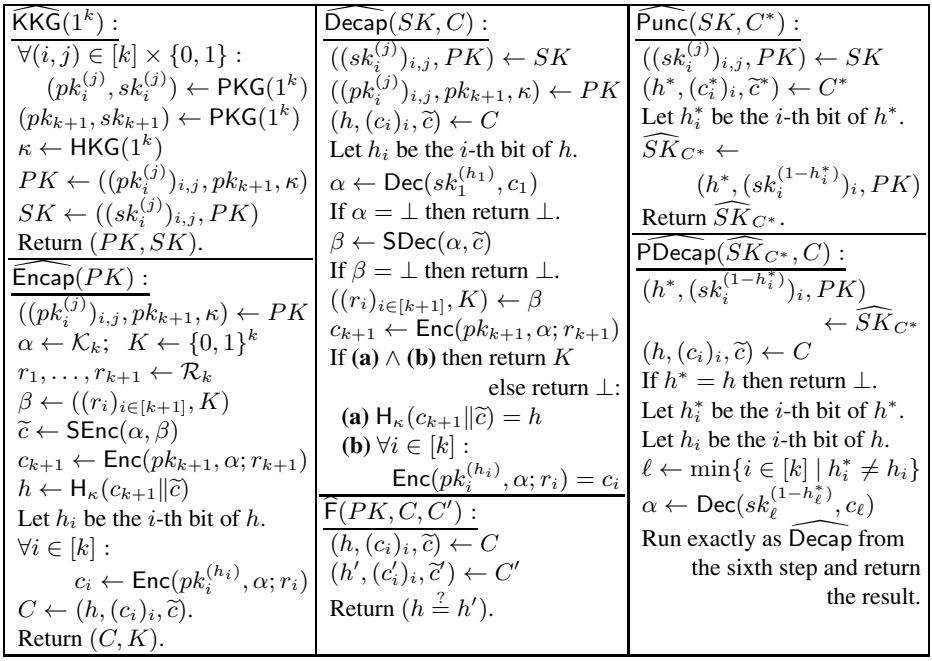

Fig. 4. The PKEM $\widehat{\Gamma}$ based on a SNCE scheme Π and a SKE scheme E. In the figure, "$(r_i)_i$" and "$(pk_i^{(j)})_{i,j}$" are the abbreviations of "$(r_i)_{i\in[k]}$" and "$(pk_i^{(j)})_{i\in[k],j\in\{0,1\}}$", respectively, and we use similar notation for other values.

Lemma 4. *If $\mathcal{H}$ is a UOWHF, then the PKEM $\widehat{\Gamma}$ satisfies strong decapsulation soundness.*

Lemma 5. *The PKEM $\widehat{\Gamma}$ satisfies strong punctured decapsulation soundness (even against computationally unbounded adversaries) unconditionally.*

Lemma 6. *If the SNCE scheme Π is* SNC *secure and the SKE scheme E is $\mathcal{F}$-*OTKDM *secure, then the PKEM $\widehat{\Gamma}$ is* eCPA *secure.*

Here, we explain high-level proof sketches for each lemma. Regarding strong decapsulation soundness (Lemma 4), recall that in the sDSND experiment, in order for a ciphertext $C' = (h', (c'_i)_i, \widetilde{c}')$ to violate (strong) decapsulation soundness, it must satisfy $\widehat{\mathsf{F}}(PK, C^*, C') = 1$ (which implies $h^* = h'$), $C' \neq C^*$, and $\widehat{\mathsf{Decap}}(SK, C') \neq \bot$, which (among other conditions) implies $h^* = \mathsf{H}_\kappa(c^*_{k+1} \| \widetilde{c}^*) = \mathsf{H}_\kappa(c'_{k+1} \| \widetilde{c}') = h'$, where the values with asterisk are those related to the challenge ciphertext $C^* = (h^*, (c^*_i)_i, \widetilde{c}^*)$ and c'_{k+1} is the intermediate value calculated during the computation of $\widehat{\mathsf{Decap}}(SK, C')$. On the other hand, a simple observation shows that the above conditions also imply another condition $(c^*_{k+1}, \widetilde{c}^*) \neq (c'_{k+1}, \widetilde{c}')$. This means that a successful ciphertext that violates (strong) decapsulation soundness leads to a collision for the UOWHF $\mathcal{H}$, which is hard to find by the security of the UOWHF $\mathcal{H}$.

Regarding punctured decapsulation soundness (Lemma 5), we show that for any (possibly invalid) ciphertext $C' = (h', (c'_i)_i, \widetilde{c}')$, if $h' \neq h^*$, then it always holds that $\widehat{\mathsf{Decap}}(SK, C') = \widehat{\mathsf{PDecap}}(\widehat{SK}_{C^*}, C')$. This can be shown due to the correctness of the building block SNCE scheme Π and the validity check by re-encryption performed at the last step of $\widehat{\mathsf{Decap}}$ and $\widehat{\mathsf{PDecap}}$. In particular, the validity check by re-encryption works like a non-interactive proof with perfect soundness in the DDN construction, and hence for any adversary, its sPDSND advantage is zero.

Finally, we explain how the eCPA security (Lemma 6) is proved. Let $\mathcal{A}$ be any eCPA adversary. Consider the following sequence of games:

Game 1: This is the eCPA experiment itself. To make it easier to define the subsequent games, we change the ordering of the operations as follows (note that this does not change $\mathcal{A}$'s view):

$\alpha^* \leftarrow \mathcal{K}_k$;
For $i \in [k+1]$:
  $(pk'_i, sk'_i) \leftarrow \mathsf{PKG}(1^k)$;
  $r^*_i \leftarrow \mathcal{R}_k$;
  $c^*_i \leftarrow \mathsf{Enc}(pk'_i, \alpha^*; r^*_i)$;
End For
$K^*_1 \leftarrow \{0,1\}^k$;
$\beta^* \leftarrow ((r^*_i)_{i \in [k+1]}, K^*_1)$;
$\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, \beta^*)$;
$\kappa \leftarrow \mathsf{HKG}(1^k)$;
$h^* = (h^*_1 \| \ldots \| h^*_k) \leftarrow \mathsf{H}_\kappa(c^*_{k+1} \| \widetilde{c}^*)$;
(Continue to the right column ↗)

For $i \in [k]$:
  $pk_i^{(h^*_i)} \leftarrow pk'_i$;
  $(pk_i^{(1-h^*_i)}, sk_i^{(1-h^*_i)}) \leftarrow \mathsf{PKG}(1^k)$;
End For
$PK \leftarrow ((pk_i^{(j)})_{i,j}, pk'_{k+1}, \kappa)$;
$C^* \leftarrow (h^*, (c^*_i)_i, \widetilde{c}^*)$;
$\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h^*_i)})_i, PK)$;
$K^*_0 \leftarrow \{0,1\}^k$;
$b \leftarrow \{0,1\}$;
$b' \leftarrow \mathcal{A}(PK, \widehat{SK}_{C^*}, C^*, K^*_b)$

Game 2: Same as Game 1, except that we generate each tuple $(pk_i^{(h_i^*)}, c_i^*, r_i^*)$ and $(pk_{k+1}, c_{k+1}^*, r_{k+1}^*)$ by using the simulation algorithms Fake and $\mathsf{Explain}$ of the SNCE scheme Π. More precisely, in this game, the step with the underline in Game 1 is replaced with: "$(pk_i', c_i^*, \omega_i^*) \leftarrow \mathsf{Fake}(1^k)$; $r_i^* \leftarrow \mathsf{Explain}(\omega_i^*, \alpha^*)$."

Game 3: Same as Game 2, except that the information of $\beta^* = ((r_i^*)_{i\in k+1}, K_1^*)$ is erased from $\widetilde{c}^*$. More precisely, in this game, the step "$\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, \beta^*)$" in Game 2 is replaced with the steps "$\beta' \leftarrow \mathcal{M}_k$; $\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, \beta')$."

For $i \in [3]$, let Succ_i be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs). We will show that $|\Pr[\mathsf{Succ}_i] - \Pr[\mathsf{Succ}_{i+1}]|$ is negligible for each $i \in [2]$, and that $\Pr[\mathsf{Succ}_3] = 1/2$, which proves the eCPA security of the PKEM $\widehat{\Gamma}$.

Firstly, we can show that $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ is negligible due to the SNC security of the $(k+1)$-repetition construction Π^{k+1}, which in turn follows from the SNC security of the underlying SNCE scheme Π by a standard hybrid argument (see the explanation in the last paragraph of Section 2.1).

Secondly, we can show that $|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$ is negligible due to the $\mathcal{F}$-OTKDM security of the SKE scheme E. Here, the key idea is that we view the plaintext $\beta^* = ((r_i^*)_{i\in[k+1]}, K_1^*) = ((\mathsf{Explain}(\omega_i^*, \alpha^*)_{i\in[k+1]}, K^*)$ which will be encrypted under the key α^* as a "key-dependent message" of the key α^*. More specifically, in the full proof we show how to construct a OTKDM adversary $\mathcal{B}_\mathsf{e}$ that uses the KDM function $f \in \mathcal{F}_k$ defined by $f(\alpha^*) = ((\mathsf{Explain}(\omega_i^*, \alpha^*)_{i\in[k+1]}, K^*)$ (where $(\omega_i^*)_{i\in[k+1]}$ and K_1^* are viewed as fixed parameters hard-coded in f) for the challenge KDM query, and depending on $\mathcal{B}_\mathsf{e}$'s challenge bit, $\mathcal{B}_\mathsf{e}$ simulates Game 2 or Game 3 perfectly for $\mathcal{A}$ so that $\mathsf{Adv}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{B}_\mathsf{e}}(k) = |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$.

Finally, observe that in Game 3, the challenge ciphertext C^* is independent of K_1^*, and the input $(PK, \widehat{SK}_{C^*}, C^*, K_b^*)$ to $\mathcal{A}$ is distributed identically for both $b \in \{0,1\}$. This implies $\Pr[\mathsf{Succ}_3] = 1/2$.

Our construction of the PKEM $\widehat{\Gamma}$, and the combination of Lemmas 4 to 6 and Theorem 1 lead to our main result in this paper:

Theorem 3. *If there exist a* SNC *secure SNCE scheme and a SKE scheme that is* OTKDM *secure with respect to efficiently computable functions, then there exist a* CCA *secure PKE scheme/KEM.*

Finally, it would be worth noting that our construction of a CCA secure PKE (via a PKEM) is black-box, in the sense that the construction uses the building blocks in a black-box manner, while our security reductions of the eCPA security is non-black-box, in the sense that our reduction algorithm needs to use the description of the Explain algorithm as a KDM encryption query. Such a situation was encountered in [50,21] where these constructions use the building block PKE scheme in a black-box manner, while the security proof (reduction) is non-black-box because they need to rely on plaintext awareness.

5 Dolev-Dwork-Naor KEM Revisited

In this section, we show that the eCPA security of the DDN-PKEM $\Gamma_{\mathtt{DDN}}$ (Fig. 5) that we reviewed in Section 3.4 can be shown from different assumptions on the PKE scheme

Π and the non-interactive argument system $\mathcal{P}$. More specifically, we show that if Π is a SNC secure SNCE scheme and $\mathcal{P}$ is WI secure, then we can still show that the PKEM $\Gamma_{\mathtt{DDN}}$ is eCPA secure. We emphasize that this change of assumptions does *not* affect the other assumptions used for decapsulation soundness and punctured decapsulation soundness, and thus we see that this result is a concrete evidence of the usefulness of "breaking down" the steps in a security proof into small separate steps. By Theorem 1, we obtain a new CCA security proof for the DDN-KEM based on a SNCE scheme and a non-interactive witness indistinguishable argument system (in the common reference string model).

We believe this new proof for the classical construction with different set of assumptions to be theoretically interesting, and another qualitative evidence of the usefulness of SNCE in the context of constructing CCA secure PKE/KEM. In particular, compared with the original DDN-KEM, our result here shows a trade-off among assumptions on building blocks: a stronger assumption on a PKE scheme and instead a weaker assumption on a non-interactive argument system. Our result shows that the difference between a CPA secure PKE scheme and a SNC secure SNCE scheme is as large/small as the difference between the ZK security and WI security of a non-interactive argument system.

Lemma 7. *If Π is a* SNC *secure SNCE scheme and the non-interactive argument system $\mathcal{P}$ is* WI *secure, then the PKEM $\Gamma_{\mathtt{DDN}}$ is* eCPA *secure.*

The formal proof is given in the full version, and here we give a proof sketch. Recall that in the proof of Lemma 3, we first use the ZK security of $\mathcal{P}$ to "cut" the relation between the components $(c_i^*)_i$ and the proof π^*, and then use the CPA security of the k-repetition construction Π^k (which in turn follows from the CPA security of Π) to "hide" the information of the challenge bit. The proof of Lemma 7 uses the properties of the building blocks in the reversed order.

Proof Sketch of Lemma 7. Let $\mathcal{A}$ be any PPTA adversary that attacks the eCPA security of $\Gamma_{\mathtt{DDN}}$. Consider the following sequence of games:

Game 1: This is the eCPA experiment itself. To make it easier to define the subsequent games, we change the ordering of the operations as follows (note that this does not change $\mathcal{A}$'s view):

$K_1^* \leftarrow \{0,1\}^k$;
For $i \in [k]$:
  $\underline{(pk'_i, sk'_i) \leftarrow \mathsf{PKG}(1^k);}$
  $\underline{r_i^* \leftarrow \mathcal{R}_k;}$
  $\underline{c_i^* \leftarrow \mathsf{Enc}(pk'_i, K_1^*; r_i^*);}$
End For
$x^* \leftarrow ((pk'_i)_i, (c_i^*)_i)$;
$w^* \leftarrow ((r_i^*)_i, K_1^*)$;
$crs \leftarrow \mathsf{CRSG}(1^k)$;
$\pi^* \leftarrow \mathsf{Prove}(crs, x^*, w^*)$;
$(vk^*, sigk^*) \leftarrow \mathsf{SKG}(1^k)$;
$\sigma^* \leftarrow \mathsf{Sign}(sigk^*, ((c_i^*)_i, \pi^*))$;
(Continue to the right column ↗)

$\kappa \leftarrow \mathsf{HKG}(1^k)$;
$h^* = (h_1^* \| \ldots \| h_k^*) \leftarrow \mathsf{H}_\kappa(vk^*)$;
For $i \in [k]$:
  $pk_i^{(h_i^*)} \leftarrow pk'_i$;
  $(pk_i^{(1-h_i^*)}, sk_i^{(1-h_i^*)}) \leftarrow \mathsf{PKG}(1^k)$;
End For
$PK \leftarrow ((pk_i^{(j)})_{i,j}, crs, \kappa)$;
$C^* \leftarrow (vk^*, (c_i^*)_i, \pi^*, \sigma^*)$;
$\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$;
$K_0^* \leftarrow \{0,1\}^k$;
$b \leftarrow \{0,1\}$;
$b' \leftarrow \mathcal{A}(PK, \widehat{SK}_{C^*}, C^*, K_b^*)$

Game 2: Same as Game 1, except that we generate each tuple $(pk_i^{(h_i^*)}, c_i^*, r_i^*)$ by using the simulation algorithms Fake and Explain of the SNCE scheme Π. More

precisely, in this game, the step with the underline in Game 1 is replaced with: "$(pk'_i, c^*_i, \omega^*_i) \leftarrow \mathsf{Fake}(1^k)$ and $r^*_i \leftarrow \mathsf{Explain}(\omega^*_i, K^*_1)$."

Game 3: Same as Game 2, except that the information of K^*_1 is erased from the witness w^*. More precisely, in this game, the steps "$r^*_i \leftarrow \mathsf{Explain}(\omega^*_i, K^*_1)$" and "$w^* \leftarrow ((r^*_i)_i, K^*_1)$" in Game 2 are replaced with the steps "$r'_i \leftarrow \mathsf{Explain}(\omega^*_i, 0^k)$" and "$w' \leftarrow ((r'_i)_i, 0^k)$," respectively.

For $i \in [3]$, let Succ_i be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs). We will show that $|\Pr[\mathsf{Succ}_i] - \Pr[\mathsf{Succ}_{i+1}]|$ is negligible for each $i \in [2]$ and that $\Pr[\mathsf{Succ}_3] = 1/2$, which proves the eCPA security of the PKEM $\Gamma_{\mathtt{DDN}}$.

Firstly, we can show that $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ is negligible due to the SNC security of the k-repetition construction Π^k, which in turn follows from the SNC security of the underlying SNCE scheme Π (see the explanation in the last paragraph of Section 2.1).

Secondly, we can show that $|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$ is negligible due to the WI security of the non-interactive argument system $\mathcal{P}$. Note that in Game 2 (and Game 3), every pair $(pk_i^{(h^*_i)}, c^*_i)$ is generated by the simulation algorithm Fake, and hence can be explained as an encryption of an arbitrary plaintext (by using Explain). This in particular means that there are many witnesses for the statement $x^* = ((pk_i^{(h^*_i)})_i, (c^*_i)_i) \in L_k$, and we exploit this fact. Specifically, for $i \in [n]$, let ω_i be the state information corresponding to $(pk_i^{(h^*_i)}, c^*_i)$, and let $w_1 = (K^*_1, (r^*_i)_i)$ (resp. $w_0 = (0^k, (r'_i)_i)$) be a witness for the fact that "each c^*_i encrypts K^*_1 (resp. 0^k)," where each r^*_i (resp. r'_i) is computed by $r^*_i = \mathsf{Explain}(\omega_i, K^*_i)$ (resp. $r'_i = \mathsf{Explain}(\omega_i, 0^k)$). We can construct a reduction algorithm that attacks the WI security of $\mathcal{P}$ so that it uses the above witnesses w_1 and w_0 as its challenge, simulates Game 2 and Game 3 for $\mathcal{A}$ depending on its challenge bit, and has advantage exactly $|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$.

Finally, observe that in Game 3, the challenge ciphertext C^* is independent of K^*_1, and the input $(PK, \widehat{SK}_{C^*}, C^*, K^*_b)$ to $\mathcal{A}$ is distributed independently for both $b \in \{0, 1\}$. This implies $\Pr[\mathsf{Succ}_3] = 1/2$. □.

References

1. Applebaum, B.: Key-dependent message security: Generic amplification and completeness. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 527–546. Springer, Heidelberg (2011)
2. Applebaum, B.: Garbling XOR gates "for free" in the standard model. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 162–181. Springer, Heidelberg (2013)
3. Applebaum, B.: Key-dependent message security: Generic amplification and completeness. J. of Cryptology 27(3), 429–451 (2014)
4. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)
5. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 398–415. Springer, Heidelberg (2013)
6. Bellare, M., Hoang, V.T., Rogaway, P.: Foundations of garbled circuits, Full version of [7] (2012), http://eprint.iacr.org/2012/265
7. Bellare, M., Hoang, V.T., Rogaway, P.: Foundations of garbled circuits. In: CCS 2012, pp. 784–796 (2012)

8. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
9. Bellare, M., Yilek, S.: Encryption schemes secure under selective opening attack. This is an updated full version of a preliminary version with Hofheinz [8]. Available at eprint.iacr.org/2009/101
10. Bendlin, R., Nielsen, J.B., Nordholt, P.S., Orlandi, C.: Lower and upper bounds for deniable public-key encryption. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 125–142. Springer, Heidelberg (2011)
11. Black, J., Rogaway, P., Shrimpton, T.: Encryption-scheme security in the presence of key-dependent messages. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 62–75. Springer, Heidelberg (2003)
12. Bleichenbacher, D.: Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 1–12. Springer, Heidelberg (1998)
13. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)
14. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: CCS 2005, pp. 320–329 (2005)
15. Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010)
16. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS 2001, pp. 136–145 (2001)
17. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable encryption. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 90–104. Springer, Heidelberg (1997)
18. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: STOC 1996, pp. 639–648 (1996)
19. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
20. Cash, D.M., Kiltz, E., Shoup, V.: The twin Diffie-Hellman problem and applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)
21. Dachman-Soled, D.: A black-box construction of a CCA2 encryption scheme from a plaintext aware (sPA1) encryption scheme. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 37–55. Springer, Heidelberg (2014)
22. Damgård, I., Jurik, M.: A generalization, a simplification and some applications of Paillier's probabilistic public-key system. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001)
23. Damgård, I.B., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (2000)
24. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: STOC 1991, pp. 542–552 (1991)
25. Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 342–360. Springer, Heidelberg (2004)

26. Fehr, S., Hofheinz, D., Kiltz, E., Wee, H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 381–402. Springer, Heidelberg (2010)
27. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. J. of Cryptology 26(1), 80–101 (2013)
28. Garay, J.A., Wichs, D., Zhou, H.-S.: Somewhat non-committing encryption and efficient adaptively secure oblivious transfer. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 505–523. Springer, Heidelberg (2009)
29. Gertner, Y., Malkin, T., Myers, S.: Towards a separation of semantic and CCA security for public key encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 434–455. Springer, Heidelberg (2007)
30. Goldreich, O.: Foundations of Cryptography - Volume 1. Cambridge University Press (2001)
31. Goldreich, O.: Foundations of Cryptography - Volume 2. Cambridge University Press (2004)
32. Goldreich, O.: Basing non-interactive zero-knowledge on (enhanced) trapdoor permutations: The state of the art. In: Goldreich, O. (ed.) Studies in Complexity and Cryptography. LNCS, vol. 6650, pp. 406–421. Springer, Heidelberg (2011)
33. Goldwasser, S., Micali, S.: Probabilistic encryption. J. of Computer and System Sciences 28(2), 270–299 (1984)
34. Hanaoka, G., Kurosawa, K.: Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 308–325. Springer, Heidelberg (2008)
35. Hazay, C., Patra, A.: One-sided adaptively secure two-party computation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 368–393. Springer, Heidelberg (2014)
36. Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer, Heidelberg (2011)
37. Hofheinz, D., Kiltz, E.: Practical chosen ciphertext secure encryption from factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
38. Hohenberger, S., Lewko, A., Waters, B.: Detecting dangerous queries: A new approach for chosen ciphertext security. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 663–681. Springer, Heidelberg (2012)
39. Huang, Z., Liu, S., Qin, B.: Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 369–385. Springer, Heidelberg (2013)
40. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
41. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive trapdoor functions and chosen-ciphertext security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)
42. Lin, H., Tessaro, S.: Amplification of chosen-ciphertext security. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 503–519. Springer, Heidelberg (2013)
43. MacKenzie, P.D., Reiter, M.K., Yang, K.: Alternatives to non-malleability: Definitions, constructions and applications. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 171–190. Springer, Heidelberg (2004)
44. Malkin, T., Teranishi, I., Yung, M.: Efficient circuit-size independent public key encryption with KDM security. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 507–526. Springer, Heidelberg (2011)

45. Matsuda, T., Hanaoka, G.: Achieving chosen ciphertext security from detectable public key encryption efficiently via hybrid encryption. In: Sakiyama, K., Terada, M. (eds.) IWSEC 2013. LNCS, vol. 8231, pp. 226–243. Springer, Heidelberg (2013)
46. Matsuda, T., Hanaoka, G.: Key encapsulation mechanisms from extractable hash proof systems, revisited. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 332–351. Springer, Heidelberg (2013)
47. Matsuda, T., Hanaoka, G.: Chosen ciphertext security via point obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 95–120. Springer, Heidelberg (2014)
48. Matsuda, T., Hanaoka, G.: Chosen ciphertext security via UCE. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 56–76. Springer, Heidelberg (2014)
49. Mol, P., Yilek, S.: Chosen-ciphertext security from slightly lossy trapdoor functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 296–311. Springer, Heidelberg (2010)
50. Myers, S., Sergi, M., shelat, a.: Blackbox construction of a more than non-malleable CCA1 encryption scheme from plaintext awareness. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 149–165. Springer, Heidelberg (2012)
51. Myers, S., Shelat, A.: Bit encryption is complete. In: FOCS 2009, pp. 607–616 (2009)
52. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: STOC 1989, pp. 33–43 (1989)
53. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990, pp. 427–437 (1990)
54. O'Neill, A., Peikert, C., Waters, B.: Bi-deniable public-key encryption. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 525–542. Springer, Heidelberg (2011)
55. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
56. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008, pp. 187–196 (2008)
57. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
58. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: STOC 1990, pp. 387–394 (1990)
59. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
60. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: STOC 2014, pp. 475–484 (2014)
61. Wee, H.: Efficient chosen-ciphertext security via extractable hash proofs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 314–332. Springer, Heidelberg (2010)

A Basic Cryptographic Primitives

Public Key Encryption. A public key encryption (PKE) scheme Π consists of the three PPTAs $(\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ with the following interface:

Key Generation:	**Encryption:**	**Decryption:**
$(pk, sk) \leftarrow \mathsf{PKG}(1^k)$	$c \leftarrow \mathsf{Enc}(pk, m)$	$m \ (\text{or } \perp) \leftarrow \mathsf{Dec}(sk, c)$

where Dec is a deterministic algorithm, (pk, sk) is a public/secret key pair, and c is a ciphertext of a plaintext m under pk. We require for all $k \in \mathbb{N}$, all (pk, sk) output by $\mathsf{PKG}(1^k)$, and all m, it holds that $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$.

$\underline{\mathsf{Expt}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k)}$:
$(pk, sk) \leftarrow \mathsf{PKG}(1^k)$
$(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(pk)$
$b \leftarrow \{0,1\}$
$c^* \leftarrow \mathsf{Enc}(pk, m_b)$
$b' \leftarrow \mathcal{A}_2(\mathsf{st}, c^*)$
Return $(b' \stackrel{?}{=} b)$.

$\underline{\mathsf{Expt}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k)}$:
$(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
$(c^*, K_1^*) \leftarrow \mathsf{Encap}(pk)$
$K_0^* \leftarrow \{0,1\}^k$
$b \leftarrow \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathsf{Decap}(sk,\cdot)}(\mathsf{st}, c^*, K_b^*)$
Return $(b' \stackrel{?}{=} b)$.

$\underline{\mathsf{Expt}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k)}$:
$(vk, sigk) \leftarrow \mathsf{SKG}(1^k)$
$(m, \mathsf{st}) \leftarrow \mathcal{A}_1(vk)$
$\sigma \leftarrow \mathsf{Sign}(sigk, m)$
$(m', \sigma') \leftarrow \mathcal{A}_2(\mathsf{st}, \sigma)$
Return 1 iff **(a)** $\wedge$ **(b)** :
(a) $\mathsf{SVer}(vk, m', \sigma') = \top$
(b) $(m', \sigma') \neq (m, \sigma)$

Fig. 5. The CPA security experiment for a PKE scheme Π (left), the ATK security experiment (with $\mathtt{ATK} \in \{\mathtt{CCA}, \mathtt{DCCA}, \mathtt{CPA}\}$) for a (detectable) KEM Γ (center), and the SOT security experiment (right)

We say that a PKE scheme Π is CPA secure if for all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the experiment $\mathsf{Expt}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k)$ is defined as in Fig. 5 (left). In the experiment, it is required that $|m_0| = |m_1|$.

(Detectable) Key Encapsulation Mechanism. A key encapsulation mechanism (KEM) Γ consists of the three PPTAs $(\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ with the following interface:

Key Generation:	**Encapsulation:**	**Decapsulation:**
$(pk, sk) \leftarrow \mathsf{KKG}(1^k)$	$(c, K) \leftarrow \mathsf{Encap}(pk)$	K (or $\perp$) $\leftarrow \mathsf{Decap}(sk, c)$

where Decap is a deterministic algorithm, (pk, sk) is a public/secret key pair, and c is a ciphertext of a session-key $K \in \{0,1\}^k$ under pk. We require for all $k \in \mathbb{N}$, all (pk, sk) output by $\mathsf{KKG}(1^k)$, and all $(c, K) \leftarrow \mathsf{Encap}(pk)$, it holds that $\mathsf{Decap}(sk, c) = K$.

A tuple of PPTAs $\Gamma = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{F})$ is said to be a *detectable* KEM if the tuple $(\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ constitutes a KEM, and F is a predicate that takes a public key pk and two ciphertexts c, c' as input and outputs either 0 or 1. (The interface is exactly the same as that of the predicate F of a PKEM introduced in Section 3.) The predicate F is used to define *detectable CCA (*DCCA*) security* (and another notion *unpredictability*) for a detectable KEM.[3]

For $\mathtt{ATK} \in \{\mathtt{CCA}, \mathtt{DCCA}, \mathtt{CPA}\}$, we say that a (detectable) KEM Γ is ATK secure if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the ATK experiment $\mathsf{Expt}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k)$ is defined as in Fig. 5 (center). In the experiment, $\mathcal{A}$ is not allowed to submit "prohibited" queries that are defined based on ATK: If $\mathtt{ATK} = \mathtt{CCA}$, then the prohibited query is c^*.; If $\mathtt{ATK} = \mathtt{DCCA}$, then the prohibited queries are c such that $\mathsf{F}(pk, c^*, c) = 1$.; If $\mathtt{ATK} = \mathtt{CPA}$, then $\mathcal{A}$ is not allowed to submit any query.

Signature. A signature scheme Σ consists of the three PPTAs $(\mathsf{SKG}, \mathsf{Sign}, \mathsf{SVer})$ with the following interface:

Key Generation:	**Signing:**	**Verification:**
$(vk, sigk) \leftarrow \mathsf{SKG}(1^k)$	$\sigma \leftarrow \mathsf{Sign}(sigk, m)$	$\top$ or $\perp \leftarrow \mathsf{SVer}(vk, m, \sigma)$

[3] In this proceedings version we do not recall *unpredictability* of a detectable KEM. For its formal definition, see the full version (or the papers [38,45]).

$\underline{\mathsf{Expt}^{\mathtt{Sound}}_{\mathcal{P},\mathcal{A}}(k):}$
$crs \leftarrow \mathsf{CRSG}(1^k)$
$(x,\pi) \leftarrow \mathcal{A}(crs)$
Return 1 iff **(a)** $\wedge$ **(b)**:
(a) $x \notin L_k$
(b) $\mathsf{PVer}(crs,x,\pi) = \top$

$\underline{\mathsf{Expt}^{\mathtt{WI}}_{\mathcal{P},\mathcal{A}}(k):}$
$(x,w_0,w_1,\mathsf{st}) \leftarrow \mathcal{A}_1(1^k)$
$crs \leftarrow \mathsf{CRSG}(1^k)$
$b \leftarrow \{0,1\}$
$\pi \leftarrow \mathsf{Prove}(crs,x,w_b)$
$b' \leftarrow \mathcal{A}_2(\mathsf{st},crs,\pi)$
Return $(b' \stackrel{?}{=} b)$.

$\underline{\mathsf{Expt}^{\mathtt{ZK\text{-}Real}}_{\mathcal{P},\mathcal{A}}(k):}$
$(x,w,\mathsf{st}) \leftarrow \mathcal{A}_1(1^k)$
$crs \leftarrow \mathsf{CRSG}(1^k)$
$\pi \leftarrow \mathsf{Prove}(crs,x,w)$
$b' \leftarrow \mathcal{A}_2(\mathsf{st},crs,\pi)$
Return b'.

$\underline{\mathsf{Expt}^{\mathtt{ZK\text{-}Sim}}_{\mathcal{P},\mathcal{S},\mathcal{A}}(k):}$
$(x,w,\mathsf{st}) \leftarrow \mathcal{A}_1(1^k)$
$(crs,td) \leftarrow \mathsf{SimCRS}(1^k)$
$\pi \leftarrow \mathsf{SimPrv}(td,x)$
$b' \leftarrow \mathcal{A}_2(\mathsf{st},crs,\pi)$
Return b'.

Fig. 6. Security experiments for a non-interactive argument system

where SVer is a deterministic algorithm, $(vk, sigk)$ is a verification/signing key pair, and σ is a signature on a message m under the key pair $(vk, sigk)$. The symbol $\top$ (resp. $\bot$) indicates "accept" (resp. "reject"). We require for all $k \in \mathbb{N}$, all $(vk, sigk)$ output by $\mathsf{SKG}(1^k)$, and all m, it holds that $\mathsf{SVer}(vk, m, \mathsf{Sign}(vk, m)) = \top$.

We say that a signature scheme Σ is strongly one-time secure (SOT secure, for short) if for all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k) = 1]$ is negligible, where the experiment $\mathsf{Expt}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k)$ is defined as in Fig. 5 (right).

A SOT secure signature scheme can be built from any one-way function [52,58].

Universal One-Way Hash Function. We say that a pair of PPTAs $\mathcal{H} = (\mathsf{HKG}, \mathsf{H})$ is a universal one-way hash function (UOWHF) if the following two properties are satisfied: (1) On input 1^k, HKG outputs a hash-key κ. For any hash-key κ output from $\mathsf{HKG}(1^k)$, H defines an (efficiently computable) function of the form $\mathsf{H}_\kappa : \{0,1\}^* \to \{0,1\}^k$. (2) For all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{A}}(k) = 1]$ is negligible, where the experiment is defined as follows:

$$\mathsf{Expt}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{A}}(k) : [\ (m,\mathsf{st}) \leftarrow \mathcal{A}_1(1^k);\ \kappa \leftarrow \mathsf{HKG}(1^k);\ m' \leftarrow \mathcal{A}_2(\mathsf{st},\kappa);\ \text{Return 1 iff } \mathsf{H}_\kappa(m') = \mathsf{H}_\kappa(m) \wedge m' \neq m.\ ].$$

A UOWHF can be built from any one-way function [52,58].

Non-interactive Argument Systems. Let $L = \{L_k\}_{k\in\mathbb{N}}$ be an NP language (for simplicity, we assume that L consists of sets L_k parameterized by the security parameter k). A non-interactive argument system $\mathcal{P}$ for L consists of the three algorithms $(\mathsf{CRSG}, \mathsf{Prove}, \mathsf{PVer})$ with the following interface:

CRS Generation:	**Proving:**	**Verification:**
$crs \leftarrow \mathsf{CRSG}(1^k)$	$\pi \leftarrow \mathsf{Prove}(crs,x,w)$	$\top$ or $\bot \leftarrow \mathsf{PVer}(crs,x,\pi)$

where PVer is a deterministic algorithm, crs is a common reference string (CRS), x is a statement, w is a witness for the fact that $x \in L_k$, and π is a proof string (that is supposed to prove that $x \in L_k$). The symbol $\top$ (resp. $\bot$) indicates "accept" (resp. "reject"). We require for all $k \in \mathbb{N}$, all crs output by $\mathsf{CRSG}(1^k)$, and all statement/witness pairs $(x, w) \in L_k \times \{0,1\}^*$ (where w is a witness for the fact that $x \in L_k$), it holds that $\mathsf{PVer}(crs, x, \mathsf{PVer}(crs, x, w)) = \top$.

We say that a non-interactive argument system $\mathcal{P}$ for a language L satisfies *adaptive soundness* if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{Sound}}_{\mathcal{P},\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{Sound}}_{\mathcal{P},\mathcal{A}}(k) = 1]$ is negligible, where the Sound experiment $\mathsf{Expt}^{\mathtt{Sound}}_{\mathcal{P},\mathcal{A}}(k)$ is defined as in Fig. 6 (leftmost).

We say that a non-interactive argument system $\mathcal{P}$ for an NP language L satisfies *witness indistinguishability* (WI security, for short) if for all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{WI}}_{\mathcal{P},\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{WI}}_{\mathcal{P},\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the WI experiment $\mathsf{Expt}^{\mathtt{WI}}_{\mathcal{P},\mathcal{A}}(k)$ is defined as in Fig. 6 (second-left), and it is required that $x \in L_k$, and both w_0 and w_1 are witnesses for the fact that $x \in L_k$ in the WI experiment. [4]

Finally, we recall the definition of the *zero-knowledge property* (ZK security, for short). We say that a non-interactive argument system $\mathcal{P}$ for an NP language L satisfies the *zero-knowledge* property (ZK secure, for short) if there exists a pair of PPTAs $\mathcal{S} = (\mathsf{SimCRS}, \mathsf{SimPrv})$ satisfying the following properties:

- (**Syntax:**) SimCRS is the "simulated common reference string" generation algorithm that takes 1^k as input, and outputs crs and a corresponding trapdoor td.; SimPrv is the "simulated proof" generation algorithm that takes td (output by SimCRS) and a statement $x \in \{0,1\}^*$ (which may not belong to L_k) as input, and outputs a "simulated proof" π.
- (**Zero-Knowledge:**) For all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{ZK}}_{\mathcal{P},\mathcal{S},\mathcal{A}}(k) := |\Pr[\mathsf{Expt}^{\mathtt{ZK\text{-}Real}}_{\mathcal{P},\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{ZK\text{-}Sim}}_{\mathcal{P},\mathcal{S},\mathcal{A}}(k) = 1]|$ is negligible, where the ZK-Real experiment $\mathsf{Expt}^{\mathtt{ZK\text{-}Real}}_{\mathcal{P},\mathcal{A}}(k)$ and the ZK-Sim experiment $\mathsf{Expt}^{\mathtt{ZK\text{-}Sim}}_{\mathcal{P},\mathcal{S},\mathcal{A}}(k)$ are defined as in Fig. 6 (second-right and rightmost, respectively), and furthermore it is required that $x \in L_k$ and w is a witness for the fact that $x \in L_k$ in both of the experiments.

B Postponed Proofs

B.1 Proof of Lemma 4: Strong Decapsulation Soundness of $\widehat{\Gamma}$

Let $\mathcal{A}$ be a PPTA sDSND adversary. Let (PK, SK, C^*, K^*) be a tuple that is input to $\mathcal{A}$ in the sDSND experiment, where $PK = ((pk_i^{(j)})_{i,j}, pk_{k+1}, \kappa)$, $SK = ((sk_i^{(j)})_{i,j}, PK)$, and $C^* = (h^*, (c_i^*)_i, \widetilde{c}^*)$.

Let us call $\mathcal{A}$'s output $C' = (h', (c'_i)_i, \widetilde{c}')$ in the sDSND experiment *successful* if C' satisfies the conditions that make the experiment output 1, i.e. $\widehat{\mathsf{F}}(PK, C^*, C') = 1$ (which is equivalent to $h' = h^*$), $C' \neq C^*$, and $\widehat{\mathsf{Decap}}(SK, C') \neq \perp$. Below, we use asterisk (*) to denote the values generated/chosen during the generation of C^*, and prime ($'$) to denote the values generated during the calculation of $\widehat{\mathsf{Decap}}(SK, C')$.

We first confirm that a successful ciphertext C' must additionally satisfy $(c'_{k+1}, \widetilde{c}') \neq (c^*_{k+1}, \widetilde{c}^*)$. To see this, assume the opposite, i.e. $(c'_{k+1}, \widetilde{c}') = (c^*_{k+1}, \widetilde{c}^*)$. Here, $c'_{k+1} = c^*_{k+1}$ implies $\alpha' = \alpha^*$ (due to the correctness of the SNCE scheme Π). This and $\widetilde{c}' = \widetilde{c}^*$ imply $(r'_i)_{i\in[k+1]} = (r^*_i)_{i\in[k+1]}$ (due to the correctness of the SKE scheme E), which in turn implies $(c'_i)_i = (c^*_i)_i$. Hence, it holds that $C' = (h', (c'_i)_i, \widetilde{c}') = (h^*, (c^*_i)_i, \widetilde{c}^*) = C^*$, but this contradicts $C' \neq C^*$.

[4] We note that unlike soundness, we do *not* need a version of the WI security in which a statement (and witnesses) may depend on a common reference string.

So far, we have seen that a successful ciphertext C' must satisfy $\mathsf{H}_\kappa(c'_{k+1}\|\widetilde{c}') = h' = h^* = \mathsf{H}_\kappa(c^*_{k+1}\|\widetilde{c}^*)$ and $(c'_{k+1},\widetilde{c}') \neq (c^*_{k+1},\widetilde{c}^*)$, which means that $(c'_{k+1}\|\widetilde{c}')$ and $(c^*_{k+1}\|\widetilde{c}^*)$ constitute a collision pair for H_κ. Using this fact, we can construct a PPTA $\mathcal{B}_\mathtt{h}$ whose advantage in the $\mathtt{UOW}$ experiment regarding $\mathcal{H}$ is exactly the probability that $\mathcal{A}$ outputs a successful ciphertext in the $\mathtt{sDSND}$ experiment, which combined with the security of the UOWHF $\mathcal{H}$, proves the lemma. Since the reduction algorithm is straightforward given the explanation here, we omit its description. (In the full version, we provide the details of the reduction algorithm.) □

B.2 Proof of Lemma 5: Strong Punctured Decapsulation Soundness of $\widehat{\Gamma}$

Let (PK, SK) be a key pair output by $\widehat{\mathsf{KKG}}(1^k)$, where $PK = ((pk_i^{(j)})_{i,j}, pk_{k+1}, \kappa)$ and $SK = ((sk_i^{(j)})_{i,j}, PK)$. Let $C^* = (h^*, (c_i^*)_i, \widetilde{c}^*)$ be any ciphertext output by $\widehat{\mathsf{Encap}}(PK)$, and let $\widehat{SK}_{C^*} = (h^*, (sk_i^{(1-h_i^*)})_i, PK)$ be the punctured secret key generated by $\widehat{\mathsf{Punc}}(SK, C^*)$. We show that for any ciphertext $C = (h, (c_i)_i, \widetilde{c})$ (which might be outside the range of $\widehat{\mathsf{Encap}}(PK)$) satisfying $\widehat{\mathsf{F}}(PK, C^*, C) = 0$ (i.e. $h \neq h^*$), it holds that $\widehat{\mathsf{Decap}}(SK, C) = \widehat{\mathsf{PDecap}}(\widehat{SK}_{C^*}, C)$. Note that this implies that there exists no ciphertext that violates (strong) punctured decapsulation soundness of the PKEM $\widehat{\Gamma}$, and thus for any (even computationally unbounded) $\mathtt{sPDSND}$ adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{sPDSND}}_{\widehat{\Gamma},\mathcal{A}}(k) = 0$, which will prove the lemma.

To show the above, fix arbitrarily a ciphertext $C = (h, (c_i)_i, \widetilde{c})$ satisfying $\widehat{\mathsf{F}}(PK, C^*, C) = 0$ (and hence $h^* \neq h$) and let $\ell = \min\{i \in [k] \mid h_i^* \neq h_i\}$, where each of h_i and h_i^* are the i-th bit of h and h^*, respectively. For notational convenience, let $\alpha_1 = \mathsf{Dec}(sk_1^{(h_1)}, c_1)$ and $\alpha_\ell = \mathsf{Dec}(sk_\ell^{(1-h_\ell^*)}, c_\ell) = \mathsf{Dec}(sk_\ell^{(h_\ell)}, c_\ell)$, where the latter equality is because $h_\ell^* \neq h_\ell$ implies $1 - h_\ell^* = h_\ell$. We consider the following two cases, and show that the results from both of the algorithms $\widehat{\mathsf{Decap}}$ and $\widehat{\mathsf{PDecap}}$ always agree.

Case $\alpha_1 = \alpha_\ell$: Both $\widehat{\mathsf{Decap}}$ and $\widehat{\mathsf{PDecap}}$ proceed identically after they respectively compute α_1 and α_ℓ, and thus the outputs from these algorithms agree.

Case $\alpha_1 \neq \alpha_\ell$: In this case, both $\widehat{\mathsf{Decap}}$ and $\widehat{\mathsf{PDecap}}$ return $\perp$. Specifically, $\alpha_1 \neq \alpha_\ell$ and the correctness of the SNCE scheme Π imply that there does not exist r_ℓ such that $\mathsf{Enc}(pk_\ell^{(h_\ell)}, \alpha_1; r_\ell) = c_\ell$, and thus $\widehat{\mathsf{Decap}}$ returns $\perp$ in its last step at the latest (it may return $\perp$ earlier if $\alpha_1 = \perp$ or $\mathsf{SDec}(\alpha_1, \widetilde{c}) = \perp$). Symmetrically, there does not exist r_1 such that $\mathsf{Enc}(pk_1^{(h_1)}, \alpha_\ell; r_1) = c_1$, and thus $\widehat{\mathsf{PDecap}}$ returns $\perp$ in its last step at the latest (it may return $\perp$ earlier as above).

This completes the proof of Lemma 5. □

B.3 Proof of Lemma 6: $\mathtt{eCPA}$ Security of $\widehat{\Gamma}$

Let $\mathcal{A}$ be any PPTA adversary that attacks the $\mathtt{eCPA}$ security of $\widehat{\Gamma}$. For this $\mathcal{A}$, we consider the sequence of games described in the explanation in Section 4. Here, we only show that $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ and $|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$ are negligible, which should be sufficient for the proof of Lemma 6, given the intuitive explanation in Section 4.

Claim 1. *There exists a PPTA* $\mathcal{B}_{\mathrm{p}}$ *such that* $\mathsf{Adv}^{\mathtt{SNC}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$.

Proof of Claim 1. We show how to construct a PPTA adversary $\mathcal{B}_{\mathrm{p}}$ that attacks the $\mathtt{SNC}$ security of the $(k+1)$-repetition construction Π^{k+1} of the SNCE scheme with the claimed advantage. The description of $\mathcal{B}_{\mathrm{p}} = (\mathcal{B}_{\mathrm{p1}}, \mathcal{B}_{\mathrm{p2}})$ as follows:

$\mathcal{B}_{\mathrm{p1}}(1^k)$**:** $\mathcal{B}_{\mathrm{p1}}$ picks $\alpha^* \in \mathcal{K}_k$ uniformly at random, and sets $\mathsf{st}_{\mathcal{B}} \leftarrow (\mathcal{B}_{\mathrm{p1}}$'s entire view). Then $\mathcal{B}_{\mathrm{p1}}$ terminates with output $(\alpha^*, \mathsf{st}_{\mathcal{B}})$ (where α^* is regarded as $\mathcal{B}_{\mathrm{p}}$'s challenge message).

$\mathcal{B}_{\mathrm{p2}}(\mathsf{st}_{\mathcal{B}}, PK' = (pk'_i)_{i\in[k+1]}, C'^* = (c^*_i)_{i\in[k+1]}, R'^* = (r^*_i)_{i\in[k+1]})$**:** $\mathcal{B}_{\mathrm{p2}}$ picks $K^*_1 \leftarrow \{0,1\}^k$ uniformly at random, sets $\beta^* \leftarrow ((r^*_i)_{i\in[k+1]}, K^*_1)$, and runs $\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, \beta^*)$, $\kappa \leftarrow \mathsf{HKG}(1^k)$, and $h^* = (h^*_1 \| \ldots \| h^*_k) \leftarrow \mathsf{H}_\kappa(c^*_{k+1} \| \widetilde{c}^*)$. For each $i \in [k]$, $\mathcal{B}_{\mathrm{p2}}$ sets $pk_i^{(h^*_i)} \leftarrow pk'_i$ and runs $(pk_i^{(1-h^*_i)}, sk_i^{(1-h^*_i)}) \leftarrow \mathsf{PKG}(1^k)$. Next $\mathcal{B}_{\mathrm{p2}}$ sets $PK \leftarrow ((pk_i^{(j)})_{i,j}, pk'_{k+1}, \kappa)$, $C^* \leftarrow (h^*, (c^*_i)_i, \widetilde{c}^*)$, and $\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h^*_i)})_i, PK)$. Then $\mathcal{B}_{\mathrm{p2}}$ picks $K^*_0 \in \{0,1\}^k$ and $b \in \{0,1\}$ uniformly at random, runs $b' \leftarrow \mathcal{A}(PK, \widehat{SK}_{C^*}, C^*, K^*_b)$, and terminates with output $(b' \stackrel{?}{=} b)$.

The above completes the description of $\mathcal{B}_{\mathrm{p}}$. Note that $\mathcal{B}_{\mathrm{p2}}$ outputs 1 only when $b' = b$ occurs. $\mathcal{B}_{\mathrm{p}}$'s $\mathtt{SNC}$ advantage can be estimated as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{SNC}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k) &= |\Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k) = 1]| \\ &= |\Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k) : b' = b] - \Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k) : b' = b]|.\end{aligned}$$

Consider the case when $\mathcal{B}_{\mathrm{p}}$ runs in $\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k)$. It is easy to see that in this case, $\mathcal{B}_{\mathrm{p}}$ perfectly simulates Game 1 for $\mathcal{A}$. In particular, every $pk_i^{(j)}$ and pk_{k+1} in PK are generated honestly by running $\mathsf{PKG}(1^k)$, and every c^*_i in C^* is generated as $c^*_i \leftarrow \mathsf{Enc}(pk_i^{(h^*_i)}, \alpha^*; r^*_i)$ where $\alpha^* \in \mathcal{K}_k$ and each of $r^*_i \in \mathcal{R}_k$ are chosen uniformly at random, as done in Game 1. Under this situation, the probability that $b' = b$ occurs is exactly the same as the probability that $\mathcal{A}$ succeeds in guessing its challenge bit in Game 1, i.e., $\Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k) : b' = b] = \Pr[\mathsf{Succ}_1]$.

When $\mathcal{B}_{\mathrm{p}}$ runs in $\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k)$, on the other hand, each of pairs $(pk_i^{(h^*_i)}, c^*_i)$ and each r^*_i are generated by using the simulation algorithms Fake and $\mathsf{Explain}$ of the underlying SNCE scheme Π, in such a way that the plaintext corresponding to c^*_i is "explained" as $\alpha^* \in \mathcal{K}_k$ that is chosen uniformly at random, as done in Game 2. The rest of the procedures remains unchanged from the above case. Therefore, the probability that $b' = b$ occurs is exactly the same as the probability that $\mathcal{A}$ succeeds in guessing its challenge bit in Game 2, i.e., $\Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k) : b' = b] = \Pr[\mathsf{Succ}_2]$.

In summary, we have $\mathsf{Adv}^{\mathtt{SNC}}_{\Pi^{k+1},\mathcal{B}_{\mathrm{p}}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$. This completes the proof of Claim 1. □

Claim 2. *There exists a PPTA* $\mathcal{B}_{\mathrm{e}}$ *such that* $\mathsf{Adv}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{B}_{\mathrm{e}}}(k) = |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$.

Proof of Claim 2. We show how to construct a PPTA adversary $\mathcal{B}_{\mathrm{e}}$ that attacks the $\mathcal{F}$-$\mathtt{OTKDM}$ security of the underlying SKE scheme E with the claimed advantage. The description of $\mathcal{B}_{\mathrm{e}} = (\mathcal{B}_{\mathrm{e1}}, \mathcal{B}_{\mathrm{e2}})$ is as follows:

$\mathcal{B}_{\mathsf{e}1}(1^k)$**:** For every $i \in [k+1]$, $\mathcal{B}_{\mathsf{e}}$ runs $(pk'_i, c^*_i, \omega^*_i) \leftarrow \mathsf{Fake}(1^k)$. Then, $\mathcal{B}_{\mathsf{e}1}$ picks $K^*_1 \in \{0,1\}^k$ uniformly at random. Next, $\mathcal{B}_{\mathsf{e}1}$ specifies the function $f : \mathcal{K}_k \to \mathcal{M}_k$ which is used as an encryption query in the $\mathtt{OTKDM}$ experiment, defined by: $\alpha \overset{f}{\mapsto} (\mathsf{Explain}(\omega^*_i, \alpha)_{i\in[k+1]}, K^*_1)$, where each ω^*_i and K^*_1 are treated as fixed parameters hard-coded in f. (Note that $f \in \mathcal{F}_k$.) Finally, $\mathcal{B}_{\mathsf{e}1}$ sets $\mathsf{st}_{\mathcal{B}} \leftarrow (\mathcal{B}_{\mathsf{e}1}$'s entire view), and terminates with output $(f, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathsf{e}2}(\mathsf{st}_{\mathcal{B}}, \widetilde{c}^*)$**:** $\mathcal{B}_{\mathsf{e}2}$ runs $\kappa \leftarrow \mathsf{HKG}(1^k)$ and $h^* = (h^*_1 \| \ldots \| h^*_k) \leftarrow \mathsf{H}_\kappa(c^*_{k+1} \| \widetilde{c}^*)$. Next, for every $i \in [k]$, $\mathcal{B}_{\mathsf{e}2}$ sets $pk_i^{(h^*_i)} \leftarrow pk'_i$ and runs $(pk_i^{(1-h^*_i)}, sk_i^{(1-h^*_i)}) \leftarrow \mathsf{PKG}(1^k)$. Then, $\mathcal{B}_{\mathsf{e}2}$ sets $PK \leftarrow ((pk_i^{(j)})_{i,j}, pk'_{k+1}, \kappa)$, $C^* \leftarrow (h^*, (c^*_i)_i, \widetilde{c}^*)$, and $\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h^*_i)})_i, PK)$. $\mathcal{B}_{\mathsf{e}2}$ picks $K^*_0 \in \{0,1\}^k$ and $b \in \{0,1\}$ uniformly at random, runs $b' \leftarrow \mathcal{A}(PK, \widehat{SK}_{C^*}, C^*, K^*_b)$, and terminates with output $\gamma' \leftarrow (b' \overset{?}{=} b)$.

The above completes the description of $\mathcal{B}_{\mathsf{e}}$. Let $\gamma \in \{0,1\}$ be $\mathcal{B}_{\mathsf{e}}$'s challenge bit. $\mathcal{B}_{\mathsf{e}}$'s $\mathcal{F}$-$\mathtt{OTKDM}$ advantage is estimate as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{B}_{\mathsf{e}}}(k) &= 2 \cdot |\Pr[\gamma' = \gamma] - \frac{1}{2}| = |\Pr[\gamma' = 1|\gamma = 1] - \Pr[\gamma' = 1|\gamma = 0]| \\ &= |\Pr[b' = b|\gamma = 1] - \Pr[b' = b|\gamma = 0]|.\end{aligned}$$

Let $\alpha^* \in \mathcal{K}_k$ be the key, and $M_1 = f(\alpha^*)$ and $M_0 \in \mathcal{M}_k$ be the plaintexts calculated/chosen in $\mathcal{B}_{\mathsf{e}}$'s $\mathtt{OTKDM}$ experiment. Consider the case when $\gamma = 1$, i.e. $\widetilde{c}^*$ is an encryption of $M_1 = f(\alpha^*) = ((r^*_i)_{i\in[k+1]}, K^*_1)$. Note that by the definition of the experiment $\mathsf{Expt}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{B}_{\mathsf{e}}}(k)$, if we regard the key $\alpha^* \in \mathcal{K}_k$ and $M^*_1 = f(\alpha^*)$ in $\mathsf{Expt}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{B}_{\mathsf{e}}}(k)$ as α^* and β^* in Game 2, then each r^*_i is generated by $r^*_i \leftarrow \mathsf{Explain}(\omega^*_i, \alpha^*)$, so that the plaintext corresponding to each c^*_i is α^*, which is how these values are generated in Game 2. Moreover, the public key PK, the values $(c^*_i)_{i\in[k+1]}$ used in the challenge ciphertext C^*, and the punctured secret key $\widehat{SK}_{C^*}$ are distributed identically to those in Game 2. Hence, $\mathcal{B}_{\mathsf{e}}$ simulates Game 2 perfectly for $\mathcal{A}$. Under this situation, the probability that $b' = b$ occurs is exactly the same as the probability that $\mathcal{A}$ succeeds in guessing the challenge bit in Game 2, i.e. $\Pr[b' = b|\gamma = 1] = \Pr[\mathsf{Succ}_2]$.

Next, consider the case when $\gamma = 0$. In this case, $\widetilde{c}^*$ is an encryption of a random message $M_0 \in \mathcal{M}_k$ that is independent of any other values. Then, if we regard the key α^* and the random message M_0 in $\mathsf{Expt}^{\mathtt{OTKDM}}_{E,\mathcal{B}_{\mathsf{e}}}(k)$ as α^* and β' in Game 3, respectively, then $\mathcal{A}$'s challenge ciphertext C^* is generated in such a way that they are distributed identically to those in Game 3, and thus $\mathcal{B}_{\mathsf{e}}$ simulates Game 3 perfectly for $\mathcal{A}$. Therefore, with a similar argument to the above, we have $\Pr[b' = b|\gamma = 0] = \Pr[\mathsf{Succ}_3]$.

In summary, we have $\mathsf{Adv}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{B}_{\mathsf{e}}}(k) = |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$. This completes the proof of Claim 2. □

Due to our assumptions on the building blocks, and the $\mathtt{SNC}$ security of the $(k+1)$-repetition construction Π^{k+1} (see the explanation in Section 2.1), we can conclude that $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ and $|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$ are negligible. Combined with the intuitive explanations given in Section 4, this completes the proof of Lemma 6. □