Constructing and Understanding Chosen Ciphertext Security via Puncturable Key Encapsulation Mechanisms[⋆]

Takahiro Matsuda and Goichiro Hanaoka

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) its 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.

Table of Contents

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 mechanism* (KEM) that are secure against chosen ciphertext attacks (CCA) [58, 63, 29] 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 [6, 12, 61] and universal composability [20, 24], but also its resilience and robustness against practical attacks such as Bleichenbacher's attack [16, 5].

There have been a number of works that show CCA secure PKE/KEMs from general cryptographic primitives: These include trapdoor permutations [29, 35, 36] (with some enhanced property [37]), identity-based encryption [23] and a weaker primitive called tag-based encryption [48, 45], lossy trapdoor function [62] and trapdoor functions with weaker functionality/security properties [65, 54, 46, 67], PKE with weaker than but close to CCA security [43, 47, 26], a combination of chosen plaintext secure (CPA secure) PKE and a hash function with some strong security [53], and techniques from program obfuscation [66, 52].

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 [34]). 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 similarly 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 [29, 62, 65, 67, 52, 53, 26] often employ the ideas and techniques of the Dolev-Dwork-Naor (DDN) construction [29], 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 [29], 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* [53] (which is in turn inspired by the notion of *puncturable pseudorandom function* [66]), and we explain the difference from [53] 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 [29] and several existing DDN-like constructions (e.g. [62, 65, 67, 52, 53]) 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. [43] 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) [15]. Roughly speaking, a SNCE scheme is a special case of non-committing encryption [22] 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] (and its efficient variant [9, §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 similarity with the "double-layered" construction of Myers and Shelat [56] and its variants [43, 50, 26], 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 [52, 53]. 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 [32, 62, 65, 56, 46, 43, 52, 53, 26]. 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 [32], a trapdoor function [62, 65, 46], a PKE scheme which achieves some security which is (weaker than but) close to CCA security [56, 43, 26], or a power of additional building blocks with (seemingly very strong) security properties [52, 53]. We show how SNCE and KDM secure SKE can be used together to overcome the circularity. Compared with the structurally similar constructions [43, 52, 53, 26], 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 [26] who constructs a CCA secure PKE scheme from a PKE scheme that satisfies (standard model) plaintext awareness and some simulatability property. The construction in [26] is similar to our proposed (P)KEM in Section 4 and the recent similar constructions [52, 53]. (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 [56].)

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 [58] and Rackoff and Simon [63]. 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 [29, 62, 65, 67, 52, 53, 26] are based on the ideas and techniques from the DDN construction [29].

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 [53]. Similarly to PKEM, PTBE is a special kind of tag-based encryption [48, 45] 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 [53] 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 [53] 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 [53] also formalized the security requirement called $\mathtt{eCPA}$ *security* whose formalization is a PTBE-analogue of $\mathtt{eCPA}$ 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 basic definitions for standard cryptographic primitives that are not reviewed in this section are given in Appendix A, which include PKE, (detectable) 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 $m, n \in \mathbb{N}$, 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. "(P)PTA" 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. Furthermore, for an algorithm or a function $\mathcal{O}$, "$\mathcal{A}^{\mathcal{O}}$" denotes an algorithm $\mathcal{A}$ with oracle access to $\mathcal{O}$. A function $\epsilon(k) : \mathbb{N} \rightarrow [0, 1]$ is said to be *negligible* if for all positive polynomials $p(k)$ 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 [31, 44], but departs from it because we need perfect correctness (or at least almost-all-keys-perfect correctness [30]) so that error-less decryption is guaranteed, which cannot be achieved by sender-equivocable encryption. We also note that recently, Hazay and Patra [40] 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 [40] later in this subsection.

Formally, a sender non-committing (public key) encryption (SNCE) scheme $\mathit{\Pi}$ 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.1), and Fake and $\mathsf{Explain}$ 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).

$\mathsf{Explain}$**:** 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.

$\mathtt{SNC}$ *Security.* For a SNCE scheme $\mathit{\Pi} = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Explain})$ (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 $\mathtt{SNC\text{-}Real}$ experiment $\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\mathit{\Pi},\mathcal{A}}(k)$ and the $\mathtt{SNC\text{-}Sim}$ experiment $\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\mathit{\Pi},\mathcal{A}}(k)$ as in Fig. 1 (left and center, respectively).

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

The Difference among Non-committing Encryption and Related Primitives. The original definition of non-committing encryption by Canetti et al. [22] 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 [22] (and

```
$\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\Pi,\mathcal{A}}(k)$ :
    $(m, \mathsf{st}) \leftarrow \mathcal{A}_1(1^k)$
    $(pk, sk) \leftarrow \mathsf{PKG}(1^k)$
    $r \leftarrow \mathcal{R}_k$
    $c \leftarrow \mathsf{Enc}(pk, m; r)$
    $b' \leftarrow \mathcal{A}_2(\mathsf{st}, pk, c, r)$
    Return $b'$.
```

```
$\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\Pi,\mathcal{A}}(k)$ :
    $(m, \mathsf{st}) \leftarrow \mathcal{A}_1(1^k)$
    $(pk, c, \omega) \leftarrow \mathsf{Fake}(1^k)$
    $r \leftarrow \mathsf{Explain}(\omega, m)$
    $b' \leftarrow \mathcal{A}_2(\mathsf{st}, pk, c, r)$
    Return $b'$.
```

```
$\mathsf{Expt}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{A}}(k)$ :
    $(f, \mathsf{st}) \leftarrow \mathcal{A}_1(1^k)$
    $K \leftarrow \mathcal{K}_k$
    $m_1 \leftarrow f(K);\ m_0 \leftarrow \mathcal{M}_k$
    $b \leftarrow \{0,1\}$
    $c^* \leftarrow \mathsf{SEnc}(K, m_b)$
    $b' \leftarrow \mathcal{A}_2(\mathsf{st}, c^*)$
    Return $(b' \overset{?}{=} b)$.
```

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).

several works [28, 33]) 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 [31, 44] 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 [21, 59, 14, 66] 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 [59, 14].

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 [30]. This is because our proposed constructions follow the Dolev-Dwork-Naor-style construction [29] 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, 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. [11] formalized the notion of *lossy encryption* [11], 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. [11] 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 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 [11, § 4.4] (which is essentially the multi-bit version of the Goldwasser-Micali scheme [38]), the scheme based on the decisional Diffie-Hellman (DDH) assumption [13, § 5.4] (which is the "bit-wise" encryption version of the DDH-based lossy encryption scheme [11, § 4.1]), and the scheme based on the decisional composite residuosity (DCR) assumption [41] (which shows that the original Paillier scheme [60] and the Damgård-Jurik scheme [27] can be extended to lossy encryption with efficient openability). In particular, the DCR-based schemes [60, 27, 41] 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 [40]. The definition of NCE for the Sender in [40] 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 [11]. 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 [40] (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.

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 $(\mathsf{SEnc}, \mathsf{SDec})$:

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 $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{A}}(k) = 1] - 1/2|$ is negligible.*

[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 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 $\mathcal{M}_k$.

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. [9, §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 [9, §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 [17], the QR assumption [19], the DCR assumption [19, 49], the learning with errors (LWE) assumption [4], and the learning parity with noise (LPN) assumption [4, 2]. Very recently, Bellare et al. [7, 8] 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. Using a version of UCE assumption, they [8] 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. $\mathtt{OTKDM}$ security is the special case of non-adaptive KDM security, and hence we can also use the result of [8] 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 $\mathtt{CCA}$ secure KEM and a $\mathtt{DCCA}$ secure detectable KEM, respectively. We also explain how a wide class of the existing constructions of $\mathtt{CCA}$ 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.

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

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

```
$\mathsf{Expt}^{\mathtt{eCPA}}_{\Gamma,\mathcal{A}}(k)$ :
  $(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
  $(c^*, K_1^*) \leftarrow \mathsf{Encap}(pk)$
  $\widehat{sk}_{c^*} \leftarrow \mathsf{Punc}(sk, c^*)$
  $K_0^* \leftarrow \{0,1\}^k$
  $b \leftarrow \{0,1\}$
  $b' \leftarrow \mathcal{A}(pk, \widehat{sk}_{c^*}, c^*, K_b^*)$
  Return $(b' \overset{?}{=} b)$.
```

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

```
$\mathsf{Expt}^{\mathtt{sPDSND}}_{\Gamma,\mathcal{A}}(k)$ :
  $(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
  $(c^*, K^*) \leftarrow \mathsf{Encap}(pk)$
  $c' \leftarrow \mathcal{A}(pk, sk, c^*, K^*)$
  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 $\mathtt{XXX} \in \{\mathtt{DSND}, \mathtt{sDSND}, \mathtt{PDSND}, \mathtt{sPDSND}\}$:

$\mathsf{Adv}^{\mathtt{XXX}}_{\Gamma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{XXX}}_{\Gamma,\mathcal{A}}(k) = 1]$

eCPA security:

$\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{eCPA}}_{\Gamma,\mathcal{A}}(k) = 1] - \frac{1}{2}|$

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

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 $\widehat{sk}_{c^*}$ (output by $\mathsf{Punc}(sk, c^*)$) and a ciphertext c as input, and outputs a session-key K which could be the special symbol $\bot$ (meaning that "c cannot be decapsulated by $\widehat{sk}_{c^*}$").

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 DCCA security and unpredictability of detectable PKE in [43]. 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 CCA security of a PKEM, while the strong notions are usually easier to work with.

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 (DSND) experiment $\mathsf{Expt}^{\mathtt{DSND}}_{\Gamma,\mathcal{A}}(k)$ and the strong decapsulation soundness (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 $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathtt{DSND}}_{\Gamma,\mathcal{A}}(k)$ (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 strong 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 $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathtt{PDSND}}_{\Gamma,\mathcal{A}}(k)$ (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 $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma,\mathcal{A}}(k)$ 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 Appendix D.1.)

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*

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

Furthermore, if Γ additionally satisfies strong *punctured decapsulation soundness, we have tight security reduction. Specifically, for any PPTA $\mathcal{A}$ that attacks the* $\mathtt{CCA}$ *security of Γ^*, there exist PPTAs $\mathcal{B}_{\mathsf{d}}$, $\mathcal{B}_{\mathsf{a}}$, and $\mathcal{B}_{\mathsf{e}}$ such that*

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

Proof Sketch of Theorem 1. The proofs for the both reductions proceed almost identically. 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. (The tightness of the reduction differs depending on whether we can assume "strong" puncutred decapsulation soundness. For the details, see the explanation in Appendix D.1.)

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. □

3.4 Understanding the Existing Constructions of $\mathtt{CCA}$ 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 $\mathtt{CCA}$ 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 [29], 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.

```
$\mathsf{KKG}_{\mathtt{DDN}}(1^k)$ :
  $\forall(i,j) \in [k] \times \{0,1\}$ :
        $(pk_i^{(j)}, sk_i^{(j)}) \leftarrow \mathsf{PKG}(1^k)$
  $crs \leftarrow \mathsf{CRSG}(1^k)$
  $\kappa \leftarrow \mathsf{HKG}(1^k)$
  $PK \leftarrow ((pk_i^{(j)})_{i,j}, crs, \kappa)$
  $SK \leftarrow ((sk_i^{(j)})_{i,j}, PK)$
  Return $(PK, SK)$.
```

```
$\mathsf{Encap}_{\mathtt{DDN}}(PK)$ :
  $((pk_i^{(j)})_{i,j}, crs, \kappa) \leftarrow PK$
  $K \leftarrow \{0,1\}^k$
  $r_1, \dots, r_k \leftarrow \mathcal{R}_k$
  $(vk, sigk) \leftarrow \mathsf{SKG}(1^k)$
  $h \leftarrow \mathsf{H}_\kappa(vk)$
  View $h$ as $(h_1 \| \dots \| h_k) \in \{0,1\}^k$.
  $\forall i \in [k] : c_i \leftarrow \mathsf{Enc}(pk_i^{(h_i)}, K; r_i)$
  $x \leftarrow ((pk_i^{(h_i)})_i, (c_i)_i)$
  $w \leftarrow ((r_i)_i, K)$
  $\pi \leftarrow \mathsf{Prove}(crs, x, w)$
  $\sigma \leftarrow \mathsf{Sign}(sigk, ((c_i)_i, \pi))$
  $C \leftarrow (vk, (c_i)_i, \pi, \sigma)$.
  Return $(C, K)$.
```

```
$\mathsf{Decap}_{\mathtt{DDN}}(SK, C)$ :
  $((sk_i^{(j)})_{i,j}, PK) \leftarrow SK$
  $((pk_i^{(j)})_{i,j}, crs, \kappa) \leftarrow PK$
  $(vk, (c_i)_i, \pi, \sigma) \leftarrow C$
  If $\mathsf{SVer}(vk, ((c_i)_i, \pi), \sigma) = \bot$
                                  then return $\bot$.
  $h \leftarrow \mathsf{H}_\kappa(vk)$
  View $h$ as $(h_1 \| \dots \| h_k) \in \{0,1\}^k$.
  $x \leftarrow ((pk_i^{(h_i)})_i, (c_i)_i)$
  If $\mathsf{PVer}(crs, x, \pi) = \bot$ then return $\bot$
  Return $K \leftarrow \mathsf{Dec}(sk_1^{(h_1)}, c_1)$.
```

```
$\mathsf{F}_{\mathtt{DDN}}(PK, C, C')$ :
  $((pk_i^{(j)})_{i,j}, crs, \kappa) \leftarrow PK$
  $(vk, (c_i)_i, \pi, \sigma) \leftarrow C$
  $(vk', (c_i')_i, \pi', \sigma') \leftarrow C'$
  Return $(\mathsf{H}_\kappa(vk) \overset{?}{=} \mathsf{H}_\kappa(vk'))$.
```

```
$\mathsf{Punc}_{\mathtt{DDN}}(SK, C^*)$ :
  $((sk_i^{(j)})_{i,j}, PK) \leftarrow SK$
  $((pk_i^{(j)})_{i,j}, crs, \kappa) \leftarrow PK$
  $(vk^*, (c_i^*)_i, \pi^*, \sigma^*) \leftarrow C$
  $h^* \leftarrow \mathsf{H}_\kappa(vk^*)$
  View $h^*$ as $(h_1^* \| \dots \| h_k^*) \in \{0,1\}^k$.
  $\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$
  Return $\widehat{SK}_{C^*}$.
```

```
$\mathsf{PDecap}_{\mathtt{DDN}}(\widehat{SK}_{C^*}, C)$ :
  $(h^*, (sk_i^{(1-h_i^*)})_i, PK) \leftarrow \widehat{SK}_{C^*}$
  $((pk_i^{(j)})_{i,j}, crs, \kappa) \leftarrow PK$
  $(vk, (c_i)_i, \pi, \sigma) \leftarrow C$
  If $\mathsf{SVer}(vk, ((c_i)_i, \pi), \sigma) = \bot$ then return $\bot$.
  $h \leftarrow \mathsf{H}_\kappa(vk)$
  If $h^* = h$ then return $\bot$.
  View $h^*$ as $(h_1^* \| \dots \| h_k^*) \in \{0,1\}^k$.
  View $h$ as $(h_1 \| \dots \| h_k) \in \{0,1\}^k$.
  $\ell \leftarrow \min\{i \in [k] : h_i^* \neq h_i\}$
  $x \leftarrow ((pk_i^{(h_i)})_i, (c_i)_i)$
  If $\mathsf{PVer}(crs, x, \pi) = \bot$ then return $\bot$.
  Return $K \leftarrow \mathsf{Dec}(sk_\ell^{(1-h_\ell^*)}, c_\ell)$.
```

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.

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:

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

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 non-interactive argument system, and a UOWHF, can be found in Appendices A.1, A.3, A.4, and A.5 respectively.) 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: (The formal proofs of Lemmas 1, 2, and 3 are given in Appendices D.2, D.3, and D.4, respectively.)

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 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 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 [65], the Peikert-Waters construction [62] 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 [62, §2.3]). Moreover, the construction based on CPA secure PKE and an obfuscator for point functions (with multi-bit output) by Matsuda and Hanaoka [52] and one based on CPA secure PKE and a hash function family satisfying the strong notion (called UCE security [7]) from the same authors [53] 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 [67] (and its extension by Matsuda and Hanaoka [51]), 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 [67, 51]. Since a number of recent practical CCA secure KEMs (e.g. [18, 25, 39, 42]) 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 Appendix C.

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 [43, 50]. Therefore, if a PKEM satisfying punctured decapsulation soundness and eCPA security additionally satisfies the *unpredictability*[2] (which we recall in Appendix A.2), it can still be used as a building block in the constructions [43, 50] to obtain fully CCA secure PKE/KEM.[3]

[2] Note that we treat unpredictability and DCCA security of a detectable KEM as separate security notions (as opposed to treat the former as a requirement of the latter [43]), which we believe is more convenient to understand the connection between a PKEM and a detectable KEM.

[3] As discussed in [50], it is easy to achieve a detectable PKE scheme with DCCA security and unpredictability from a detectable KEM satisfying the same security notions, by combining the detectable KEM with a one-time secure SKE scheme (i.e. a SKE scheme which is secure under one-time encryption query).

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^*. □

On Unpredictability of PKEMs. We note that the DDN-KEM reviewed in Section 3.4 and our proposed KEM in Section 4 achieve strong unpredictabiilty (based on the security of the building blocks), which we show in Appendices E.1 and E.2, respectively.

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 $E = (\mathsf{SEnc}, \mathsf{SDec})$ 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:

$$\mathcal{F}_k := \left\{ \begin{array}{c} f_z : \mathcal{K}_k \to \mathcal{M}_k \text{ given by} \\ f_z(\alpha) := ((\mathsf{Explain}(\omega_i, \alpha))_{i \in [k+1]}, K) \end{array} \middle| \begin{array}{c} z = ((\omega_i)_{i \in [k+1]}, K) \text{ where } K \in \{0,1\}^k \\ \text{and each } \omega_i \text{ is output from } \mathsf{Fake}(1^k) \end{array} \right\} \quad (3)$$

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 D.5, D.6, and D.7, respectively.)

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.*

```
$\widehat{\mathsf{KKG}}(1^k)$ :
  $\forall (i,j) \in [k] \times \{0,1\}$ :
        $(pk_i^{(j)}, sk_i^{(j)}) \leftarrow \mathsf{PKG}(1^k)$
  $(pk_{k+1}, sk_{k+1}) \leftarrow \mathsf{PKG}(1^k)$
  $\kappa \leftarrow \mathsf{HKG}(1^k)$
  $PK \leftarrow ((pk_i^{(j)})_{i,j}, pk_{k+1}, \kappa)$
  $SK \leftarrow ((sk_i^{(j)})_{i,j}, PK)$
  Return $(PK, SK)$.
```

```
$\widehat{\mathsf{Encap}}(PK)$ :
  $((pk_i^{(j)})_{i,j}, pk_{k+1}, \kappa) \leftarrow PK$
  $\alpha \leftarrow \mathcal{K}_k$
  $((r_i)_{i \in [k+1]}, K) \leftarrow (\mathcal{R}_k)^{k+1} \times \{0,1\}^k$
  $\beta \leftarrow ((r_i)_{i \in [k+1]}, K)$
  $\widetilde{c} \leftarrow \mathsf{SEnc}(\alpha, \beta)$
  $c_{k+1} \leftarrow \mathsf{Enc}(pk_{k+1}, \alpha; r_{k+1})$
  $h \leftarrow \mathsf{H}_\kappa(c_{k+1} \| \widetilde{c})$
  View $h$ as $(h_1 \| \dots \| h_k) \in \{0,1\}^k$.
  $\forall i \in [k] : c_i \leftarrow \mathsf{Enc}(pk_i^{(h_i)}, \alpha; r_i)$
  $C \leftarrow (h, (c_i)_i, \widetilde{c})$.
  Return $(C, K)$.
```

```
$\widehat{\mathsf{Decap}}(SK, C)$ :
  $((sk_i^{(j)})_{i,j}, PK) \leftarrow SK$
  $((pk_i^{(j)})_{i,j}, pk_{k+1}, \kappa) \leftarrow PK$
  $(h, (c_i)_i, \widetilde{c}) \leftarrow C$
  View $h$ as $(h_1 \| \dots \| h_k) \in \{0,1\}^k$.
  $\alpha \leftarrow \mathsf{Dec}(sk_1^{(h_1)}, c_1)$
  If $\alpha = \bot$ then return $\bot$.
  $\beta \leftarrow \mathsf{SDec}(\alpha, \widetilde{c})$
  If $\beta = \bot$ then return $\bot$.
  $((r_i)_{i \in [k+1]}, K) \leftarrow \beta$
  $c_{k+1} \leftarrow \mathsf{Enc}(pk_{k+1}, \alpha; r_{k+1})$
  If (a) $\wedge$ (b) then return $K$ else return $\bot$:
    (a) $\forall i \in [k] : \mathsf{Enc}(pk_i^{(h_i)}, \alpha; r_i) = c_i$
    (b) $\mathsf{H}_\kappa(c_{k+1} \| \widetilde{c}) = h$
```

```
$\widehat{\mathsf{F}}(PK, C, C')$ :
  $(h, (c_i)_i, \widetilde{c}) \leftarrow C$
  $(h', (c'_i)_i, \widetilde{c}') \leftarrow C'$
  Return $(h \overset{?}{=} h')$.
```

```
$\widehat{\mathsf{Punc}}(SK, C^*)$ :
  $((sk_i^{(j)})_{i,j}, PK) \leftarrow SK$
  $(h^*, (c_i^*)_i, \widetilde{c}^*) \leftarrow C^*$
  View $h^*$ as $(h_1^* \| \dots \| h_k^*) \in \{0,1\}^k$.
  $\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$
  Return $\widehat{SK}_{C^*}$.
```

```
$\widehat{\mathsf{PDecap}}(\widehat{SK}_{C^*}, C)$ :
  $(h^*, (sk_i^{(1-h_i^*)})_i, PK) \leftarrow \widehat{SK}_{C^*}$
  $(h, (c_i)_i, \widetilde{c}) \leftarrow C$
  If $h^* = h$ then return $\bot$.
  View $h^*$ as $(h_1^* \| \dots \| h_k^*) \in \{0,1\}^k$.
  View $h$ as $(h_1 \| \dots \| h_k) \in \{0,1\}^k$.
  $\ell \leftarrow \min\{i \in [k] : h_i^* \neq h_i\}$
  $\alpha \leftarrow \mathsf{Dec}(sk_\ell^{(1-h_\ell^*)}, c_\ell)$
  Run exactly as $\widehat{\mathsf{Decap}}$ from
    the sixth step and return the result.
```

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.

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):

```
Game 1:
  $\alpha^* \leftarrow \mathcal{K}_k$;
  For $i \in [k+1]$ :
    $\underline{(pk'_i, sk'_i) \leftarrow \mathsf{PKG}(1^k);}$
    $\underline{r_i^* \leftarrow \mathcal{R}_k;}$
    $\underline{c_i^* \leftarrow \mathsf{Enc}(pk'_i, \alpha^*; r_i^*);}$
  End For
  $K_1^* \leftarrow \{0,1\}^k$;
  (Continue to the center column $\nearrow$)
$\beta^* \leftarrow ((r_i^*)_{i \in [k+1]}, K_1^*)$;
$\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, \beta^*)$;
$\kappa \leftarrow \mathsf{HKG}(1^k)$;
$h^* = (h_1^* \| \dots \| h_k^*) \leftarrow \mathsf{H}_\kappa(c_{k+1}^* \| \widetilde{c}^*)$;
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
(Continue to the right column $\nearrow$)
$PK \leftarrow ((pk_i^{(j)})_{i,j}, pk'_{k+1}, \kappa)$;
$C^* \leftarrow (h^*, (c_i^*)_i, \widetilde{c}^*)$;
$\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$;
$K_0^* \leftarrow \{0,1\}^k$;
$b \leftarrow \{0,1\}$;
$b' \leftarrow \mathcal{A}(PK, \widehat{SK}_{C^*}, C^*, K_b^*)$
```

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 $\mathtt{eCPA}$ security of the PKEM $\widehat{\Gamma}$.

Firstly, we can show that $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ is negligible due to the $\mathtt{SNC}$ security of the $(k+1)$-repetition construction Π^{k+1}, which in turn follows from the $\mathtt{SNC}$ security of the underlying SNCE scheme Π by a standard hybrid argument (see Lemma 10 in Appendix B.1).

Secondly, we can show that $|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$ is negligible due to the $\mathcal{F}$-$\mathtt{OTKDM}$ security of the SKE scheme E. Here, the key idea in this proof 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 $\mathtt{OTKDM}$ 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* $\mathtt{SNC}$ *secure SNCE scheme and a SKE scheme that is* $\mathtt{OTKDM}$ *secure with respect to efficiently computable functions, then there exist a* $\mathtt{CCA}$ *secure PKE scheme/KEM.*

Finally, it would be worth noting that our construction of a $\mathtt{CCA}$ 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 $\mathtt{eCPA}$ security is non-black-box, in the sense that our reduction algorithm needs to use the description of the $\mathsf{Explain}$ algorithm as a KDM encryption query. Such a situation was encountered in [55, 26] 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 its plaintext awareness. Specifically, in the security proofs of [55, 26], reduction algorithms need to use a "(plaintext) extractor" that is dependent on the description of a $\mathtt{CCA}$ adversary (and the building block PKE scheme for which plaintext awareness is assumed).

5 Dolev-Dwork-Naor KEM Revisited

In this section, we show that the $\mathtt{eCPA}$ 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 $\mathtt{SNC}$ secure SNCE scheme and $\mathcal{P}$ is $\mathtt{WI}$ secure, then we can still show that the PKEM $\Gamma_{\mathtt{DDN}}$ is $\mathtt{eCPA}$ 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 $\mathtt{CCA}$ 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 $\mathtt{CCA}$ 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 $\mathtt{CPA}$ secure PKE scheme and a $\mathtt{SNC}$ secure SNCE scheme is as large/small as the difference between the $\mathtt{ZK}$ security and $\mathtt{WI}$ security of a non-interactive argument system.

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

The formal proof is given in Appendix D.8. Here, we explain some intuition on the proof of Lemma 7.

Recall that in the proof based on the $\mathtt{CPA}$ security of Π and the $\mathtt{ZK}$ security of $\mathcal{P}$, we first use the $\mathtt{ZK}$ security of $\mathcal{P}$ to "cut" the relation between the components $(c_i^*)_i$ and the proof π^*, and then use the $\mathtt{CPA}$ security of the k-repetition construction Π^k (which in turn follows from the $\mathtt{CPA}$ security of Π) to "hide" the information of the challenge bit. The proof of Lemma 7 in Appendix D.8 uses the properties of the building blocks in the reversed order: we first use the $\mathtt{SNC}$ security of the k-repetition construction Π^k to generate each tuple $(pk_i^{(h_i^*)}, c_i^*, r_i^*)$ using the simulation algorithms Fake and $\mathsf{Explain}$ of the SNCE scheme Π. Because now each c_i^* can be explained as an encryption of not just K_1^* but any plaintext due to the simulation algorithms Fake and $\mathsf{Explain}$, this change "cut" the relation between the components $(c_i^*)_i$ and the proof π^*. Furthermore, due to the $\mathtt{SNC}$ security, an $\mathtt{eCPA}$ adversary cannot notice this difference from the original $\mathtt{eCPA}$ experiment. Then, we use the $\mathtt{WI}$ security of the non-interactive argument system $\mathcal{P}$ to "erase" the information of the challenge bit b. That each c_i^* can be explained as an encryption of any plaintext means that there are many witnesses for the statement $x^* = ((pk_i^{(h_i^*)})_i, (c_i^*)_i) \in L_k$, and hence the $\mathtt{WI}$ security suffices to "hide" the information on b. For more formal details, see Appendix D.8.

Acknowledgement

The authors would like thank the members of the study group "Shin-Akarui-Angou-Benkyou-Kai" and the anonymous reviewers for their invaluable comments and suggestions.

References

1. B. Applebaum. Key-dependent message security: Generic amplification and completeness. In *Proc. of EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 527–546. Springer, 2011.

2. B. Applebaum. Garbling XOR gates "for free" in the standard model. In *Proc. of TCC 2013*, volume 7785 of *LNCS*, pages 162–181. Springer, 2013.
3. B. Applebaum. Key-dependent message security: Generic amplification and completeness. *J. of Cryptology*, 27(3):429–451, 2014.
4. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Proc. of CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, 2009.
5. R. Bardou, R. Focardi, Y. Kawamoto, L. Simionato, G. Steel, and J.-K. Tsay. Efficient padding oracle attacks on cryptographic hardware. In *Proc. of CRYPTO 2012*, volume 7417 of *LNCS*, pages 608–625. Springer, 2012.
6. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Proc. of CRYPTO 1998*, volume 1462 of *LNCS*, pages 26–45. Springer, 1998.
7. M. Bellare, V.T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs. In *Proc. of CRYPTO 2013(2)*, volume 8043 of *LNCS*, pages 398–415. Springer, 2013.
8. M. Bellare, V.T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs, 2013. Updated full version of [7]. Available at http://eprint.iacr.org/2013/424/.
9. M. Bellare, V.T. Hoang, and P. Rogaway. Foundations of garbled circuits, 2012. Full version of [10]. http://eprint.iacr.org/2012/265.
10. M. Bellare, V.T. Hoang, and P. Rogaway. Foundations of garbled circuits. In *Proc. of CCS 2012*, pages 784–796. ACM, 2012.
11. M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Proc. of EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, 2009.
12. M. Bellare and A. Sahai. Non-malleable encryption: Equivalence between two notions, and indistinguishability-based characterization. In *Proc. of CRYPTO 1999*, volume 1666 of *LNCS*, pages 519–536. Springer, 1999.
13. M. Bellare and S. Yilek. Encryption schemes secure under selective opening attack, 2012. This is an updated full version of a preliminary version with Hofheinz [11]. http://eprint.iacr.org/2009/101.
14. R. Bendlin, J.B. Nielsen, P.S. Nordholt, and C. Orlandi. Lower and upper bounds for deniable public-key encryption. In *Proc. of ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 125–142. Springer, 2011.
15. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Proc. of SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, 2003.
16. D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *Proc. of CRYPTO 1998*, volume 1462 of *LNCS*, pages 1–12. Springer, 1998.
17. D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In *Proc. of CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, 2008.
18. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *Proc. of CCS 2005*, pages 320–329. ACM, 2005.
19. Z. Brakerski and S. Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *Proc. of CRYPTO 2010*, volume 6223 of *LNCS*, pages 1–20. Springer, 2010.
20. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proc. of FOCS 2001*, pages 136–145. IEEE Computer Society, 2001.
21. R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky. Deniable encryption. In *Proc. of CRYPTO 1997*, volume 1294 of *LNCS*, pages 90–104. Springer, 1997.
22. R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively secure multi-party computation. In *Proc. of STOC 1996*, pages 639–648. ACM, 1996.
23. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Proc. of EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.
24. R. Canetti, H. Krawczyk, and J.B. Nielsen. Relaxing chosen-ciphertext security. In *Proc. of CRYPTO 2003*, volume 2729 of *LNCS*, pages 565–582. Springer, 2003.
25. D. Cash, E. Kiltz, and V. Shoup. The twin Diffie-Hellman problem and applications. In *Proc. of EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 127–145. Springer, 2008.
26. D. Dachman-Soled. A black-box construction of a CCA2 encryption scheme from a plaintext aware (sPA1) encryption scheme. In *Proc. of PKC 2014*, volume 8383 of *LNCS*, pages 37–55. Springer, 2014.
27. I. Damgård and M. Jurik. A generalization, a simplification and some applications of Paillier's probabilistic public-key system. In *Proc. of PKC 2001*, volume 1992 of *LNCS*, pages 119–136. Springer, 2001.
28. I. Damgård and J.B. Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In *Proc. of CRYPTO 2000*, volume 1880 of *LNCS*, pages 432–450. Springer, 2000.
29. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proc. of STOC 1991*, pages 542–552. ACM, 1991.
30. C. Dwork, M. Naor, and O. Reingold. Immunizing encryption schemes from decryption errors. In *Proc. of EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 342–360. Springer, 2004.

31. S. Fehr, D. Hofheinz, E. Kiltz, and H. Wee. Encryption schemes secure against chosen-ciphertext selective opening attacks. In *Proc. of EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 381–402. Springer, 2010.
32. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *J. of Cryptology*, 26(1):80–101, 2013.
33. J.A. Garay, D. Wichs, and H.-S. Zhou. Somewhat non-committing encryption and efficient adaptively secure oblivious transfer. In *Proc. of CRYPTO 2009*, volume 5677 of *LNCS*, pages 505–523. Springer, 2009.
34. Y. Gertner, T. Malkin, and S. Myers. Towards a separation of semantic and CCA security for public key encryption. In *Proc. of TCC 2007*, volume 4392 of *LNCS*, pages 434–455. Springer, 2007.
35. O. Goldreich. *Foundations of Cryptography - Volume 1*. Cambridge University Press, 2001.
36. O. Goldreich. *Foundations of Cryptography - Volume 2*. Cambridge University Press, 2004.
37. O. Goldreich. Basing non-interactive zero-knowledge on (enhanced) trapdoor permutations: The state of the art. In *Studies in Complexity and Cryptography*, volume 6650 of *LNCS*, pages 406–421. Springer, 2011.
38. S. Goldwasser and S. Micali. Probabilistic encryption. *J. of Computer and System Sciences*, 28(2):270–299, 1984.
39. G. Hanaoka and K. Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In *Proc. of ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 308–325. Springer, 2008.
40. C. Hazay and A. Patra. One-sided adaptively secure two-party computation. In *Proc. of TCC 2014*, volume 8349 of *LNCS*, pages 368–393. Springer, 2014.
41. B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *Proc. of ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 70–88. Springer, 2011.
42. D. Hofheinz and E. Kiltz. Practical chosen ciphertext secure encryption from factoring. In *Proc. of EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332. Springer, 2009.
43. S. Hohenberger, A. Lewko, and B. Waters. Detecting dangerous queries: A new approach for chosen ciphertext security. In *Proc. of EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 663–681. Springer, 2012.
44. Z. Huang, S. Liu, and B. Qin. Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited. In *Proc. of PKC 2013*, volume 7778 of *LNCS*, pages 369–385. Springer, 2013.
45. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *Proc. of TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, 2006.
46. E. Kiltz, P. Mohassel, and A. O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *Proc. of EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 673–692. Springer, 2010.
47. H. Lin and S. Tessaro. Amplification of chosen-ciphertext security. In *Proc. of EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 503–519. Springer, 2013.
48. P. MacKenzie, M.K. Reiter, and K. Yang. Alternatives to non-malleability: Definitions, constructions and applications. In *Proc. of TCC 2004*, volume 2951 of *LNCS*, pages 171–190. Springer, 2004.
49. T. Malkin, I. Teranishi, and M. Yung. Efficient circuit-size independent public key encryption with KDM security. In *Proc. of EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 507–526. Springer, 2011.
50. T. Matsuda and G. Hanaoka. Achieving chosen ciphertext security from detectable public key encryption efficiently via hybrid encryption. In *Proc. of IWSEC 2013*, volume 8231 of *LNCS*, pages 226–243. Springer, 2013.
51. T. Matsuda and G. Hanaoka. Key encapsulation mechanisms from extractable hash proof systems, revisited. In *Proc. of PKC 2013*, volume 7778 of *LNCS*, pages 332–351. Springer, 2013.
52. T. Matsuda and G. Hanaoka. Chosen ciphertext security via point obfuscation. In *Proc. of TCC 2014*, volume 8349 of *LNCS*, pages 95–120. Springer, 2014.
53. T. Matsuda and G. Hanaoka. Chosen ciphertext security via UCE. In *Proc. of PKC 2014*, volume 8383 of *LNCS*, pages 56–76. Springer, 2014.
54. P. Mol and S. Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. In *Proc. of PKC 2010*, volume 6056 of *LNCS*, pages 296–311. Springer, 2010.
55. S. Myers, M. Sergi, and A. Shelat. Blackbox construction of a more than non-malleable CCA1 encryption scheme from plaintext awareness. In *Proc. of SCN 2012*, volume 7485 of *LNCS*, pages 149–165. Springer, 2012.
56. S. Myers and A. Shelat. Bit encryption is complete. In *FOCS 2009*, pages 607–616. IEEE Computer Society, 2009.
57. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proc. of STOC 1989*, pages 33–43. ACM, 1989.
58. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proc. of STOC 1990*, pages 427–437. ACM, 1990.
59. A. O'Neill, C. Peikert, and B. Waters. Bi-deniable public-key encryption. In *Proc. of CRYPTO 2011*, volume 6841 of *LNCS*, pages 525–542. Springer, 2011.

60. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proc. of EUROCRYPT 1999*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.
61. R. Pass, A. Shelat, and V. Vaikuntanathan. Relations among notions of non-malleability for encryption. In *Proc. of ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 519–535. Springer, 2007.
62. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *Proc. of STOC 2008*, pages 187–196. ACM, 2008.
63. C. Rackoff and D.R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Proc. of CRYPTO 1991*, volume 576 of *LNCS*, pages 433–444. Springer, 1992.
64. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proc. of STOC 1990*, pages 387–394. ACM, 1990.
65. A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *Proc. of TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, 2009.
66. A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proc. of STOC 2014*, pages 475–484. ACM, 2014.
67. H. Wee. Efficient chosen-ciphertext security via extractable hash proofs. In *Proc. of CRYPTO 2010*, volume 6223 of *LNCS*, pages 314–332. Springer, 2010.

A Basic Cryptographic Primitives

A.1 Public Key Encryption

A public key encryption (PKE) scheme $\mathit{\Pi}$ consists of the following three PPTAs $(\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$:

PKG**:** The key generation algorithm that takes 1^k as input, and outputs a public/secret key pair (pk, sk).
Enc**:** The encryption algorithm that takes pk and a plaintext m as input, and outputs a ciphertext c.
Dec**:** The (deterministic) decryption algorithm that takes sk and c as input, and outputs a plaintext m which could be the special symbol $\perp$ meaning "c is invalid under (pk, sk)."

Correctness. We require for all $k \in \mathbb{N}$, all (pk, sk) output by $\mathsf{PKG}(1^k)$, and all plaintexts m, it holds that $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$.

$\mathtt{CPA}$ *Security.* For a PKE scheme $\mathit{\Pi} = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define the $\mathtt{CPA}$ experiment $\mathsf{Expt}^{\mathtt{CPA}}_{\mathit{\Pi},\mathcal{A}}(k)$ as follows:

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

where it is required that $|m_0| = |m_1|$.

Definition 6. *We say that a PKE scheme $\mathit{\Pi}$ is* $\mathtt{CPA}$ *secure if for all PPTAs $\mathcal{A}$, the advantage* $\mathsf{Adv}^{\mathtt{CPA}}_{\mathit{\Pi},\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{ATK}}_{\mathit{\Pi},\mathcal{A}}(k) = 1] - 1/2|$ *is negligible.*

A.2 (Detectable) Key Encapsulation Mechanisms

A key encapsulation mechanism (KEM) $\mathit{\Gamma}$ consists of the following three PPTAs $(\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$:

KKG**:** The key generation algorithm that takes 1^k as input, and outputs a public/secret key pair (pk, sk).
Encap**:** The encapsulation algorithm that takes pk as input, and outputs a ciphertext/session-key pair (c, K).
Decap**:** The decapsulation algorithm that takes sk and c as input, and outputs a session-key K which could be the special symbol $\perp$ meaning that "c is invalid under (pk, sk)."

For simplicity but without loss of generality, in this paper the session-key space of a KEM is assumed to be $\{0,1\}^k$ when Encap and Decap are used with keys (pk, sk) output from $\mathsf{KKG}(1^k)$.

Correctness. We require for all $k \in \mathbb{N}$, all (pk, sk) output by $\mathsf{KKG}(1^k)$, and all (c, K) output by $\mathsf{Encap}(pk)$, it holds that $\mathsf{Decap}(sk, c) = K$.

Detectable KEM. In this paper, we will treat the KEM-analogue of a detectable PKE scheme [43], and thus we introduce it here.

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 the security notions (*detectable CCA security* and *unpredictability*) for a detectable KEM.

As in the case of a PKEM, intuitively, the predicate $\mathsf{F}(pk, c^*, \cdot)$ divides the ciphertext space into two classes: ciphertexts that are "close" from c^* and those that are "far" from c^*. In the spirit of [43], the predicate F indicates whether the decapsulation of ciphertexts c is "dangerous": Namely, the decapsulation of a close ciphertext c (such that $\mathsf{F}(pk, c^*, c) = 1$) may help an adversary to obtain some useful information about (the decapsulation of) the challenge ciphertext c^*.

`CPA`/`DCCA`/`CCA` *Security.* For a (detectable) KEM $\Gamma = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ and an adversary $\mathcal{A}$, we define the `CCA` experiment $\mathsf{Expt}^{\mathtt{CCA}}_{\Gamma,\mathcal{A}}(k)$ as in Fig. 5 (left), where in the experiment, $\mathcal{A}$ is not allowed to submit c^* to the oracle. We define the `DCCA` experiment $\mathsf{Expt}^{\mathtt{DCCA}}_{\Gamma,\mathcal{A}}(k)$ in the same way as the `CCA` experiment, except that $\mathcal{A}$ is not allowed to submit a query c satisfying $\mathsf{F}(pk, c^*, c) = 1$. Furthermore, the `CPA` experiment $\mathsf{Expt}^{\mathtt{CPA}}_{\Gamma,\mathcal{A}}(k)$ is also defined similarly to the `CCA` experiment, except that $\mathcal{A}$ is not allowed to submit any query.

Definition 7. *Let* $\mathtt{ATK} \in \{\mathtt{CPA}, \mathtt{DCCA}, \mathtt{CCA}\}$. *We say that a (detectable) KEM* Γ *is* `ATK` *secure if for all PPTAs* $\mathcal{A}$*, the advantage* $\mathsf{Adv}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k) = 1] - 1/2|$ *is negligible.*

Unpredictability of a Detectable KEM. Here, we recall the definition of unpredictability and strong unpredictability of a detectable KEM, which are straightforward KEM-analogues of those of a detectable PKE scheme defined by Hohenberger et al. [43]. (The ordinary (i.e. non-strong) version of unpredictability of a detectable KEM was also defined in [50].) Informally, this security notion requires that it is hard to find a ciphertext c' such that it is "close" to any unseen ciphertext c^* (i.e. $\mathsf{F}(pk, c^*, c') = 1$).

Formally, for a detectable KEM Γ and an adversary $\mathcal{A}$, consider the `UNP` experiment $\mathsf{Expt}^{\mathtt{UNP}}_{\Gamma,\mathcal{A}}(k)$ and the `sUNP` experiment $\mathsf{Expt}^{\mathtt{sUNP}}_{\Gamma,\mathcal{A}}(k)$ as in Fig. 5 (center and right, respectively).

Definition 8. *We say that a detectable KEM* Γ *satisfies* unpredictability *(resp.* strong unpredictability*) if for all PPTAs* $\mathcal{A}$*, the advantage* $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{UNP}}_{\Gamma,\mathcal{A}}(k) = 1]$ *(resp.* $\mathsf{Adv}^{\mathtt{sUNP}}_{\Gamma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{sUNP}}_{\Gamma,\mathcal{A}}(k) = 1]$*) is negligible.*

Like the security notions of a PKEM, in order to use a detectable KEM as a building block to construct fully `CCA` secure PKE/KEM via the constructions of [43, 50], the ordinary unpredictability suffices. However, the strong version is usually easier to work with.

A.3 Signature

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

SKG**:** The key generation algorithm that takes 1^k as input, and outputs a verification/signing key pair $(vk, sigk)$.

Sign**:** The signing algorithm that takes $sigk$ and a message m as input, and outputs a signature σ.

SVer**:** The verification algorithm that takes vk and a message/signature pair (m, σ) as input, and outputs either $\top$ (meaning "accept") or $\bot$ (meaning "reject").

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

```
$\mathsf{Expt}^{\mathtt{UNP}}_{\Gamma,\mathcal{A}}(k)$ :
  $(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
  $c' \leftarrow \mathcal{A}^{\mathsf{Decap}(sk,\cdot)}(pk)$
  $(c^*, K^*) \leftarrow \mathsf{Encap}(pk)$
  Return $\mathsf{F}(pk, c^*, c')$.
```

```
$\mathsf{Expt}^{\mathtt{sUNP}}_{\Gamma,\mathcal{A}}(k)$ :
  $(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
  $c' \leftarrow \mathcal{A}(pk, sk)$
  $(c^*, K^*) \leftarrow \mathsf{Encap}(pk)$
  Return $\mathsf{F}(pk, c^*, c')$.
```

Fig. 5. Security experiments for a (detectable/ordinary) KEM.

Correctness. We require for all $k \in \mathbb{N}$, all $(vk, sigk)$ output by $\mathsf{SKG}(1^k)$, and all messages m, it holds that $\mathsf{SVer}(vk, m, \mathsf{Sign}(sigk, m)) = \top$.

Strong One-time Unforgeability. Here we recall the strong unforgeability under one-time chosen message attacks (SOT security, for short).

Definition 9. *We say that a signature scheme $\Sigma = (\mathsf{SKG}, \mathsf{Sign}, \mathsf{SVer})$ is strongly unforgeable under one-time chosen message attacks (*SOT *secure, for short), if for all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage $\mathsf{Adv}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k) = 1]$ is negligible, where the experiment $\mathsf{Expt}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k)$ is defined as follows:*

$$\begin{aligned}\mathsf{Expt}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k) : [\ &(vk, sigk) \leftarrow \mathsf{SKG}(1^k);\ (m, \mathsf{st}) \leftarrow \mathcal{A}_1(vk);\ \sigma \leftarrow \mathsf{Sign}(sigk, m);\\ &(m', \sigma') \leftarrow \mathcal{A}_2(\mathsf{st}, \sigma);\ \text{Return } 1 \text{ iff } \mathsf{SVer}(vk, m', \sigma') = \top \wedge (m', \sigma') \neq (m, \sigma)\ ].\end{aligned}$$

A SOT secure signature scheme can be built from any one-way function [57, 64].

A.4 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 following three algorithms $(\mathsf{CRSG}, \mathsf{Prove}, \mathsf{PVer})$:

CRSG**:** The common reference string (CRS) generation algorithm that takes 1^k as input, and outputs a common reference string crs. We assume that crs implicitly contains the information on k, and specifies the set L_k of statements whose validity can be proved and verified by the following algorithms.

Prove**:** The prover algorithm that takes crs, a statement $x \in L_k$, and a witness w for the fact that $x \in L_k$ as input, and outputs a proof π.

PVer**:** The verification algorithm that takes crs, and a statement/proof pair $(x, \pi) \in \{0,1\}^* \times \{0,1\}^*$ as input, and outputs either $\top$ (meaning "accept") or $\bot$ (meaning "reject").

Correctness. We require perfect correctness for non-interactive argument systems: for all $k \in \mathbb{N}$, all $crs \leftarrow \mathsf{CRSG}(1^k)$, 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 $\mathsf{PVer}(crs, x, \mathsf{Prove}(crs, x, w)) = \top$.

Security Definitions of Non-interactive Argument Systems. Here, we recall the basic security definitions for a non-interactive argument system: *adaptive soundness*, *witness indistinguishability*, and *zero-knowledge*.

We first recall the definition of *adaptive soundness*. We note that in our proposed construction, we need the adaptive soundness in which the (false) statement x output by an adversary can depend on a common reference string crs.

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

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

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

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

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

Definition 10. *We say that a non-interactive argument system $\mathcal{P}$ for a language L satisfies* adaptive soundness *if for all PPTAs $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathtt{Sound}}_{\mathcal{P},\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{Sound}}_{\mathcal{P},\mathcal{A}}(k) = 1]$ is negligible, where the* `Sound` *experiment $\mathsf{Expt}^{\mathtt{Sound}}_{\mathcal{P},\mathcal{A}}(k)$ is defined as in Fig. 6 (leftmost).*

We next recall the definition of *witness indistinguishability* (`WI` security, for short). 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.

Definition 11. *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 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage $\mathsf{Adv}^{\mathtt{WI}}_{\mathcal{P},\mathcal{A}}(k)$ 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.*

Finally, we recall the definition of the *zero-knowledge property* (`ZK` security, for short). Again as in `WI` security, in this paper we do not need "adaptive" version of the `ZK` security in which a statement (and a witness) dependent on a common reference string is taken into account.

Definition 12. *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:)** $(\mathsf{SimCRS}, \mathsf{SimPrv})$ has the following interface:*
  SimCRS**:** *This algorithm is the "simulated common reference string" generation algorithm that takes 1^k as input, and outputs crs and a corresponding trapdoor td.*
  SimPrv**:** *This algorithm 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 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage $\mathsf{Adv}^{\mathtt{ZK}}_{\mathcal{P},\mathcal{S},\mathcal{A}}(k) := |\Pr[\mathsf{Expt}^{\mathtt{ZK\text{-}Real}}_{\mathcal{P},\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{ZK\text{-}Sim}}_{\mathcal{P},\mathcal{S},\mathcal{A}}(k) = 1]|$ 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.*

A.5 Universal One-Way Hash Functions

Here, we recall the definition of a universal one-way hash function (UOWHF) [57].

Definition 13. *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:*

- ***(Syntax:)*** *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* $\mathsf{H}_\kappa : \{0,1\}^* \to \{0,1\}^k$.
- ***(Universal One-wayness:)*** *For all PPTAs* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, *the advantage* $\mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{A}}(k) = 1]$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{A}}(k)$ *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 [57, 64].

B Some Useful Facts

In this section, we review several useful facts used in this paper.

B.1 Useful Facts on (Sender Non-committing) Public Key Encryption

Extending the Plaintext Space by Concatenation. The plaintext space of a SNCE scheme can be extended by considering a simple contatenation.

Formally, let $\mathit{\Pi} = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Explain})$ be a SNCE scheme whose plaintext space is $\{0,1\}$ and whose randomness space (of Enc for security parameter k) is $\mathcal{R}_k$, and let $n = n(k)$ be a positive polynomial. Then a simple n-wise "concatenation" construction $\mathit{\Pi}^{\|n} = (\mathsf{PKG}^{\|n}, \mathsf{Enc}^{\|n}, \mathsf{Dec}^{\|n}, \mathsf{Fake}^{\|n}, \mathsf{Explain}^{\|n})$ given in Fig. 7 (left) is a SNCE scheme whose plaintext space is $\{0,1\}^n$.

The security of the n-wise concatenation construction is guaranteed by the following lemmas (which can be proved by applying a standard hybrid argument, and thus omitted).

Lemma 8. *Let* n *be a positive polynomial. If the SNCE scheme* $\mathit{\Pi}$ *is* $\mathtt{SNC}$ *secure, then so is the* n*-wise concatenation construction* $\mathit{\Pi}^{\|n}$. *In particular, for any positive polynomial* n *and any PPTA* $\mathcal{A}$, *there exists a PPTA* $\mathcal{B}$ *such that* $\mathsf{Adv}^{\mathtt{SNC}}_{\mathit{\Pi}^{\|n},\mathcal{A}}(k) \leq n \cdot \mathsf{Adv}^{\mathtt{SNC}}_{\mathit{\Pi},\mathcal{B}}(k)$.

Repetition Construction. It is a well-known fact that the $\mathtt{CPA}$ security of a PKE scheme is preserved even if we encrypt a same plaintext under multiple independently generated public keys. Similarly, the $\mathtt{SNC}$ security of a SNCE scheme is preserved even if we encrypt a same plaintext under multiple independently generated public keys.

Formally, let $\mathit{\Pi} = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Explain})$ be a SNCE scheme whose randomness space (of Enc for security parameter k) is $\mathcal{R}_k$, and let $n = n(k)$ be a positive polynomial. Then the n-repetition construction $\mathit{\Pi}^n = (\mathsf{PKG}^n, \mathsf{Enc}^n, \mathsf{Dec}^n, \mathsf{Fake}^n, \mathsf{Explain}^n)$ based on $\mathit{\Pi}$ is as in Fig. 7 (right). (For an ordinary PKE scheme, we do not consider simulation algorithms Fake and $\mathsf{Explain}$.)

The security of the n-repetition construction is formally stated by the following lemmas (which can be proved by applying a standard hybrid argument, and thus omitted).

Lemma 9. *Let* n *be a positive polynomial. If* $\mathit{\Pi}$ *is a* $\mathtt{CPA}$ *secure PKE scheme, then so is the* n*-repetition construction* $\mathit{\Pi}^n$. *In particular, for any positive polynomial* n *and any PPTA* $\mathcal{A}$, *there exists a PPTA* $\mathcal{B}$ *such that* $\mathsf{Adv}^{\mathtt{CPA}}_{\mathit{\Pi}^n,\mathcal{A}}(k) \leq n \cdot \mathsf{Adv}^{\mathtt{CPA}}_{\mathit{\Pi},\mathcal{B}}(k)$.

Lemma 10. *Let* n *be a positive polynomial. If* $\mathit{\Pi}$ *is a* $\mathtt{SNC}$ *secure SNCE scheme, then so is the* n*-repetition construction* $\mathit{\Pi}^n$. *In particular, for any positive polynomial* n *and any PPTA* $\mathcal{A}$, *there exists a PPTA* $\mathcal{B}$ *such that* $\mathsf{Adv}^{\mathtt{SNC}}_{\mathit{\Pi}^n,\mathcal{A}}(k) \leq n \cdot \mathsf{Adv}^{\mathtt{SNC}}_{\mathit{\Pi},\mathcal{B}}(k)$.

```
$\mathsf{PKG}^{\|n}(1^k)$ :
  $\forall i \in [n] : (pk_i, sk_i) \leftarrow \mathsf{PKG}(1^k)$
  $PK \leftarrow (pk_i)_{i \in [n]}$
  $SK \leftarrow (sk_i)_{i \in [n]}$
  Return $(PK, SK)$.
$\mathsf{Enc}^{\|n}(PK, m; R)$ :
  $(pk_i)_{i \in [n]} \leftarrow PK$
  Parse $m$ as $(m_1, \dots, m_n) \in \{0,1\}^n$.
  Parse $R$ as $(r_1, \dots, r_n) \in (\mathcal{R}_k)^n$.
  $\forall i \in [n] : c_i \leftarrow \mathsf{Enc}(pk_i, m_i; r_i)$
  Return $C \leftarrow (c_i)_{i \in [n]}$.
$\mathsf{Dec}^{\|n}(SK, C)$ :
  $(sk_i)_{i \in [n]} \leftarrow SK$
  $(c_i)_{i \in [n]} \leftarrow C$
  $\forall i \in [n] : m_i \leftarrow \mathsf{Dec}(sk_i, c_i)$
  If $\exists i$ s.t. $m_i = \bot$ then return $\bot$.
  Return $m \leftarrow (m_1 \| \dots \| m_n)$.
$\mathsf{Fake}^{\|n}(1^k)$ :
  $\forall i \in [n] : (pk_i, c_i, \omega_i) \leftarrow \mathsf{Fake}(1^k)$
  $PK \leftarrow (pk_i)_{i \in [n]}$
  $C \leftarrow (c_i)_{i \in [n]}$
  $W \leftarrow (\omega_i)_{i \in [n]}$
  Return $(PK, C, W)$.
$\mathsf{Explain}^{\|n}(W, m)$ :
  $(\omega_i)_{i \in [n]} \leftarrow W$
  Parse $m$ as $(m_1, \dots, m_n) \in \{0,1\}^n$.
  $\forall i \in [n] : r_i \leftarrow \mathsf{Explain}(\omega_i, m_i)$
  Return $R \leftarrow (r_i)_{i \in [n]}$.
```

```
$\mathsf{PKG}^{n}(1^k)$ :
  $\forall i \in [n] : (pk_i, sk_i) \leftarrow \mathsf{PKG}(1^k)$
  $PK \leftarrow (pk_i)_{i \in [n]}$
  $SK \leftarrow (sk_i)_{i \in [n]}$
  Return $(PK, SK)$.
$\mathsf{Enc}^{n}(PK, m; R)$ :
  $(pk_i)_{i \in [n]} \leftarrow PK$
  Parse $R$ as $(r_1, \dots, r_n) \in (\mathcal{R}_k)^n$.
  $\forall i \in [n] : c_i \leftarrow \mathsf{Enc}(pk_i, m; r_i)$
  Return $C \leftarrow (c_i)_{i \in [n]}$.
$\mathsf{Dec}^{n}(SK, C)$ :
  $(sk_i)_{i \in [n]} \leftarrow SK$
  $(c_i)_{i \in [n]} \leftarrow C$
  $\forall i \in [n] : m_i \leftarrow \mathsf{Dec}(sk_i, c_i)$
  If $m_1 = \cdots = m_k$
    then return $m_1$ else return $\bot$.
$\mathsf{Fake}^{n}(1^k)$ :
  $\forall i \in [n] : (pk_i, c_i, \omega_i) \leftarrow \mathsf{Fake}(1^k)$
  $PK \leftarrow (pk_i)_{i \in [n]}$
  $C \leftarrow (c_i)_{i \in [n]}$
  $W \leftarrow (\omega_i)_{i \in [n]}$
  Return $(PK, C, W)$.
$\mathsf{Explain}^{n}(W, m)$ :
  $(\omega_i)_{i \in [n]} \leftarrow W$
  $\forall i \in [n] : r_i \leftarrow \mathsf{Explain}(\omega_i, m)$
  Return $R \leftarrow (r_i)_{i \in [n]}$.
```

Fig. 7. The n-wise concatenation construction $\Pi^{\|n}$ (left) and the n-repetition construction Π^n (right) of a PKE/SNCE scheme based on a base PKE/SNCE scheme Π.

B.2 Statistical Properties of Basic Primitives

To show the strong unpredictability of the PKEMs in Appendix E, we use the following simple statistical properties of a SOT signature scheme and a UOWHF (where the formal definitions of these primitives and their security can be found in Appendices A.3 and A.5, respectively).

Statistical Property of a SOT *Secure Signature Scheme.* The following lemma states that if a signature scheme is SOT secure, then it is information-theoretically hard to guess an "unseen" verification key.

Lemma 11. *Let $\Sigma = (\mathsf{SKG}, \mathsf{Sign}, \mathsf{SVer})$ be a* SOT *secure signature scheme. Then, the following quantity (which we call the* smoothness *of verification keys of Σ):*

$$\mathsf{Smth}_{\Sigma}(k) := \max_{h \in \{0,1\}^*} \Pr[(vk, sigk) \leftarrow \mathsf{SKG}(1^k) : vk = h]$$

is negligible.

Proof of Lemma 11. For $k \in \mathbb{N}$ and a string $h \in \{0,1\}^*$, let $P_k(h) = \Pr[(vk, sigk) \leftarrow \mathsf{SKG}(1^k) : vk = h]$. Furthermore, let h_k^* be a string such that $P_k(h_k^*) \geq P_k(h)$ for any string $h \in \{0,1\}^*$. (If there are more than one such h_k^*, then choose the lexicographically smallest.) Note that $\mathsf{Smth}_{\Sigma}(k) = P_k(h_k^*)$.

Consider a SOT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against Σ that is defined as follows:

$\mathcal{A}_1(vk):$ $\mathcal{A}_1$ picks any message m, and runs $(vk', sigk') \leftarrow \mathsf{SKG}(1^k)$. If $vk' = vk$, then $\mathcal{A}_1$ sets $\mathsf{st} \leftarrow$ ($\mathcal{A}_1$'s entire view). Otherwise (i.e. $vk' \neq vk$), then $\mathcal{A}_1$ prepares a state information st that tells $\mathcal{A}_2$ that $\mathcal{A}_1$ has given up. Finally, $\mathcal{A}_1$ terminates with output (m, st).

$\mathcal{A}_2(\mathsf{st}, \sigma):$ $\mathcal{A}_2$ first checks whether $\mathcal{A}_1$ has given up by looking at st, and aborts if this is the case. Otherwise, $\mathcal{A}_2$ picks any message $m' \neq m$, runs $\sigma' \leftarrow \mathsf{Sign}(sigk', m')$, and terminates with output (m', σ').

The above completes the description of $\mathcal{A}$. Note that $\mathcal{A}$ is a PPTA, and if (and only if) $vk' = vk$ occurs $\mathcal{A}$ succeeds in outputting a forged message/signature pair due to the correctness of the signature scheme Σ. Therefore, $\mathcal{A}$'s $\mathtt{SOT}$ advantage can be calculated as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k) &= \Pr[(vk, sigk), (vk', sigk') \leftarrow \mathsf{SKG}(1^k) : vk' = vk] \\
&\geq \Pr[(vk, sigk), (vk', sigk') \leftarrow \mathsf{SKG}(1^k) : vk = h^*_k \wedge vk' = h^*_k] \\
&= \Big(\Pr[(vk, sigk) \leftarrow \mathsf{SKG}(1^k) : vk = h^*_k] \Big)^2 \\
&= P_k(h^*_k)^2 = \mathsf{Smth}_{\Sigma}(k)^2,
\end{aligned}$$

Therefore, we have

$$\mathsf{Smth}_{\Sigma}(k) \leq \sqrt{\mathsf{Adv}^{\mathtt{SOT}}_{\Sigma,\mathcal{A}}(k)}.$$

Since $\mathcal{A}$ is a PPTA and Σ is $\mathtt{SOT}$ secure, the right hand side of the above inequality is negligible. This completes the proof of Lemma 11. □

Statistical Property of a UOWHF. The following lemma says that if the input of a UOWHF is chosen according to a distribution with high min-entropy, then it is information-theoretically hard to guess the evaluation result, even if a hash index is given in advance.

Lemma 12. *Let $\mathcal{H} = (\mathsf{HKG}, \mathsf{H})$ be a UOWHF, and let X be an efficiently samplable distribution such that $\mathbf{H}_\infty(X) \in \omega(\log k)$. Let $\mathcal{F}_k$ be the set of all functions of the form $F : [\mathsf{HKG}(1^k)] \rightarrow \{0,1\}^k$ (where "$[\mathsf{HKG}(1^k)]$" denotes the support of $\mathsf{HKG}(1^k)$ and F may not be efficiently computable). Then, the following quantity (which we call the smoothness of $\mathcal{H}$)*

$$\mathsf{Smth}_{\mathcal{H}}(k) := \max_{F \in \mathcal{F}_k} \Pr[\kappa \leftarrow \mathsf{HKG}(1^k); x \leftarrow X : \mathsf{H}_\kappa(x) = F(\kappa)]$$

is negligible. (Here, two functions in $\mathcal{F}_k$ having the same input-output behavior are identified as a single function.)

Proof of Lemma 12. For each $k \in \mathbb{N}$ and $F \in \mathcal{F}_k$, let $P_k(F) = \Pr[\kappa \leftarrow \mathsf{HKG}(1^k); x \leftarrow X : \mathsf{H}_\kappa(x) = F(\kappa)]$. Furthermore, let $F^*_k \in \mathcal{F}_k$ be a function such that $P_k(F^*_k) \geq P_k(F)$ for any function $F \in \mathcal{F}_k$. (If there are multiple such functions that maximize $P_k(F_k)$, choose the "first" one in some canonical ordering of the functions in $\mathcal{F}_k$.) Note that $\mathsf{Smth}_{\mathcal{H}}(k) = P_k(F^*_k)$.

Consider a $\mathtt{UOW}$ adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against $\mathcal{H}$ such that each $\mathcal{A}_i$ picks $x_i \leftarrow X$ and simply outputs x_i. Since X is efficiently samplable, $\mathcal{A}$ is a PPTA. Then, $\mathcal{A}$'s $\mathtt{UOW}$ advantage can be calculated as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{A}}(k) &= \Pr[x_1, x_2 \leftarrow X; \kappa \leftarrow \mathsf{HKG}(1^k) : \mathsf{H}_\kappa(x_1) = \mathsf{H}_\kappa(x_2) \wedge x_1 \neq x_2] \\
&= \Pr[x_1, x_2 \leftarrow X; \kappa \leftarrow \mathsf{HKG}(1^k) : \mathsf{H}_\kappa(x_1) = \mathsf{H}_\kappa(x_2)] - \Pr[x_1, x_2 \leftarrow X : x_1 = x_2] \\
&\geq \Pr[x_1, x_2 \leftarrow X; \kappa \leftarrow \mathsf{HKG}(1^k) : \mathsf{H}_\kappa(x_1) = \mathsf{H}_\kappa(x_2)] - 2^{-\mathbf{H}_\infty(X)} \\
&\geq \Pr[x_1, x_2 \leftarrow X; \kappa \leftarrow \mathsf{HKG}(1^k) : \mathsf{H}_\kappa(x_1) = F^*_k(\kappa) \wedge \mathsf{H}_\kappa(x_2) = F^*_k(\kappa)] - 2^{-\mathbf{H}_\infty(X)} \\
&= \mathop{\mathbf{E}}_{\kappa \leftarrow \mathsf{HKG}(1^k)}\Big[\Pr[x_1, x_2 \leftarrow X : \mathsf{H}_\kappa(x_1) = F^*_k(\kappa) \wedge \mathsf{H}_\kappa(x_2) = F^*_k(\kappa)] \Big] - 2^{-\mathbf{H}_\infty(X)} \\
&\overset{(*)}{=} \mathop{\mathbf{E}}_{\kappa \leftarrow \mathsf{HKG}(1^k)}\Big[\Big(\Pr[x \leftarrow X : \mathsf{H}_\kappa(x) = F^*_k(\kappa)] \Big)^2 \Big] - 2^{-\mathbf{H}_\infty(X)} \\
&\overset{(\dagger)}{\geq} \Big(\mathop{\mathbf{E}}_{\kappa \leftarrow \mathsf{HKG}(1^k)}\Big[\Pr[x \leftarrow X : \mathsf{H}_\kappa(x) = F^*_k(\kappa)] \Big] \Big)^2 - 2^{-\mathbf{H}_\infty(X)} \\
&= \Big(\Pr[\kappa \leftarrow \mathsf{HKG}(1^k); x \leftarrow X : \mathsf{H}_\kappa(x) = F^*_k(\kappa)] \Big)^2 - 2^{-\mathbf{H}_\infty(X)} \\
&= P_k(F^*_k)^2 - 2^{-\mathbf{H}_\infty(X)} \\
&= \mathsf{Smth}_{\mathcal{H}}(k)^2 - 2^{-\mathbf{H}_\infty(X)},
\end{aligned}$$

where the inequality (*) follows from the fact that the events "$\mathsf{H}_\kappa(x_1) = F^*_k(\kappa)$" and "$\mathsf{H}_\kappa(x_2) = F^*_k(\kappa)$" become independent once κ is fixed, and the inequality (†) is due to the Jensen's inequality[4]. Therefore, we have

$$\mathsf{Smth}_{\mathcal{H}}(k) \leq \sqrt{\mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{A}}(k) + 2^{-\mathbf{H}_\infty(X)}}.$$

Since $\mathcal{H}$ is a UOWHF, $\mathcal{A}$ is a PPTA, and $\mathbf{H}_\infty(X) \in \omega(\log k)$, the right hand side of the above inequality is negligible. This completes the proof of Lemma 12. □

C On Capturing ABO-XHPS-Based KEMs via Puncturable KEM

Recall that an ABO-XHPS [67] is a special kind of a designated-verifier zero-knowledge proof of knowledge system for a family of "one-way" relations (one-way relation family) $\mathcal{R}$ defined over some set $\mathcal{U} \times \mathcal{S}$ such that given $u \in \mathcal{U}$, it is hard to find $s \in \mathcal{S}$ (where we can generate a proof such that "I know the answer s to the problem u"). Proofs produced from ABO-XHPS are "tag-based." An ABO-XHPS has a normal mode (called "extraction mode" in [67]) and a simulation mode (called "all-but-one" mode in [67]), where the former is used for normal operations of a KEM and the latter is used in the security proof, and each mode has its own key generation to generate a public/secret key pair. A normal key pair (pk, sk) can be generated if one knows a private parameter of a one-way relation family with which ABO-XHPS is associated. Given sk, an instance u of a one-way relation, and a valid proof π that proves that "I know the answer s corresponding to u", one can extract s. (In the ABO-XHPS-based KEM [67], a ciphertext consists of u and π, and s is used as a "seed" of a session-key.) Even if one does not know a private parameter of a one-way relation family, one can generate a "simulated" public/secret key pair $(pk, \widehat{sk}_{\mathsf{tag}^*})$ for any tag tag^*, so that pk is statistically indistinguishable from one in the normal mode. The simulation secret key $\widehat{sk}_{\mathsf{tag}^*}$ can be used to extract an answer s from an instance/proof pair (u, π), except that it cannot extract s from proofs generated under

[4] If X is a random variable and f is a convex function, then $\mathbf{E}[f(X)] \geq f(\mathbf{E}[X])$. We use $f(x) = x^2$.

tag^*, while $\widehat{sk}_{\mathsf{tag}^*}$ can be used to generate a fake proof π^* which looks like a valid proof under tag^* for any instance u.

In [67], Wee showed how to use an ABO-XHPS and its associated one-way relation to construct a CCA secure KEM. The security proof in [67] (and the other proof with slightly different assumptions on one-way relation family and ABO-XHPS in [51]) follows a line very similar to the security proof of a CCA security of a PKEM.

Here, the obstacle that prevents Wee's KEM to be interpreted as a PKEM is that two different modes in an ABO-XHPS, namely the normal mode and the simulation mode, have separate setups. Thus, if an ABO-XHPS has the additional property that enables us to generate a simulation secret key $\widehat{sk}_{\mathsf{tag}^*}$ for any tag tag^* from a normal secret key sk in such a way that (1) the public key pk corresponding to sk is not changed, and (2) $\widehat{sk}_{\mathsf{tag}^*}$ generated from sk and that generated by the "conventional" simulation set up as defined in the original ABO-XHPS are statistically close, and if ABO-XHPS has the computational security property called *computational soundness* (defined in [51]), then we can capture an ABO-XHPS-based KEM and its CCA security as a PKEM having the three properties introduced in Section 3.2. (Specifically, strong decapsulation soundness will follow from the security of a building block UOWHF (or, equivalently, the security of a target collision resistant hash function), the (ordinary) decapsulation soundness will follow from the computational soundness of an ABO-XHPS (with the above mentioned additional property), and the eCPA security will follow from the security of a one-way relation family.) Interestingly, as far as we know, all existing ABO-XHPS defined/explained in [67, 51] with computational soundness have (or can be modified to have) the above mentioned additional property, and hence we believe that this property is quite natural.

Formalizing the above discussion requires some effort and is beyond our scope (the main purpose in this paper is to show new generic constructions of CCA secure KEMs from general cryptographic assumptions), and hence we would like to leave it as our future work.

D Postponed Proofs

D.1 Proof of Theorem 1: CCA Security of a PKEM

In the following we show the first part of the proof of Theorem 1 (i.e. the equation (1)). The proof for the second part (i.e. the tight version in the equation (2) using strong punctured decapsulation soundness) is almost the same as that of the first part, and thus we only explain the difference at the end of this subsection.

Let $\mathcal{A}$ be any PPTA adversary that attacks the CCA security of the KEM Γ^* and makes in total $Q = Q(k)(> 0)$ decapsulation queries. Then, consider the following sequence of games:

Game 1: This is the 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 $\bot$.

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). Using the above notation, $\mathcal{A}$'s CCA advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma^*,\mathcal{A}}(k) = 2 \cdot |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \le 2 \cdot \sum_{i \in [2]} |\Pr[\mathsf{Succ}_i] - \Pr[\mathsf{Succ}_{i+1}]| + 2 \cdot |\Pr[\mathsf{Succ}_3] - \frac{1}{2}|. \quad (4)$$

In the following, we show an upperbound of each term in the right hand side of the above inequality.

Claim 1 *There exists a PPTA $\mathcal{B}_{\mathsf{d}}$ such that* $\mathsf{Adv}^{\mathtt{DSND}}_{\Gamma,\mathcal{B}_{\mathsf{d}}}(k) \geq |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$.

Proof of Claim 1. For $i \in \{1, 2\}$, let Valid_i be the event that in Game i, $\mathcal{A}$ submits at least one decapsulation query c satisfying $\mathsf{F}(pk, c^*, c) = 1$ and $\mathsf{Decap}(sk, c) \neq \perp$. Game 1 and Game 2 proceed identically unless $\mathcal{A}$ makes such a query, and hence we have

$$|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| \leq \Pr[\mathsf{Valid}_1] = \Pr[\mathsf{Valid}_2].$$

Now we show that we can construct a PPTA adversary $\mathcal{B}_{\mathsf{d}}$ that attacks the decapsulation soundness of the PKEM Γ with advantage $\mathsf{Adv}^{\mathtt{DSND}}_{\Gamma,\mathcal{B}_{\mathsf{d}}}(k) = \Pr[\mathsf{Valid}_1]$, which, combined with the above inequality, proves the claim. The description of $\mathcal{B}_{\mathsf{d}}$ is as follows:

$\mathcal{B}_{\mathsf{d}}^{\mathsf{Decap}(sk,\cdot)}(pk, c^*, K^*)$**:** $\mathcal{B}_{\mathsf{d}}$ sets $K_1^* \leftarrow K^*$, picks $K_0^* \in \{0, 1\}^k$ and $b \in \{0, 1\}$ uniformly at random, and runs $b' \leftarrow \mathcal{A}^{\mathsf{Decap}(sk,\cdot)}(pk, c^*, K_b^*)$, where $\mathcal{B}_{\mathsf{d}}$ uses $\mathcal{B}_{\mathsf{d}}$'s own decapsulation oracle to answer to $\mathcal{A}$'s decapsulation queries. When $\mathcal{A}$ terminates, $\mathcal{B}_{\mathsf{d}}$ checks whether $\mathcal{A}$ has submitted a query c satisfying $\mathsf{F}(pk, c^*, c) = 1$ and $\mathsf{Decap}(sk, c) \neq \perp$, which can be checked by $\mathcal{B}_{\mathsf{d}}$'s oracle and using F. If $\mathcal{A}$ has submitted such a query, then $\mathcal{B}_{\mathsf{d}}$ outputs one of such queries and terminates. Otherwise, $\mathcal{B}_{\mathsf{d}}$ simply gives up and aborts.

The above completes the description of $\mathcal{B}_{\mathsf{d}}$. It is easy to see that $\mathcal{B}_{\mathsf{d}}$ perfectly simulates Game 1 for $\mathcal{A}$. Therefore, the probability that $\mathcal{A}$ submits a decapsulation query c satisfying $\mathsf{F}(pk, c^*, c) = 1$ and $\mathsf{Decap}(sk, c) \neq \perp$ is exactly $\Pr[\mathsf{Valid}_1]$. Moreover, recall that $\mathcal{A}$'s queries c must be different from c^* according to the rule of the $\mathtt{CCA}$ experiment. Thus, if $\mathcal{A}$'s query c satisfies the conditions of Valid_1, then the query additionally satisfies $c \neq c^*$. This means that all conditions that make $\mathcal{B}_{\mathsf{d}}$'s $\mathtt{DSND}$ experiment return 1 are satisfied. Furthermore, whenever $\mathcal{A}$ submits such a query, $\mathcal{B}_{\mathsf{d}}$ can always find such a query by using its oracle and F. Hence, we have $\mathsf{Adv}^{\mathtt{DSND}}_{\Gamma,\mathcal{B}_{\mathsf{d}}}(k) = \Pr[\mathsf{Valid}_1]$. This completes the proof of Claim 1. □

Claim 2 *There exists a PPTA $\mathcal{B}_{\mathsf{a}}$ such that* $\mathsf{Adv}^{\mathtt{PDSND}}_{\Gamma,\mathcal{B}_{\mathsf{a}}}(k) \geq (1/Q) \cdot |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$.

Proof of Claim 2. For $i \in \{2, 3\}$, let Diff_i be the event that in Game i, $\mathcal{A}$ submits at least one decapsulation query c satisfying $\mathsf{F}(pk, c^*, c) = 0$ and $\mathsf{Decap}(sk, c) \neq \mathsf{PDecap}(\widehat{sk}_{c^*}, c)$. Game 2 and Game 3 proceed identically unless $\mathcal{A}$ makes such a query, and hence we have

$$|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]| \leq \Pr[\mathsf{Diff}_2] = \Pr[\mathsf{Diff}_3].$$

Now we show that we can construct a PPTA adversary $\mathcal{B}_{\mathsf{a}}$ that attacks the punctured decapsulation soundness of the PKEM Γ with advantage $\mathsf{Adv}^{\mathtt{PDSND}}_{\Gamma,\mathcal{B}_{\mathsf{a}}}(k) \geq (1/Q) \cdot \Pr[\mathsf{Diff}_3]$, which, combined with the above inequality, proves the claim. The description of $\mathcal{B}_{\mathsf{a}}$ is as follows:

$\mathcal{B}_{\mathsf{a}}^{\mathsf{PDecap}(\widehat{sk}_{c^*},\cdot)}(pk, c^*, K^*)$**:** $\mathcal{B}_{\mathsf{a}}$ sets $K_1^* \leftarrow K^*$, picks $K_0^* \in \{0, 1\}^k$ and $b \in \{0, 1\}$ uniformly at random, and runs $b' \leftarrow \mathcal{A}(pk, c^*, K_b^*)$, where $\mathcal{B}_{\mathsf{a}}$ answers to $\mathcal{A}$'s decapsulation queries as Game 3 does, which is possible because $\mathcal{B}_{\mathsf{a}}$ has access to the oracle $\mathsf{PDecap}(\widehat{sk}_{c^*}, \cdot)$ and F is publicly computable. When $\mathcal{A}$ terminates, $\mathcal{B}_{\mathsf{a}}$ picks one of $\mathcal{A}$'s queries uniformly at random, outputs it, and terminates.

The above completes the description of $\mathcal{B}_{\mathsf{a}}$. Since $\mathcal{B}_{\mathsf{a}}$ perfectly simulates Game 3 for $\mathcal{A}$, the probability that $\mathcal{A}$ submits a decapsulation query c satisfying $\mathsf{F}(pk, c^*, c) = 0$ and $\mathsf{Decap}(sk, c) \neq \mathsf{PDecap}(\widehat{sk}_{c^*}, c)$ is exactly $\Pr[\mathsf{Diff}_3]$. Furthermore, since $\mathcal{B}_{\mathsf{a}}$ picks one of $\mathcal{A}$'s queries randomly, conditioned on the event that $\mathcal{A}$ submits a query satisfying the conditions of Diff_3, $\mathcal{B}_{\mathsf{a}}$ can output such a query with probability at least $1/Q$. Therefore, we have $\mathsf{Adv}^{\mathtt{PDSND}}_{\Gamma,\mathcal{B}_{\mathsf{a}}}(k) \geq (1/Q) \cdot \Pr[\mathsf{Diff}_3]$. This completes the proof of Claim 2. □

Claim 3 *There exists a PPTA $\mathcal{B}_{\mathsf{e}}$ such that* $\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma,\mathcal{B}_{\mathsf{e}}}(k) = 2 \cdot |\Pr[\mathsf{Succ}_3] - 1/2|$.

Proof of Claim 3. We show how to construct a PPTA adversary $\mathcal{B}_{\mathsf{e}}$ that attacks the $\mathtt{eCPA}$ security of the PKEM Γ with the claimed advantage. The description of $\mathcal{B}_{\mathsf{e}}$ is as follows:

$\mathcal{B}_{\mathsf{e}}(pk, \widehat{sk}_{c^*}, c^*, K_b^*)$**:** $\mathcal{B}_{\mathsf{e}}$ runs $b' \leftarrow \mathcal{A}(pk, c^*, K_b^*)$, where $\mathcal{B}_{\mathsf{e}}$ answers to $\mathcal{A}$'s decapsulation queries as Game 3 does, which is possible because $\mathcal{B}_{\mathsf{e}}$ possesses $\widehat{sk}_{c^*}$ and F is publicly computable. Finally, $\mathcal{B}_{\mathsf{e}}$ outputs b' and terminates.

The above completes the description of $\mathcal{B}_{\mathsf{e}}$. Note that $\mathcal{B}_{\mathsf{e}}$ perfectly simulates Game 3 for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is that of $\mathcal{B}_{\mathsf{e}}$'s. Since $\mathcal{B}_{\mathsf{e}}$ outputs $\mathcal{A}$'s output as it is, the probability that $b' = b$ occurs is exactly $\Pr[\mathsf{Succ}_3]$. Therefore, we have $\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma,\mathcal{B}_{\mathsf{e}}}(k) = 2 \cdot |\Pr[b' = b] - 1/2| = 2 \cdot |\Pr[\mathsf{Succ}_3] - 1/2|$. This completes the proof of Claim 3. □

Claims 1 to 3 and the inequality (4) guarantee that there exist PPTAs $\mathcal{B}_{\mathsf{d}}$, $\mathcal{B}_{\mathsf{a}}$, and $\mathcal{B}_{\mathsf{e}}$ satisfying the inequality (1), as required.

Tight Reduction with Strong Punctured Decapsulation Soundness. In the equation (1), 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. the condition $\mathsf{Decap}(sk, c') \neq \mathsf{PDecap}(\widehat{sk}_{c^*}, c')$. 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. A bit more formally, we can construct a reduction algorithm $\mathcal{B}'_{\mathsf{a}}$ such that $\mathsf{Adv}^{\mathtt{sPDSND}}_{\Gamma,\mathcal{B}'_{\mathsf{a}}}(k) \geq |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$ and use this as an alternative of Claim 2. Since the description of $\mathcal{B}'_{\mathsf{a}}$ is easily inferred from the explanation here, we do not write down it.

This completes the proof of Theorem 1. □

D.2 Proof of Lemma 1: Strong Decapsulation Soundness of $\Gamma_{\mathtt{DDN}}$

Let $\mathcal{A}$ be a PPTA $\mathtt{sDSND}$ adversary. Let (PK, SK, C^*, K^*) be a tuple that is input to $\mathcal{A}$ in the $\mathtt{sDSND}$ experiment, where $PK = ((pk_i^{(j)})_{i,j}, crs, \kappa)$, $SK = ((sk_i^{(j)})_{i,j}, PK)$, and $C^* = (vk^*, (c_i^*)_i, \pi^*, \sigma^*)$.

Let us call $\mathcal{A}$'s output $C' = (vk', (c_i')_i, \pi', \sigma')$ in the $\mathtt{sDSND}$ experiment *successful* if C' satisfies the conditions that violate the strong decapsulation soundness of $\Gamma_{\mathtt{DDN}}$, i.e. (a) $\mathsf{F}_{\mathtt{DDN}}(PK, C^*, C') = 1$, (b) $C' \neq C^*$, and (c) $\mathsf{Decap}_{\mathtt{DDN}}(SK, C') \neq \bot$. Note that the condition (a) implies $\mathsf{H}_\kappa(vk') = \mathsf{H}_\kappa(vk^*)$. Furthermore, the condition (c) implies $\mathsf{SVer}(vk', ((c_i')_i, \pi'), \sigma') = \top$. Therefore, taking into account the conditions (a) to (c) and additionally whether $vk' = vk^*$ holds or not, any successful ciphertext C' can be classified into the following two types:

Type 1: $vk' \neq vk^* \wedge \mathsf{H}_\kappa(vk') = \mathsf{H}_\kappa(vk^*)$
Type 2: $vk' = vk^* \wedge \mathsf{SVer}(vk', ((c_i')_i, \pi'), \sigma') = \top \wedge ((c_i')_i, \pi', \sigma') \neq ((c_i^*)_i, \pi^*, \sigma^*)$

It is easy to see that the probability that $\mathcal{A}$ succeeds in outputting a ciphertext of type 1 is negligible due to the security of the UOWHF $\mathcal{H}$, and the probability that $\mathcal{A}$ succeeds in outputting a ciphertext of type 2 is negligible due to the $\mathtt{SOT}$ security of the signature scheme Σ. We can easily describe reduction algorithms for both cases, and we omit them because they are straightforward. This completes the proof of Lemma 1. □

D.3 Proof of Lemma 2: Strong Punctured Decapsulation Soundness of $\varGamma_{\mathtt{DDN}}$

Let $\mathcal{A}$ be a PPTA $\mathtt{sPDSND}$ adversary. Let (PK, SK, C^*, K^*) be the tuple input to $\mathcal{A}$ in the $\mathtt{sPDSND}$ experiment, where $PK = ((pk_i^{(j)})_{i,j}, crs, \kappa)$, $SK = ((sk_i^{(j)})_{i,j}, PK)$, and $C^* = (vk^*, (c_i^*)_i, \pi^*, \sigma^*)$. Furthermore, let $\widehat{SK}_{C^*} = (h^*, (sk_i^{(1-h_i^*)})_i, PK) = \mathsf{Punc}_{\mathtt{DDN}}(SK, C^*)$, where $h^* = (h_1^* \| \dots \| h_k^*) = \mathsf{H}_\kappa(vk^*)$.

Let us call $\mathcal{A}$'s output $C' = (vk', (c'_i)_i, \pi', \sigma')$ in the $\mathtt{sPDSND}$ experiment *successful* if C' satisfies the conditions that violate the strong punctured decapsulation soundness of $\varGamma_{\mathtt{DDN}}$, namely (a) $\mathsf{F}_{\mathtt{DDN}}(PK, C^*, C') = 0$ (which implies $h^* \neq h' = (h'_1 \| \dots \| h'_k) = \mathsf{H}_\kappa(vk')$) and (b) $\mathsf{Decap}_{\mathtt{DDN}}(SK, C') \neq \mathsf{PDecap}_{\mathtt{DDN}}(\widehat{SK}_{C^*}, C')$.

We first confirm that if $\mathcal{A}$'s output is successful, then it must be the case that $\mathsf{SVer}(vk', ((c'_i)_i, \pi'), \sigma') = \top$, $\mathsf{PVer}(crs, x', \pi') = \top$, and $x' \notin L_k$ where $x' = ((pk_i^{(h'_i)})_i, (c'_i)_i)$. To see this, consider the opposite: $\mathsf{SVer}(vk', ((c'_i)_i, \pi'), \sigma') = \bot$, $\mathsf{PVer}(crs, x', \pi') = \bot$, or $x' \in L_k$. Either of the first two conditions makes both $\mathsf{Decap}_{\mathtt{DDN}}$ and $\mathsf{PDecap}_{\mathtt{DDN}}$ output $\bot$, which contradicts the condition $\mathsf{Decap}_{\mathtt{DDN}}(SK, C') \neq \mathsf{PDecap}_{\mathtt{DDN}}(\widehat{SK}_{C^*}, C')$. Moreover, if $\mathsf{SVer}(vk', ((c'_i)_i, \pi'), \sigma') = \top$ and $\mathsf{PVer}(crs, x', \pi') = \top$ hold, then we have that $\mathsf{Decap}_{\mathtt{DDN}}(SK, C') = \mathsf{Dec}(sk_1^{(h'_1)}, c'_1)$ and $\mathsf{PDecap}_{\mathtt{DDN}}(\widehat{SK}_{C^*}, C') = \mathsf{Dec}(sk_\ell^{(1-h_\ell^*)}, c'_\ell) = \mathsf{Dec}(sk_\ell^{(h'_\ell)}, c'_\ell)$, where the latter equality is because $h'_\ell = 1 - h_\ell^*$. However, if $x' \in L_k$, then every component c'_i is an honestly generated ciphertext of a same plaintext $K' \in \{0,1\}^k$, and thus $\mathsf{Dec}(sk_1^{(h'_1)}, c'_1) = \mathsf{Dec}(sk_\ell^{(h'_\ell)}, c'_\ell)$ holds, but this again contradicts the condition of $\mathsf{Decap}_{\mathtt{DDN}}(SK, C') \neq \mathsf{PDecap}_{\mathtt{DDN}}(\widehat{SK}_{C^*}, C')$.

Now, we have seen that if $\mathcal{A}$'s output is successful, then it holds that $\mathsf{PVer}(crs, x', \pi') = \top$ and $x' \notin L_k$. However, this is exactly the conditions that violate the adaptive soundness of the non-interactive argument system $\mathcal{P}$. More specifically, using an adversary $\mathcal{A}$ that outputs a successful ciphertext with non-negligible advantage in the $\mathtt{sPDSND}$ experiment, we can construct another adversary $\mathcal{B}_{\mathsf{s}}$ that has non-negligible advantage in breaking the adaptive soundness of the non-interactive argument system $\mathcal{P}$. Since the reduction algorithm is straightforward (which runs $\mathcal{A}$ using a crs that it receives, and checks whether $\mathcal{A}$ outputs a successful ciphertext), we omit the details. This completes the proof of Lemma 2. □

D.4 Proof of Lemma 3: $\mathtt{eCPA}$ Security of $\varGamma_{\mathtt{DDN}}$

Let $\mathcal{S} = (\mathsf{SimCRS}, \mathsf{SimPrv})$ be the simulation algorithms for the non-interactive argument system $\mathcal{P}$ guaranteed by its $\mathtt{ZK}$ security.

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

Game 1: This is the $\mathtt{eCPA}$ experiment itself.

Game 2: Same as Game 1, except that we use the simulation algorithms SimCRS and SimPrv to generate crs and π^*, respectively. More precisely, in this game, the steps "$crs \leftarrow \mathsf{CRSG}(1^k)$" and "$\pi^* \leftarrow \mathsf{Prove}(crs, x^*, w^*)$" in Game 1 are replaced with the steps "$(crs, td) \leftarrow \mathsf{SimCRS}(1^k)$" and "$\pi^* \leftarrow \mathsf{SimPrv}(td, x^*)$," respectively.

Game 3: Same as Game 2, except that the information of K_1^* is erased from the ciphertexts $(c_i^*)_i$. More precisely, in this game, each step "$c_i^* \leftarrow \mathsf{Enc}(pk_i^{(h_i^*)}, K_1^*; r_i^*)$" in Game 2 is replaced with the step "$c_i^* \leftarrow \mathsf{Enc}(pk_i^{(h_i^*)}, 0^k; r_i^*)$."

For $i \in [3]$, let Succ_i be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game i. By definition, $\mathcal{A}$'s $\mathtt{eCPA}$ advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma_{\mathtt{DDN}},\mathcal{A}}(k) = 2 \cdot |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \leq 2 \cdot \sum_{i \in [2]} |\Pr[\mathsf{Succ}_i] - \Pr[\mathsf{Succ}_{i+1}]| + 2 \cdot |\Pr[\mathsf{Succ}_3] - \frac{1}{2}|. \quad (5)$$

In the following we upperbound each term in the right hand side of the above inequality.

Claim 4 *There exists a PPTA* $\mathcal{B}_{\mathtt{z}}$ *such that* $\mathsf{Adv}^{\mathtt{ZK}}_{\mathcal{P},\mathcal{S},\mathcal{B}_{\mathtt{z}}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$.

Proof of Claim 4. We show how to construct a PPTA adversary $\mathcal{B}_{\mathtt{z}}$ that attacks the $\mathtt{ZK}$ security of the non-interactive argument system $\mathcal{P}$ with the claimed advantage. The description of $\mathcal{B}_{\mathtt{z}} = (\mathcal{B}_{\mathtt{z}1}, \mathcal{B}_{\mathtt{z}2})$ as follows:

$\mathcal{B}_{\mathtt{z}1}(1^k)$**:** For every $(i,j) \in [k] \times \{0,1\}$, $\mathcal{B}_{\mathtt{z}1}$ runs $(pk_i^{(j)}, sk_i^{(j)}) \leftarrow \mathsf{PKG}(1^k)$. Next, $\mathcal{B}_{\mathtt{z}1}$ picks $K_1^* \in \{0,1\}^k$ and $(r_i^*)_i \in (\mathcal{R}_k)^k$ uniformly at random. $\mathcal{B}_{\mathtt{z}1}$ then executes $(vk^*, sigk^*) \leftarrow \mathsf{SKG}(1^k)$, $\kappa \leftarrow \mathsf{HKG}(1^k)$, and $h^* = (h_1^* \| \dots \| h_k^*) \leftarrow \mathsf{H}_\kappa(vk^*)$. Then, for every $i \in [k]$, $\mathcal{B}_{\mathtt{z}1}$ runs $c_i^* \leftarrow \mathsf{Enc}(pk_i^{(h_i^*)}, K_1^*; r_i^*)$. Finally, $\mathcal{B}_{\mathtt{z}1}$ sets $x^* \leftarrow ((pk_i^{(h_i^*)})_i, (c_i^*)_i)$, $w^* \leftarrow ((r_i^*)_i, K_1^*)$, and $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathtt{z}1}$'s entire view), and terminates with output $(x^*, w^*, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathtt{z}2}(\mathsf{st}_{\mathcal{B}}, crs, \pi^*)$**:** $\mathcal{B}_{\mathtt{z}2}$ first runs $\sigma^* \leftarrow \mathsf{Sign}(sigk^*, ((c_i^*)_i, \pi^*))$, and sets $PK \leftarrow ((pk_i^{(j)})_{i,j}, crs, \kappa)$, $C^* \leftarrow (vk^*, (c_i^*)_i, \pi^*, \sigma^*)$, and $\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$. Then $\mathcal{B}_{\mathtt{z}2}$ 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 $\mathcal{B}_{\mathtt{z}}$. Note that $\mathcal{B}_{\mathtt{z}2}$ outputs 1 only when $b' = b$ occurs. $\mathcal{B}_{\mathtt{z}}$'s $\mathtt{ZK}$ advantage can be estimated as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{ZK}}_{\mathcal{P},\mathcal{S},\mathcal{B}_{\mathtt{z}}}(k) &= |\Pr[\mathsf{Expt}^{\mathtt{ZK}\text{-}\mathtt{Real}}_{\mathcal{P},\mathcal{B}_{\mathtt{z}}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{ZK}\text{-}\mathtt{Sim}}_{\mathcal{P},\mathcal{S},\mathcal{B}_{\mathtt{z}}}(k) = 1]| \\ &= |\Pr[\mathsf{Expt}^{\mathtt{ZK}\text{-}\mathtt{Real}}_{\mathcal{P},\mathcal{B}_{\mathtt{z}}}(k) : b' = b] - \Pr[\mathsf{Expt}^{\mathtt{ZK}\text{-}\mathtt{Sim}}_{\mathcal{P},\mathcal{S},\mathcal{B}_{\mathtt{z}}}(k) : b' = b]|.\end{aligned}$$

Consider the case when $\mathcal{B}_{\mathtt{z}}$ runs in $\mathsf{Expt}^{\mathtt{ZK}\text{-}\mathtt{Real}}_{\mathcal{P},\mathcal{B}_{\mathtt{z}}}(k)$. It is easy to see that in this case, $\mathcal{B}_{\mathtt{z}}$ perfectly simulates Game 1 for $\mathcal{A}$. In particular, the common reference string crs and the proof π^* are generated from CRSG and Prove, respectively, which is exacly how they are generated 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., $\Pr[\mathsf{Expt}^{\mathtt{ZK}\text{-}\mathtt{Real}}_{\mathcal{P},\mathcal{B}_{\mathtt{z}}}(k) : b' = b] = \Pr[\mathsf{Succ}_1]$.

When $\mathcal{B}_{\mathtt{z}}$ runs in $\mathsf{Expt}^{\mathtt{ZK}\text{-}\mathtt{Sim}}_{\mathcal{P},\mathcal{S},\mathcal{B}_{\mathtt{z}}}(k)$, on the other hand, crs and π^* are genearted from SimCRS and SimPrv, respectively, as done in Game 2. Since this is the only change from the above, with a similar argument to the above, we have $\Pr[\mathsf{Expt}^{\mathtt{ZK}\text{-}\mathtt{Sim}}_{\mathcal{P},\mathcal{S},\mathcal{B}_{\mathtt{z}}}(k) : b' = b] = \Pr[\mathsf{Succ}_2]$.

In summary, we have $\mathsf{Adv}^{\mathtt{ZK}}_{\mathcal{P},\mathcal{S},\mathcal{B}_{\mathtt{z}}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$. This completes the proof of Claim 4. □

Claim 5 *There exists a PPTA* $\mathcal{B}_{\mathtt{p}}$ *such that* $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi^k,\mathcal{B}_{\mathtt{p}}}(k) = |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$.

Proof of Claim 5. We show how to construct a PPTA adversary $\mathcal{B}_{\mathsf{p}}$ that attacks the $\mathtt{CPA}$ security of the k-repetition construction Π^k with the claimed advantage. The description of $\mathcal{B}_{\mathsf{p}} = (\mathcal{B}_{\mathsf{p}1}, \mathcal{B}_{\mathsf{p}2})$ is as follows:

$\mathcal{B}_{\mathsf{p}1}(PK' = (pk'_i)_i)$**:** $\mathcal{B}_{\mathsf{p}1}$ picks $K_1^* \in \{0,1\}^k$ uniformly at random, sets $(M_0, M_1) \leftarrow (0^k, K_1^*)$ and $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathsf{p}1}$'s entire view), and terminates with output $(M_0, M_1, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathsf{p}2}(\mathsf{st}_{\mathcal{B}}, C'^* = (c_i^*)_i)$**:** $\mathcal{B}_{\mathsf{p}2}$ runs $(vk^*, sigk^*) \leftarrow \mathsf{SKG}(1^k)$, $\kappa \leftarrow \mathsf{HKG}(1^k)$, and $h^* = (h_1^* \| \dots \| h_k^*) \leftarrow \mathsf{H}_\kappa(vk^*)$. Next, for every $i \in [k]$, $\mathcal{B}_{\mathsf{p}2}$ sets $pk_i^{(h_i^*)} \leftarrow pk'_i$ and runs $(pk_i^{(1-h_i^*)}, sk_i^{(1-h_i^*)}) \leftarrow \mathsf{PKG}(1^k)$. $\mathcal{B}_{\mathsf{p}2}$ then sets $x^* \leftarrow ((pk_i^{(h_i^*)})_i, (c_i^*)_i)$, runs $(crs, td) \leftarrow \mathsf{SimCRS}(1^k)$, $\pi^* \leftarrow \mathsf{SimPrv}(td, x^*)$, and $\sigma^* \leftarrow \mathsf{Sign}(sigk^*, ((c_i^*)_i, \pi^*))$. Then, $\mathcal{B}_{\mathsf{p}2}$ sets $PK \leftarrow ((pk_i^{(j)})_{i,j}, crs, \kappa)$, $C^* \leftarrow (vk^*, (c_i^*)_i, \pi^*, \sigma^*)$, and $\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$. Finally, $\mathcal{B}_{\mathsf{p}2}$ 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 $\mathcal{B}_{\mathsf{p}}$. Let $\gamma \in \{0,1\}$ be $\mathcal{B}_{\mathsf{p}}$'s challenge bit. $\mathcal{B}_{\mathsf{p}}$'s $\mathtt{CPA}$ advantage is estimated as follows:

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

Consider the case when $\gamma = 1$, i.e. each c_i^* is an encryption of $M_1 = K_1^*$. It is easy to see that in this case, the challenge ciphertext C^* is generated in exactly the same way as that in Game 2, and $\mathcal{B}_{\mathsf{p}}$ 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, each c_i^* is an encryption of 0^k, which is exactly how it is generated in Game 3. Since this is the only change from the above, with a similar discussion, we have $\Pr[b' = b|\gamma = 0] = \Pr[\mathsf{Succ}_3]$.

In summary, we have $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi^k, \mathcal{B}_{\mathsf{p}}}(k) = |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$. This completes the proof of Claim 5. □

Claim 6 $\Pr[\mathsf{Succ}_3] = 1/2$.

Proof of Claim 6. In Game 3, C^* is independent of K_1^*. In particular, each of the components c_i^* in C^* is an encryption of 0^k. Since both K_1^* and K_0^* are chosen uniformly at random, $\mathcal{A}$'s view is identically distributed regardless of the challenge bit $b \in \{0,1\}$. This must mean that the probability that $\mathcal{A}$ succeeds in guessing the challenge bit b is exactly $1/2$. This completes the proof of Claim 6. □

Claims 4 to 6 and the inequality (5) guarantee that there exist PPTAs $\mathcal{B}_{\mathsf{z}}$ and $\mathcal{B}_{\mathsf{p}}$ such that

$$\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma_{\mathtt{DDN}}, \mathcal{A}}(k) \le 2 \cdot \Big(\mathsf{Adv}^{\mathtt{ZK}}_{\mathcal{P}, \mathcal{S}, \mathcal{B}_{\mathsf{z}}}(k) + \mathsf{Adv}^{\mathtt{CPA}}_{\Pi^k, \mathcal{B}_{\mathsf{p}}}(k)\Big),$$

which, due to our assumptions on the building blocks and Lemma 9 (see Appendix B.1), implies that $\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma_{\mathtt{DDN}}, \mathcal{A}}(k)$ is negligible. Recall that the choice of the PPTA $\mathtt{eCPA}$ adversary $\mathcal{A}$ was arbitrarily, and thus for any PPTA $\mathtt{eCPA}$ adversary $\mathcal{A}$ we can show a negligible upperbound for $\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma_{\mathtt{DDN}}, \mathcal{A}}(k)$ as above. Hence, the PKEM $\Gamma_{\mathtt{DDN}}$ is $\mathtt{eCPA}$ secure. This completes the proof of Lemma 3. □

D.5 Proof of Lemma 4: Strong Decapsulation Soundness of $\widehat{\Gamma}$

Let $\mathcal{A}$ be a PPTA $\mathtt{sDSND}$ adversary. Let (PK, SK, C^*, K^*) be a tuple that is input to $\mathcal{A}$ in the $\mathtt{sDSND}$ 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 $\mathtt{sDSND}$ 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 $\widehat{\mathsf{Decap}}(SK, C') \neq \bot$. 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 $\mathit{\Pi}$). 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 $C' \neq C^*$.

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 now show that 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.

The description of $\mathcal{B}_{\mathtt{h}} = (\mathcal{B}_{\mathtt{h1}}, \mathcal{B}_{\mathtt{h2}})$ is as follows.

$\mathcal{B}_{\mathtt{h1}}(1^k):$ $\mathcal{B}_{\mathtt{h1}}$ picks $\alpha^* \in \mathcal{K}_k$ and $\beta^* = ((r_i^*)_{i \in [k+1]}, K^*) \in (\mathcal{R}_k)^{k+1} \times \{0,1\}^k$ uniformly at random, and runs $(pk_{k+1}, sk_{k+1}) \leftarrow \mathsf{PKG}(1^k)$, $c_{k+1}^* \leftarrow \mathsf{Enc}(pk_{k+1}, \alpha^*; r_{k+1}^*)$, and $\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, \beta^*)$. Then $\mathcal{B}_{\mathtt{h1}}$ sets $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathtt{h1}}$'s entire view), and terminates with output $((c_{k+1}^* \| \widetilde{c}^*), \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathtt{h2}}(\mathsf{st}, \kappa):$ $\mathcal{B}_{\mathtt{h2}}$ runs $(pk_i^{(j)}, sk_i^{(j)}) \leftarrow \mathsf{PKG}(1^k)$ for every $(i,j) \in [k] \times \{0,1\}$, and then sets $PK \leftarrow ((pk_i^{(j)})_{i,j}, pk_{k+1}, \kappa)$ and $SK \leftarrow ((sk_i^{(j)})_{i,j}, PK)$. Then $\mathcal{B}_{\mathtt{h2}}$ executes $h^* = (h_1^* \| \dots \| h_k^*) \leftarrow \mathsf{H}_\kappa(c_{k+1}^* \| \widetilde{c}^*)$ and $c_i^* \leftarrow \mathsf{Enc}(pk_i^{(h_i^*)}, \alpha^*; r_i^*)$ for every $i \in [k]$, sets $C^* \leftarrow (h^*, (c_i^*)_i, \widetilde{c}^*)$, and runs $C' = (h', (c_i')_i, \widetilde{c}') \leftarrow \mathcal{A}(PK, SK, C^*, K^*)$. Then $\mathcal{B}_{\mathtt{h2}}$ executes $\widehat{\mathsf{Decap}}(SK, C')$ until it calculates c_{k+1}'. If $\widehat{\mathsf{Decap}}(SK, C')$ returns $\bot$ before it calculates c_{k+1}', then $\mathcal{B}_{\mathtt{h2}}$ simply gives up and aborts. Otherwise, $\mathcal{B}_{\mathtt{h2}}$ terminates with output $(c_{k+1}' \| \widetilde{c}')$.

The above completes the description of $\mathcal{B}_{\mathtt{h}}$. It is easy to see that $\mathcal{B}_{\mathtt{h}}$ simulates the $\mathtt{sDSND}$ experiment perfectly for $\mathcal{A}$, and thus $\mathcal{B}_{\mathtt{h}}'$'s advantage in the $\mathtt{UOW}$ expriment for $\mathcal{H}$ is exactly the probabality that $\mathcal{A}$ outputs a successful ciphertext, as required. This completes the proof of Lemma 4. □

D.6 Proof of Lemma 5: Strong Punctured Decapsulation Soundness of $\widehat{\Gamma}$

Let (PK, SK) be the 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 a 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 $\ell = \min\{i \in [k] : h_i^* \neq h_i\}$, 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 $\mathit{\Pi}$ 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. □

D.7 Proof of Lemma 6: eCPA Security of $\widehat{\mathit{\Gamma}}$

Let $\mathcal{A}$ be any PPTA adversary that attacks the eCPA security of $\widehat{\mathit{\Gamma}}$. 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):

```
Game 1:
  $\alpha^* \leftarrow \mathcal{K}_k$;
  For $i \in [k+1]$ :
    $\underline{(pk'_i, sk'_i) \leftarrow \mathsf{PKG}(1^k);}$
    $\underline{r_i^* \leftarrow \mathcal{R}_k;}$
    $\underline{c_i^* \leftarrow \mathsf{Enc}(pk'_i, \alpha^*; r_i^*);}$
  End For
  $K_1^* \leftarrow \{0,1\}^k$;
  (Continue to the center column ↗)

$\beta^* \leftarrow ((r_i^*)_{i \in [k+1]}, K_1^*)$;
$\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, \beta^*)$;
$\kappa \leftarrow \mathsf{HKG}(1^k)$;
$h^* = (h_1^* \| \dots \| h_k^*) \leftarrow \mathsf{H}_\kappa(c_{k+1}^* \| \widetilde{c}^*)$;
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
(Continue to the right column ↗)

$PK \leftarrow ((pk_i^{(j)})_{i,j}, pk'_{k+1}, \kappa)$;
$C^* \leftarrow (h^*, (c_i^*)_i, \widetilde{c}^*)$;
$\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$;
$K_0^* \leftarrow \{0,1\}^k$;
$b \leftarrow \{0,1\}$;
$b' \leftarrow \mathcal{A}(PK, \widehat{SK}_{C^*}, C^*, K_b^*)$
```

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 $\mathit{\Pi}$. More precisely, in this game, the step with the <u>underline</u> 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). By definition, $\mathcal{A}$'s eCPA advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{eCPA}}_{\widehat{\mathit{\Gamma}},\mathcal{A}}(k) = 2 \cdot |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \leq 2 \cdot \sum_{i \in [2]} |\Pr[\mathsf{Succ}_i] - \Pr[\mathsf{Succ}_{i+1}]| + 2 \cdot |\Pr[\mathsf{Succ}_3] - \frac{1}{2}|. \tag{6}$$

In the following we upperbound each term in the right hand side of the above inequality.

Claim 7 *There exists a PPTA $\mathcal{B}_\mathsf{p}$ such that* $\mathsf{Adv}^{\mathtt{SNC}}_{\mathit{\Pi}^{k+1},\mathcal{B}_\mathsf{p}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$.

Proof of Claim 7. We show how to construct a PPTA adversary $\mathcal{B}_{\mathsf{p}}$ that attacks the $\mathtt{SNC}$ security of the $(k+1)$-repetition construction $\varPi^{k+1}$ of the SNCE scheme with the claimed advantage. The description of $\mathcal{B}_{\mathsf{p}} = (\mathcal{B}_{\mathsf{p1}}, \mathcal{B}_{\mathsf{p2}})$ as follows:

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

$\mathcal{B}_{\mathsf{p2}}(\mathsf{st}_{\mathcal{B}}, PK' = (pk'_i)_{i \in [k+1]}, C'^* = (c^*_i)_{i \in [k+1]}, R'^* = (r^*_i)_{i \in [k+1]})$**:** $\mathcal{B}_{\mathsf{p2}}$ 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 $h^* = (h^*_1 \| \dots \| h^*_k) \leftarrow \mathsf{H}_\kappa(c^*_{k+1} \| \widetilde{c}^*)$. For each $i \in [k]$, $\mathcal{B}_{\mathsf{p2}}$ 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 $\mathcal{B}_{\mathsf{p2}}$ 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 $\mathcal{B}_{\mathsf{p2}}$ 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 $(b' \overset{?}{=} b)$.

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

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

Consider the case when $\mathcal{B}_{\mathsf{p}}$ runs in $\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\varPi^{k+1}, \mathcal{B}_{\mathsf{p}}}(k)$. It is easy to see that in this case, $\mathcal{B}_{\mathsf{p}}$ 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., $\Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\varPi^{k+1}, \mathcal{B}_{\mathsf{p}}}(k) : b' = b] = \Pr[\mathsf{Succ}_1]$.

When $\mathcal{B}_{\mathsf{p}}$ runs in $\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\varPi^{k+1}, \mathcal{B}_{\mathsf{p}}}(k)$, 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 $\varPi$, 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., $\Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\varPi^{k+1}, \mathcal{B}_{\mathsf{p}}}(k) : b' = b] = \Pr[\mathsf{Succ}_2]$.

In summary, we have $\mathsf{Adv}^{\mathtt{SNC}}_{\varPi^{k+1}, \mathcal{B}_{\mathsf{p}}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$. This completes the proof of Claim 7. □

Claim 8 *There exists a PPTA $\mathcal{B}_{\mathsf{e}}$ such that* $\mathsf{Adv}^{\mathtt{OTKDM}}_{E, \mathcal{F}, \mathcal{B}_{\mathsf{e}}}(k) = |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$.

Proof of Claim 8. We show how to construct a PPTA adversary $\mathcal{B}_{\mathsf{e}}$ that attacks the $\mathcal{F}$-$\mathtt{OTKDM}$ security of the underlying SKE scheme E with the claimed advantage. The description of $\mathcal{B}_{\mathsf{e}} = (\mathcal{B}_{\mathsf{e1}}, \mathcal{B}_{\mathsf{e2}})$ is as follows:

$\mathcal{B}_{\mathsf{e1}}(1^k)$**:** For every $i \in [k+1]$, $\mathcal{B}_{\mathsf{e}}$ runs $(pk'_i, c^*_i, \omega^*_i) \leftarrow \mathsf{Fake}(1^k)$. Then, $\mathcal{B}_{\mathsf{e1}}$ picks $K^*_1 \in \{0,1\}^k$ uniformly at random. Next, $\mathcal{B}_{\mathsf{e1}}$ specifies the function $f : \mathcal{K}_k \rightarrow \mathcal{M}_k$ 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, $\mathcal{B}_{\mathsf{e1}}$ sets $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathsf{e1}}$'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 $h^* = (h_1^* \| \dots \| h_k^*) \leftarrow \mathsf{H}_\kappa(c_{k+1}^* \| \widetilde{c}^*)$. 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 8. □

Claim 9 $\Pr[\mathsf{Succ}_3] = 1/2$.

Proof of Claim 9. In Game 3, the challenge ciphertext C^* is made independent of the "real" session key K_1^*, and both K_1^* and K_0^* are distributed identically (uniformly at random in $\{0,1\}^k$). Hence, the challenge bit b is information-theoretically hidden from the view of $\mathcal{A}$. This means that the probability that $\mathcal{A}$ succeeds in guessing the challenge bit is exactly $1/2$ (even if $\mathcal{A}$ is computationally unbounded). This completes the proof of Claim 9. □

Claims 7 to 9 and the inequality (6) guarantee that there exist PPTAs $\mathcal{B}_{\mathsf{p}}$ and $\mathcal{B}_{\mathsf{e}}$ such that

$$\mathsf{Adv}^{\mathtt{eCPA}}_{\widehat{\Gamma},\mathcal{A}}(k) \leq 2 \cdot \left(\cdot \mathsf{Adv}^{\mathtt{SNC}}_{\Pi^{k+1},\mathcal{B}_{\mathsf{p}}}(k) + \mathsf{Adv}^{\mathtt{OTKDM}}_{E,\mathcal{F},\mathcal{B}_{\mathsf{e}}}(k) \right),$$

which, due to our assumptions on the building blocks and Lemma 10, implies that $\mathsf{Adv}^{\mathtt{eCPA}}_{\widehat{\Gamma},\mathcal{A}}(k)$ is negligible. Recall that the choice of the PPTA $\mathtt{eCPA}$ adversary $\mathcal{A}$ was arbitrarily, and thus for any PPTA $\mathtt{eCPA}$ adversary $\mathcal{A}$ we can show a negligible upperbound for $\mathsf{Adv}^{\mathtt{eCPA}}_{\widehat{\Gamma},\mathcal{A}}(k)$ as above. Hence, the PKEM $\widehat{\Gamma}$ is $\mathtt{eCPA}$ secure. This completes the proof of Lemma 6. □

D.8 Proof of Lemma 7: eCPA Security of $\mathit{\Gamma}_{\mathtt{DDN}}$ from Different Assumptions

Let $\mathcal{A}$ be any PPTA adversary that attacks the eCPA security of $\mathit{\Gamma}_{\mathtt{DDN}}$. 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):

```
Game 1:
  $K_1^* \leftarrow \{0,1\}^k$;
  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
  $x^* \leftarrow ((pk_i')_i, (c_i^*)_i)$;
  $w^* \leftarrow ((r_i^*)_i, K_1^*)$;
  (Continue to the center column $\nearrow$)
  $crs \leftarrow \mathsf{CRSG}(1^k)$;
  $\pi^* \leftarrow \mathsf{Prove}(crs, x^*, w^*)$;
  $(vk^*, sigk^*) \leftarrow \mathsf{SKG}(1^k)$;
  $\sigma^* \leftarrow \mathsf{Sign}(sigk^*, ((c_i^*)_i, \pi^*))$;
  $\kappa \leftarrow \mathsf{HKG}(1^k)$;
  $h^* = (h_1^* \| \dots \| h_k^*) \leftarrow \mathsf{H}_\kappa(vk^*)$;
  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
  (Continue to the right column $\nearrow$)
  $PK \leftarrow ((pk_i^{(j)})_{i,j}, crs, \kappa)$;
  $C^* \leftarrow (vk^*, (c_i^*)_i, \pi^*, \sigma^*)$;
  $\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$;
  $K_0^* \leftarrow \{0,1\}^k$;
  $b \leftarrow \{0,1\}$;
  $b' \leftarrow \mathcal{A}(PK, \widehat{SK}_{C^*}, C^*, K_b^*)$
```

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 $\mathsf{Explain}$ of the SNCE scheme $\mathit{\Pi}$. 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). By definition, $\mathcal{A}$'s eCPA advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{eCPA}}_{\mathit{\Gamma}_{\mathtt{DDN}},\mathcal{A}}(k) = 2 \cdot |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \le 2 \cdot \sum_{i \in [2]} |\Pr[\mathsf{Succ}_i] - \Pr[\mathsf{Succ}_{i+1}]| + 2 \cdot |\Pr[\mathsf{Succ}_3] - \frac{1}{2}|. \quad (7)$$

In the following we upperbound each term in the right hand side of the above inequality.

Claim 10 *There exists a PPTA $\mathcal{B}_{\mathsf{p}}$ such that* $\mathsf{Adv}^{\mathtt{SNC}}_{\mathit{\Pi}^k,\mathcal{B}_{\mathsf{p}}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$.

Proof of Claim 10. We show how to construct a PPTA adversary $\mathcal{B}_{\mathsf{p}}$ that attacks the SNC security of the k-repetition construction $\mathit{\Pi}^k$ of the SNCE scheme with the claimed advantage. The description of $\mathcal{B}_{\mathsf{p}} = (\mathcal{B}_{\mathsf{p}1}, \mathcal{B}_{\mathsf{p}2})$ as follows:

$\mathcal{B}_{\mathsf{p}1}(1^k)$**:** $\mathcal{B}_{\mathsf{p}1}$ picks $K_1^* \in \{0,1\}^k$ uniformly at random. Then $\mathcal{B}_{\mathsf{p}1}$ sets $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathsf{p}1}$'s entire view), and terminates with output $(K_1^*, \mathsf{st}_{\mathcal{B}})$ (where K_1^* is regarded as $\mathcal{B}_{\mathsf{p}}$'s challenge message).

$\mathcal{B}_{\mathsf{p}2}(\mathsf{st}_{\mathcal{B}}, PK' = (pk_i')_{i \in [k]}, C'^* = (c_i^*)_{i \in [k]}, R'^* = (r_i^*)_{i \in [k]})$**:** $\mathcal{B}_{\mathsf{p}2}$ sets $x^* \leftarrow ((pk_i')_i, (c_i^*)_i)$ and $w^* \leftarrow ((r_i^*)_i, K_1^*)$, and runs $crs \leftarrow \mathsf{CRSG}(1^k)$ and $\pi^* \leftarrow \mathsf{Prove}(crs, x^*, w^*)$. Next, $\mathcal{B}_{\mathsf{p}2}$ runs $(vk^*, sigk^*) \leftarrow \mathsf{SKG}(1^k)$, $\sigma^* \leftarrow \mathsf{Sign}(sigk^*, ((c_i^*)_i, \pi^*))$, $\kappa \leftarrow \mathsf{HKG}(1^k)$, and $h^* = (h_1^* \| \dots \| h_k^*) \leftarrow \mathsf{H}_\kappa(vk^*)$. For each $i \in [k]$, $\mathcal{B}_{\mathsf{p}2}$ sets $pk_i^{(h_i^*)} \leftarrow pk_i'$ and generates $(pk_i^{(1-h_i^*)}, sk_i^{(1-h_i^*)}) \leftarrow \mathsf{PKG}(1^k)$. Then, $\mathcal{B}_{\mathsf{p}2}$ sets $PK \leftarrow ((pk_i^{(j)})_{i,j}, crs, \kappa)$, $C^* \leftarrow (vk^*, (c_i^*)_i, \pi^*, \sigma^*)$, and $\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$. Finally, $\mathcal{B}_{\mathsf{p}2}$ 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 $\mathcal{B}_\mathsf{p}$. Note that $\mathcal{B}_{\mathsf{p}2}$ outputs 1 only when $b' = b$ occurs. $\mathcal{B}_\mathsf{p}$'s $\mathtt{SNC}$ advantage can be estimated as follows:

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

Consider the case when $\mathcal{B}_\mathsf{p}$ runs in $\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\Pi^k,\mathcal{B}_\mathsf{p}}(k)$. It is easy to see that in this case, $\mathcal{B}_\mathsf{p}$ perfectly simulates Game 1 for $\mathcal{A}$. In particular, every $pk_i^{(j)}$ in PK is generated honestly by running $\mathsf{PKG}(1^k)$, and every c_i^* in C^* is generated as $c_i^* \leftarrow \mathsf{Enc}(pk_i^{(h_i^*)}, K_1^*; r_i^*)$ where $K_1^* \in \{0,1\}^k$ 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., $\Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Real}}_{\Pi^k,\mathcal{B}_\mathsf{p}}(k) : b' = b] = \Pr[\mathsf{Succ}_1]$.

When $\mathcal{B}_\mathsf{p}$ runs in $\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\Pi^k,\mathcal{B}_\mathsf{p}}(k)$, 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 $K_1^* \in \{0,1\}^k$ 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., $\Pr[\mathsf{Expt}^{\mathtt{SNC\text{-}Sim}}_{\Pi^k,\mathcal{B}_\mathsf{p}}(k) : b' = b] = \Pr[\mathsf{Succ}_2]$.

In summary, we have $\mathsf{Adv}^{\mathtt{SNC}}_{\Pi^k,\mathcal{B}_\mathsf{p}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$. This completes the proof of Claim 10. □

Claim 11 *There exists a PPTA* $\mathcal{B}_\mathtt{w}$ *such that* $\mathsf{Adv}^{\mathtt{WI}}_{\mathcal{P},\mathcal{B}_\mathtt{w}}(k) = |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$.

Proof of Claim 11. We show how to construct a PPTA adversary $\mathcal{B}_\mathtt{w}$ that attacks the $\mathtt{WI}$ security of the underlying non-interactive argument system $\mathcal{P}$ with the claimed advantage. The description of $\mathcal{B}_\mathtt{w} = (\mathcal{B}_{\mathtt{w}1}, \mathcal{B}_{\mathtt{w}2})$ is as follows:

$\mathcal{B}_{\mathtt{w}1}(1^k)$**:** For every $i \in [k]$, $\mathcal{B}_\mathtt{w}$ runs $(pk_i', c_i^*, \omega_i^*) \leftarrow \mathsf{Fake}(1^k)$. Next, $\mathcal{B}_{\mathtt{w}1}$ picks $K_1^* \in \{0,1\}^k$ uniformly at random, and for every $i \in [k]$, $\mathcal{B}_{\mathtt{w}1}$ computes $r_i^* \leftarrow \mathsf{Explain}(\omega_i^*, K_1^*)$ and $r_i' \leftarrow \mathsf{Explain}(\omega_i^*, 0^k)$. Then, $\mathcal{B}_{\mathtt{w}1}$ sets $x^* \leftarrow ((pk_i')_i, (c_i^*)_i)$, $w_1 \leftarrow ((r_i^*)_i, K_1^*)$, and $w_0 \leftarrow ((r_i')_i, 0^k)$. Note that both w_0 and w_1 are witnesses to the fact that $x^* \in L_k$. Finally, $\mathcal{B}_{\mathtt{w}1}$ sets $\mathsf{st}_\mathcal{B} \leftarrow$ ($\mathcal{B}_{\mathtt{w}1}$'s entire view), and terminates with output $(x^*, w_0, w_1, \mathsf{st}_\mathcal{B})$.

$\mathcal{B}_{\mathtt{w}2}(\mathsf{st}_\mathcal{B}, crs, \pi^*)$**:** $\mathcal{B}_{\mathtt{w}2}$ runs $(vk^*, sigk^*) \leftarrow \mathsf{SKG}(1^k)$, $\sigma^* \leftarrow \mathsf{Sign}(sigk^*, ((c_i^*)_i, \pi^*))$, $\kappa \leftarrow \mathsf{HKG}(1^k)$, and $h^* = (h_1^* \| \dots \| h_k^*) \leftarrow \mathsf{H}_\kappa(vk^*)$. Next, for each $i \in [k]$, $\mathcal{B}_{\mathtt{w}2}$ sets $pk_i^{(h_i^*)} \leftarrow pk_i'$ and generates $(pk_i^{(1-h_i^*)}, sk_i^{(1-h_i^*)}) \leftarrow \mathsf{PKG}(1^k)$. $\mathcal{B}_{\mathtt{w}2}$ then sets $PK \leftarrow ((pk_i^{(j)})_{i,j}, crs, \kappa)$, $C^* \leftarrow (vk^*, (c_i^*)_i, \pi^*, \sigma^*)$, and $\widehat{SK}_{C^*} \leftarrow (h^*, (sk_i^{(1-h_i^*)})_i, PK)$. $\mathcal{B}_{\mathtt{w}2}$ 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 $\mathcal{B}_\mathtt{w}$. Let $\gamma \in \{0,1\}$ be $\mathcal{B}_\mathtt{w}$'s challenge bit. $\mathcal{B}_\mathtt{w}$'s $\mathtt{WI}$ advantage is estimated as follows:

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

Consider the case when $\gamma = 1$, i.e. $w_1 = ((r_i^*)_i, K_1^*)$ is used as a witness for generating π^*. It is easy to see that in this case, the value C^* is generated in exactly the same way as that in Game 2, and $\mathcal{B}_{\mathtt{w}}$ 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, $w_0 = ((r_i')_i, 0^k)$ is used as a witness for generating π^*, which is exactly how it is generated in Game 3. Since this is the only change from the above, with a similar discussion, we have $\Pr[b' = b | \gamma = 0] = \Pr[\mathsf{Succ}_3]$.

In summary, we have $\mathsf{Adv}^{\mathtt{WI}}_{\mathcal{P},\mathcal{B}_{\mathtt{w}}}(k) = |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$. This completes the proof of Claim 11. □

Claim 12 $\Pr[\mathsf{Succ}_3] = 1/2$.

Proof of Claim 12. In Game 3, C^* is independent of K_1^*. (In particular, π^* is generated by using a witness $((r_i'^*)_i, 0^k)$, and each r_i' is generated by $r_i'^* \leftarrow \mathsf{Explain}(\omega_i^*, 0^k)$.) Since both K_1^* and K_0^* are chosen uniformly at random, $\mathcal{A}$'s view is identically distributed regardless of the challenge bit $b \in \{0, 1\}$. This must mean that the probability that $\mathcal{A}$ succeeds in guessing the challenge bit b is exactly $1/2$. This completes the proof of Claim 12. □

Claims 10 to 12 and the inequality (7) guarantee that there exist PPTAs $\mathcal{B}_{\mathtt{p}}$ and $\mathcal{B}_{\mathtt{w}}$ such that

$$\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma_{\mathtt{DDN}},\mathcal{A}}(k) \le 2 \cdot \left(\mathsf{Adv}^{\mathtt{SNC}}_{\Pi^k,\mathcal{B}_{\mathtt{p}}}(k) + \mathsf{Adv}^{\mathtt{WI}}_{\mathcal{P},\mathcal{B}_{\mathtt{w}}}(k) \right),$$

which, due to our assumptions on the building blocks and Lemma 10, implies that $\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma_{\mathtt{DDN}},\mathcal{A}}(k)$ is negligible. Recall that the choice of the PPTA `eCPA` adversary $\mathcal{A}$ was arbitrarily, and thus for any PPTA `eCPA` adversary $\mathcal{A}$ we can show a negligible upperbound for $\mathsf{Adv}^{\mathtt{eCPA}}_{\Gamma_{\mathtt{DDN}},\mathcal{A}}(k)$ as above. Hence, the PKEM $\Gamma_{\mathtt{DDN}}$ is `eCPA` secure. This completes the proof of Lemma 7. □

E Unpredictability of the Proposed PKEMs

E.1 Strong Unpredictability of $\Gamma_{\mathtt{DDN}}$

Lemma 13. *If the signature scheme Σ is* `SOT` *secure and $\mathcal{H}$ is a UOWHF, then the detectable KEM $\Gamma^{\dagger}_{\mathtt{DDN}} = (\mathsf{KKG}_{\mathtt{DDN}}, \mathsf{Encap}_{\mathtt{DDN}}, \mathsf{Decap}_{\mathtt{DDN}}, \mathsf{F}_{\mathtt{DDN}})$ (which is obtained naturally from the PKEM $\Gamma_{\mathtt{DDN}}$ in Fig. 3) satisfies strong unpredictability.*

Proof of Lemma 13. Let $\mathcal{A}$ be a PPTA adversary that attacks the strong unpredictability of the detectable KEM $\Gamma^{\dagger}_{\mathtt{DDN}}$. Let $(PK = ((pk_i^{(j)})_{i,j}, crs, \kappa), SK)$ be a key pair input to $\mathcal{A}$, $C' = (vk', (c_i')_i, \pi', \sigma')$ be $\mathcal{A}$'s output, and $C^* = (vk^*, (c_i^*)_i, \pi^*, \sigma^*)$ be a ciphertext computed in the `sUNP` experiment.

Let us call $\mathcal{A}$'s output ciphertext C' *successful* if C' makes the `sUNP` experiment output 1, i.e. it holds that $\mathsf{F}_{\mathtt{DDN}}(PK, C^*, C') = 1$, which implies $\mathsf{H}_{\kappa}(vk^*) = \mathsf{H}_{\kappa}(vk')$. Consider two cases $vk^* \ne vk'$ and $vk^* = vk'$. It is easy to see that an adversary $\mathcal{A}$ which outputs a successful ciphertext C' with $vk^* \ne vk'$ can be used to attack the UOWHF $\mathcal{H}$. (Since this is trivial to see, we omit describing a reduction algorithm). Furthermore, with an information-theoretic argument, we can bound the probability of the adversary $\mathcal{A}$ outputting a (even unsuccessful) ciphertext C' containing vk^* to be negligible by the `SOT` security of Σ and Lemma 11 (in Appendix B.2), because it corresponds to guessing an "unseen" verification key vk^*. This completes the proof of Lemma 13. □

E.2 Strong Unpredictability of $\widehat{\Gamma}$

Lemma 14. *If $\mathcal{H}$ is a UOWHF, then the detectable KEM $\widehat{\Gamma}^{\dagger} = (\widehat{\mathsf{KKG}}, \widehat{\mathsf{Encap}}, \widehat{\mathsf{Decap}}, \widehat{\mathsf{F}})$ (which is obtained naturally from the PKEM $\widehat{\Gamma}$ in Fig. 4) satisfies strong unpredictability (even against computationally unbounded adversaries).*

Proof of Lemma 14. Let $\mathcal{A}$ be any (possibly computationally unbounded) adversary that attacks the strong unpredictability of the detectable KEM $\widehat{\Gamma}^{\dagger}$. Fix all randomness in the $\mathtt{sUNP}$ experiment except for κ and K^*, so that they maximize $\mathcal{A}$'s $\mathtt{sUNP}$ advantage. Let $r_{\mathcal{A}}$ be $\mathcal{A}$'s random coin fixed in this step.

Let $F : [\mathsf{HKG}(1^k)] \to \{0,1\}^k$ be the function that takes $\kappa \in [\mathsf{HKG}(1^k)]$ as input, sets up (PK, SK) using κ and the key materials that are already fixed as above, runs $C' = (h', (c_i'), \widetilde{c}') \leftarrow \mathcal{A}(PK, SK; r_{\mathcal{A}})$, and outputs h'.

Let X be the distribution in which $K^* \in \{0,1\}^k$ is picked uniformly at random, calculates $\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, ((r_i^*)_{i \in [k+1]}, K^*); r')$ where α^*, $(r_i^*)_{i \in [k+1]}$, and r' are the values that have been already fixed. Then, X outputs $(c_{k+1}^* \| \widetilde{c}^*)$, where $c_{k+1}^* = \mathsf{Enc}(pk_{k+1}, \alpha^*; r_i^*)$.

Due to the correctness of E, $\mathsf{SEnc}(\alpha^*, \cdot; r')$ is injective for any $\alpha^* \in \mathcal{K}_k$ and any randomness r', and thus the min-entropy of X is exactly that of K^*, i.e. $\mathbf{H}_\infty(X) \geq k \in \omega(\log k)$. Furthermore, X is efficiently samplable because SEnc is efficient.

Recall that $\widehat{\mathsf{F}}(PK, C^*, C') = 1$ if and only if $h^* = \mathsf{H}_\kappa(c_{k+1}^* \| \widetilde{c}^*) = F(\kappa) = h'$. Therefore, by the definitions of F, X, and the $\mathtt{sUNP}$ experiment regarding $\widehat{\Gamma}^{\dagger}$, we have

$$\mathsf{Adv}^{\mathtt{sUNP}}_{\widehat{\Gamma}^{\dagger}, \mathcal{A}}(k) \leq \Pr[\kappa \leftarrow \mathsf{HKG}(1^k);\ (c_{k+1}^* \| \widetilde{c}^*) \leftarrow X : \mathsf{H}_\kappa(c_{k+1}^* \| \widetilde{c}^*) = F(\kappa)].$$

Then, by Lemma 12 (in Appendix B.2), $\mathsf{Adv}^{\mathtt{sUNP}}_{\widehat{\Gamma}^{\dagger}, \mathcal{A}}(k)$ is negligible. Note that this holds even if $\mathcal{A}$ is computationally unbounded, because Lemma 12 holds also for inefficient functions F. This completes the proof of Lemma 14. □