An Asymptotically Optimal Method for Converting Bit Encryption to Multi-Bit Encryption[⋆]

Takahiro Matsuda and Goichiro Hanaoka

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

Abstract. Myers and Shelat (FOCS 2009) showed how to convert a chosen ciphertext secure (CCA secure) PKE scheme that can encrypt only 1-bit plaintexts into a CCA secure scheme that can encrypt arbitrarily long plaintexts (via the notion of key encapsulation mechanism (KEM) and hybrid encryption), and subsequent works improved efficiency and simplicity. In terms of efficiency, the best known construction of a CCA secure KEM from a CCA secure 1-bit PKE scheme, has the public key size $\Omega(k) \cdot |pk|$ and the ciphertext size $\Omega(k^2) \cdot |c|$, where k is a security parameter, and $|pk|$ and $|c|$ denote the public key size and the ciphertext size of the underlying 1-bit scheme, respectively.
In this paper, we show a new CCA secure KEM based on a CCA secure 1-bit PKE scheme which achieves the public key size $2 \cdot |pk|$ and the ciphertext size $(2k + o(k)) \cdot |c|$. These sizes are asymptotically optimal in the sense that they are (except for a constant factor) the same as those of the simplest "bitwise-encrypt" construction (seen as a KEM by encrypting a k-bit random session-key) that works for the chosen plaintext attack and non-adaptive chosen ciphertext attack settings. We achieve our main result by developing several new techniques and results on the "double-layered" construction (which builds a KEM from an inner PKE/KEM and an outer PKE scheme) by Myers and Shelat and on the notion of detectable PKE/KEM by Hohenberger, Lewko, and Waters (EUROCRYPT 2012).

Keywords: public key encryption, key encapsulation mechanism, chosen ciphertext security, double-layered construction, detectable PKE and KEM.

[⋆] An extended abstract of this paper appears in the proceedings of ASIACRYPT 2015. This is the full version.

Table of Contents

1 Introduction

1.1 Background and Motivation

In this paper, we revisit the problem of how to construct a chosen ciphertext secure (CCA2, or just CCA) public key encryption (PKE) scheme that can encrypt plaintexts of arbitrary length from a CCA secure PKE scheme whose plaintext space is only 1-bit. (Hereafter, we call a PKE scheme whose plaintext space is $\{0,1\}^n$ an *n-bit PKE scheme.*) It is well-known that if we only consider chosen plaintext attack (CPA) and non-adaptive chosen ciphertext attack (CCA1) settings, then the simple(st) "*bitwise-encrypt*" construction suffices, in which a plaintext is encrypted bit-by-bit (under the same public key) by a 1-bit PKE scheme, and the concatenation of all ciphertexts is regarded as a ciphertext of the construction. However, for the CCA setting, until recently, the simple question of how (and even whether) one can realize such a "1-bit-to-multi-bit" conversion had been left open.

This open problem was resolved affirmatively by Myers and Shelat [22]. They actually constructed a CCA secure key encapsulation mechanism (KEM) which encrypts a random session-key, and can be used together with a CCA secure symmetric key encryption (SKE) scheme to achieve a full-fledged CCA secure PKE scheme via hybrid encryption [8]. One of the important steps of the approach by Myers and Shelat is to consider the "*double-layered*" construction of a KEM from an "inner" PKE scheme and an "outer" PKE scheme, where the inner ciphertext encrypts a plaintext (or a session-key if one wants to construct a KEM) and a randomness used for outer encryption, and the outer ciphertext encrypts the inner ciphertext using the randomness encrypted in the inner ciphertext. To decrypt a ciphertext, one first decrypts the outer ciphertext, and then the resulting inner ciphertext, to recover a plaintext and a randomness (for outer encryption), and the plaintext is output if the re-encryption of the inner ciphertext using the recovered randomness results in the outer ciphertext. Myers and Shelat showed that if the outer scheme that is built from a 1-bit scheme satisfies the security notion called "unquoted CCA" (UCCA) security (which is a weaker security notion than CCA security that can be considered only for a PKE scheme constructed using another PKE scheme as a building block), and the inner scheme satisfies "1-wise non-malleability against UCCA" (which has a similar flavor to 1-bounded CCA security [7]), the resulting construction achieves CCA security.

The efficiency and simplicity of the construction by Myers and Shelat were improved by Hohenberger, Lewko, and Waters [16]. Specifically, they introduced the notion of a *detectable PKE* scheme, which is a PKE scheme that has an efficiently computable predicate F as part of the syntax, and whose security notions are defined with respect to this F. In particular, they introduced the notions of *detectable CCA* (DCCA) security (which is a relaxed variant of CCA security) and *unpredictability*, and considered a construction which has a mixed flavor of the double-layered construction of Myers and Shelat, and the double (parallel) encryption of Naor and Yung [23] (this construction has two PKE schemes for the outer encryption). They showed that if the "inner" PKE scheme satisfies DCCA security and unpredictability, and the "outer" PKE schemes are CPA secure and 1-bounded CCA secure [7], respectively, then the resulting PKE scheme is CCA secure. They also showed that the "bitwise-encrypt" construction based on a CCA secure 1-bit PKE scheme yields a DCCA secure and unpredictable detectable PKE scheme for long plaintexts, and thus achieves a 1-bit-to-multi-bit conversion for CCA security. (In their construction, in fact a 1-bit scheme satisfying only DCCA security and unpredictability suffices as the building block.) The efficiency of the construction in [16] was further improved by Matsuda and Hanaoka [21] using the ideas and techniques of hybrid encryption.

Despite the elegant ideas employed in [22, 16, 21], however, even in the best construction of [21] (in terms of efficiency), the public key size is $\Omega(k) \cdot |pk|$ and the ciphertext size (when seen as a KEM) is $\Omega(k^2) \cdot |c|$, where k is a security parameter, and $|pk|$ and $|c|$ denote the public key size and the ciphertext size of a CCA secure 1-bit scheme, respectively. On the other hand, for constructing a CPA (resp. CCA1) secure KEM from a CPA (resp. CCA1) secure 1-bit scheme, one can use the above mentioned bitwise-encrypt construction in which one encrypts a k-bit random string and regards this as a session-key of a KEM. Note that the public key size of this KEM is just $|pk|$ and the ciphertext size is $k \cdot |c|$. Compared to this simplest and most straightforward method, in the CCA setting, the known constructions have the public key size and the ciphertext size that are at least $\Omega(k)$ times larger.

Motivated by the above, in this paper we study the following question: *How efficient can a* 1*-bit-to-multi-bit conversion for CCA security be?*

1.2 Our Contributions

As our main result, we show a new 1-bit-to-multi-bit construction for the CCA setting, i.e., a construction of a CCA secure KEM based on a CCA secure 1-bit PKE scheme, with much better asymptotic efficiency than the existing constructions. Specifically, our construction achieves the public key size $2 \cdot |pk|$, and the ciphertext size $(2k + o(k)) \cdot |c| = O(k) \cdot |c|$, which are asymptotically optimal in the sense that these sizes are (except for a constant factor) the same as for the simple bitwise-encrypt construction for CPA and CCA1 security.

We achieve our main result by developing several new techniques and results on the double-layered construction of Myers and Shelat [22] and on the notion of detectable PKE/KEM by Hohenberger, Lewko, and Waters [16]. Our technical contributions in this paper lie in (1) coming up with appropriate security notions for detectable PKE/KEM so that we can conduct CCA security proofs for the double-layered construction using the language of detectable PKE/KEM (without addressing the details of how each of the inner and outer schemes is constructed) which we believe helps us understand our proposed construction (and more generally the double-layered approach itself) in a clearer manner, and (2) showing how one can realize the inner and outer schemes (satisfying the requirements of our security proofs) from a CCA secure 1-bit PKE scheme, so that the resulting CCA secure KEM achieves asymptotically optimal efficiency with respect to the bitwise-encrypt construction.

Below we explain more technical details of our results.

New Security Notions for Detectable PKE/KEM. In Section 3, we introduce new security notions for detectable PKE and detectable KEMs. Recall that DCCA security of [16] is defined like ordinary CCA security, except that in the security experiment, the decryption oracle is restricted according to the predicate F (which is a part of the syntax of detectable PKE/KEM): an adversary is not allowed to query a ciphertext c such that $\mathsf{F}(c^*, c) = 1$ where c^* is the challenge ciphertext. The first notion we introduce is a weak form of *non-malleability* [12, 3, 24] under DCCA that we simply name `wNM-DCCA` *security*, which is defined like DCCA security except that we allow an adversary to make one "unrestricted" decryption query (which is not affected by the restriction of F). We also introduce an even weaker variant, which is a "replayable"-CCA-analogue [4] of `wNM-DCCA` security, which we call `wRNM-DCCA` *security*, that is defined like `wNM-DCCA` security except that the final unrestricted decryption query (and only this query) is answered like a decryption query in the replayable CCA security.

We also introduce a new security notion for detectable PKE/KEM that we call *randomness-inextractability*. Recall that a DCCA secure detectable PKE scheme is meaningful only if it also satisfies another security notion that prevents the predicate F from outputting 1 for every input (which makes DCCA security equivalent to CPA security). *Unpredictability* [16] is one example of a security notion that prevents DCCA security from being trivial, which ensures that a ciphertext c satisfying $\mathsf{F}(c^*, c) = 1$ is hard to find without seeing c^*. Randomness-inextractability is another such security notion for detectable PKE: Informally, it requires that if an adversary is given a ciphertext c^* (that encrypts a plaintext m of the adversary's choice), it cannot come up with a pair of a (possibly different) plaintext m' and randomness r' such that $\mathsf{F}(c^*, c') = 1$, where c' is the encryption of m' generated using the randomness r'. We also show that randomness-inextractability and unpredictability do not imply each other, even if we combine one notion with `wNM-DCCA` security. See Section 3 for the details.

New CCA Security Proofs for the Double-Layered Construction Based on Detectable PKE/KEM. In Section 4, we show our main technical results: two new CCA security proofs for the double-layered construction of Myers and Shelat [22]. Our first security proof shows that if the inner KEM is a detectable KEM satisfying DCCA security and unpredictability, and the outer PKE scheme is a detectable PKE scheme satisfying `wRNM-DCCA` security and randomness-inextractability, then the KEM obtained from the double-layered construction is CCA secure. Our main result with asymptotically optimal efficiency is obtained from this security proof.

Our second security proof shows that if the inner KEM is `wNM-DCCA` secure and unpredictable, and the outer PKE scheme is DCCA secure and randomness-inextractable, then the KEM obtained from the double-layered construction is CCA secure. Interestingly, this security proof can be seen as a generalization of Myers-Shelat's original security proof of their construction [22].

Both of the security proofs have similar flavors to the security proofs of [16, 21]. Namely, DCCA security of the inner KEM guarantees that a session-key (hidden in the challenge ciphertext) is random as long as an adversary does not submit a "dangerous" decryption query (which are defined with respect to the predicate F from the inner detectable KEM), and we then upperbound the probability that the adversary comes up with such "dangerous" decryption queries to be negligible by the combination of the security properties of the outer PKE scheme and the inner KEM. However, unlike the previous works [16, 21] that use a "detectable" primitive only for the inner scheme, we employ a detectable primitive also for the outer scheme. Consequently, we have to deal with two types of "dangerous" decryption queries in the security proofs: an "inner-dangerous" query and an "outer-dangerous" query, which, as the names indicate, are related to the inner KEM and the outer PKE scheme, respectively. Our two security proofs differ in the treatment of the inner- and outer-dangerous queries, which lead to the difference between which of the inner KEM or the outer PKE scheme needs to be "non-malleable" under DCCA. In both of the proofs, randomness-inextractability of the outer PKE scheme is used to show that the adversary's outer-dangerous queries do not help.

We also show an evidence that indicates that our reliance on "non-malleability" under DCCA for either the inner KEM or the outer PKE scheme would be unavoidable, by showing a counterexample for the double-layered construction that does not achieve CCA security if the inner and outer schemes only satisfy DCCA security, unpredictability, and randomness-inextractability. For the details, see Section 4.

A Detectable PKE Scheme Satisfying `wRNM-DCCA` *Security and Randomness-Inextractability from CCA Secure* 1*-bit PKE.* In Section 5, we show a construction of a detectable PKE scheme satisfying `wRNM-DCCA` security and randomness-inextractability, using a CCA secure 1-bit PKE scheme and a *non-malleable code* [14] for "bitwise-tampering and bit-level permutations" [1, 2]. The idea of this construction is based on the recent result by Agrawal et al. [1] who showed how to transform a 1-bit commitment scheme secure against chosen commitment attacks (CCA) into a non-malleable string commitment scheme: We first encode a plaintext by a non-malleable code, and then do "bitwise-encryption" of the encoded value by a CCA secure 1-bit PKE scheme. (Due to its structure, we call this construction the "*Encode-then-Bitwise-Encrypt*" (EtBE) construction.) Our contribution regarding this construction is to clarify that the approach of [1] also works well for detectable PKE as we require.

Agrawal et al. [2] recently constructed a non-malleable code for the above mentioned class of functions with "optimal rate", meaning that the ratio between the length n of a codeword and the length k of a message can be made arbitrarily close to 1 (i.e. $n = k + o(k)$). We employ this non-malleable code to achieve the asymptotic efficiency of our proposed KEM.

The Proposed 1*-Bit-to-Multi-Bit Conversion, and More.* Our main result, i.e. a CCA secure KEM from a CCA secure 1-bit PKE scheme that achieves optimal asymptotic efficiency in terms of the public key and ciphertext sizes, is obtained by using the above mentioned detectable PKE scheme (together with some hybrid encryption techniques) as the outer PKE scheme, and using the bitwise-encrypt construction of a detectable KEM as the inner KEM, in the double-layered construction, via our first security proof. In Section 6, we show the full description of our construction. As noted above, our construction uses only two key pairs of the underlying 1-bit PKE scheme.

Interestingly, there (and in Appendix C), we also show that if a 2-bit PKE scheme can be used instead of a 1-bit PKE scheme, then one can construct a CCA secure KEM (with almost the same construction as our main construction) that uses only one key pair.

On the Necessity of Two Key Pairs. As mentioned above, our proposed KEM from a 1-bit PKE scheme uses two key pairs of the underlying CCA secure 1-bit PKE scheme. Given this, it is natural to ask if the number 2 of key pairs of the underlying 1-bit scheme is optimal for achieving 1-bit-to-multi-bit constructions for CCA security. Although we could not answer this question affirmatively or negatively, in Section 8, we show that the one-key variant of our proposed construction is vulnerable to a CCA attack. This negative result shows a necessity of different techniques and ideas than ours towards answering the question. It also contrasts strikingly with our 2-bit-to-multi-bit construction for CCA security that uses only one key pair of the underlying 2-bit scheme.

We leave it as an open problem to clarify whether one can achieve a 1-bit-to-multi-bit conversion using only one key pair of the underlying 1-bit scheme, or it is generally impossible.

1.3 Related Work

The double-layered construction [22, 16], and extension of the plaintext space of encryption schemes based on it, have been used in several works: Lin and Tessaro [19] showed how to turn a 1-bit PKE scheme whose correctness is not perfect and which only satisfies weak CCA security (weak in the sense that an adversary may have bounded but non-negligible CCA advantage), into a PKE scheme (with a large plaintext space) satisfying ordinary CCA security, via the construction of [16]. Dachman-Soled et al. [9] studied the notion of "enhanced" CCA security for PKE schemes with randomness recovery property, where the decryption oracle in the security experiment returns not only the decryption result of a queried ciphertext but also a randomness that is consistent with the ciphertext, and (among other things) showed that the construction of [16] can be used to achieve a 1-bit-to-multi-bit conversion for enhanced CCA security. Most recently, Kitagawa et al. [18] showed that a simpler variant of the double-layered construction which does not have validity check by re-encryption in the decryption algorithm, can be used to extend the plaintext space of PKE satisfying key-dependent message (KDM) security against CCA with respect to projection functions (projection-KDM-CCA security).

Very recently, Coretti et al. [6] showed a 1-bit-to-multi-bit conversion for a PKE scheme. However, the security notion considered in their construction is so-called "self-destruct" CCA security, which is defined like ordinary CCA security except that in the security experiment, once an adversary submits an invalid ciphertext (which does not decrypt to a valid plaintext) as a decryption query, the decryption oracle "self-destructs", i.e. it will not answer to subsequent decryption queries. This security notion is strictly weaker than ordinary CCA security. Furthermore, in another recent work, Coretti et al. [5] considered non-malleability under self-destruct CCA, which is also strictly weaker than ordinary CCA security, and showed a 1-bit-to-multi-bit conversion for a PKE scheme satisfying this security notion. The 1-bit-to-multi-bit constructions of [6, 5] share the same idea with Agrawal et al.'s conversion (and hence with our "outer" PKE scheme): first encode a plaintext by a suitable non-malleable code, and then do bitwise encryption. The main differences between these works [6, 5] and our "double-layered" construction are: (1) Ours achieves ordinary (full) CCA security, while they achieve weaker security notions. (2) Our construction uses only two key pairs of the underlying 1-bit scheme, while the constructions in [6, 5] use $O(k)$ key pairs, of the building block 1-bit scheme. (3) The requirements of the used non-malleable codes are all different: [6, 5] need stronger form of non-malleability called "continuous" non-malleability [15] (and its extension), while we only need the original definition of non-malleability in [14] that captures "one-time" tampering.; The tampering functions with respect to which non-malleability is considered in [6, 5] are based on bit-wise tampering (extended to take into account continuous non-malleability), while ours requires additionally non-malleability against bit-level permutation (as in [2, 1]).

1.4 Paper Organization

The rest of this paper is organized as follows:

- Section 2 and Appendix A review the basic notation and definitions of the primitives.
- In Section 3, we introduce our new security notions for detectable PKE scheme, and also show several useful facts on them. (New security notions for a detectable KEM is given in Appendix B.)
- In Section 4, we show our main technical results: two new security proofs for the "double-layered" construction. We also explain some evidence that justifies our reliance on non-malleability under DCCA.
- In Section 5, we show how to build a detectable PKE scheme satisfying our new security notions based on a CCA secure 1-bit PKE scheme and a non-malleable code.
- In Section 6, we provide the full description of our 1-bit-to-multi-bit conversion for CCA security. There (and in Appendix C), we also explain our 2-bit-to-multi-bit construction with a single key pair.
- In Section 7, and we give a comparison among 1-bit-to-multi-bit constructions.

– Finally, in Section 8, we show that a variant of our 1-bit-to-multi-bit conversion, in which we only use one key pair of the underlying 1-bit scheme, is vulnerable to a CCA attack.

For theorems and lemmas, we only give proof sketches or intuitive explanations in the main body of the paper, and the formal proofs of them are given in Appendices D or E.

2 Preliminaries

In this section, we review the basic notation and the definitions of the cryptographic primitives. The definitions for standard primitives that are not reviewed in this section are given in Appendix A, which include symmetric key encryption (SKE), message authentication codes (MAC), and a pseudorandom generator (PRG).

2.1 Basic Notation

$\mathbb{N}$ denotes the set of all natural numbers. For $n \in \mathbb{N}$, we define $[n] := \{1, \dots, n\}$. "$x \leftarrow y$" 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 "$(x \overset{?}{=} y)$" is defined to be 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. If furthermore $\mathcal{O}$ is an algorithm, then "$\mathcal{A}^{\mathcal{O}}$" denotes that $\mathcal{A}$ has oracle access to $\mathcal{O}$. A function $\epsilon(\cdot) : \mathbb{N} \to [0, 1]$ is said to be *negligible* if for all positive polynomials $p(k)$ 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.

2.2 (Detectable) Public Key Encryption

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

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

where Dec is a deterministic algorithm, (pk, sk) is a public/secret key pair, and c is a ciphertext of a plaintext m under pk. We say that a PKE scheme satisfies *correctness* if for all $k \in \mathbb{N}$, all keys (pk, sk) output from $\mathsf{PKG}(1^k)$, and all plaintexts m, it holds that $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$.

Detectable PKE. In this paper, we use the notion of *detectable PKE* as defined in [16]. It is a PKE scheme that has a predicate F that tests whether two ciphertexts c and c' are "related" in the sense that to decrypt c, the information of the decryption result of c' is useful (and hence, revealing the decryption result of c' is "dangerous"). This predicate F is used to define multiple security notions of the primitive, and hence we explicitly define it as a part of the syntax of the primitive (this approach is also taken in [16] and [21]).

Formally, a tuple of PPTAs $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{F})$ is said to be a *detectable* PKE scheme if $(\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ constitutes PKE, and F is a predicate that takes a public key pk and two ciphertexts

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

```
$\mathsf{Expt}^{\mathtt{UNP}}_{\Pi,\mathcal{A}}(k)$ :
    $(pk, sk) \leftarrow \mathsf{PKG}(1^k)$
    $(m, c) \leftarrow \mathcal{A}^{\mathsf{Dec}(sk,\cdot)}(pk)$
    $c^* \leftarrow \mathsf{Enc}(pk, m)$
    Return $\mathsf{F}(pk, c^*, c)$
```

```
$\mathsf{Expt}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k)$ :
    $(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
    $(c^*, K_1^*) \leftarrow \mathsf{Encap}(pk)$
    $K_0^* \leftarrow \mathcal{K}$
    $b \leftarrow \{0,1\}$
    $b' \leftarrow \mathcal{A}^{\mathsf{Decap}(sk,\cdot)}(pk, c^*, K_b^*)$
    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 $\mathsf{F}(pk, c^*, c)$
```

```
$\mathsf{Expt}^{\mathcal{F}\text{-}\mathtt{NM}}_{\mathcal{C},\mathcal{A}}(k)$ :
    $(f, m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(1^k)$
    $b \leftarrow \{0,1\}$
    $s^* \leftarrow \mathsf{E}(1^k, m_b)$
    $s' \leftarrow f(s^*)$
    $m' \leftarrow \mathsf{D}(1^k, s')$
    If $m' \in \{m_0, m_1\}$ then $m' \leftarrow \mathsf{same}$
    $b' \leftarrow \mathcal{A}_2(\mathsf{st}, m')$
    Return $(b' \overset{?}{=} b)$
```

Fig. 1. The experiments for defining the security of detectable PKE (left-top/bottom), of detectable KEM (center-top/bottom), and of an $\mathcal{F}$-non-malleable code (right). In the CCA (resp. DCCA) experiment for PKE, $\mathcal{A}_2$ is not allowed to query c^* (resp. ciphertexts c such that $\mathsf{F}(pk, c^*, c) = 1$). The same restrictions apply to $\mathcal{A}$ in the CCA/DCCA experiment for KEMs.

c, c' as input, and outputs either 0 or 1. We require that for all $k \in \mathbb{N}$, all public keys pk output by $\mathsf{PKG}(1^k)$, and all ciphertexts c output by $\mathsf{Enc}(pk, \cdot)$, we have $\mathsf{F}(pk, c, c) = 1$. [1]

Security Notions. Here we recall *chosen ciphertext security (*CCA *security)* for PKE, and *detectable CCA (*DCCA*) security* and *unpredictability* for detectable PKE [16].

Let $\mathtt{ATK} \in \{\mathtt{CCA}, \mathtt{DCCA}\}$. For a (detectable) PKE scheme Π and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, consider the ATK experiment $\mathsf{Expt}^{\mathtt{ATK}}_{\Pi,\mathcal{A}}(k)$ described in Fig. 1 (left-top). In the experiment, it is required that $|m_0| = |m_1|$, and $\mathcal{A}_2$ is not allowed to submit the "prohibited" queries to the decryption oracle: If $\mathtt{ATK} = \mathtt{CCA}$, then the prohibited query is c^*, and if $\mathtt{ATK} = \mathtt{DCCA}$, then the prohibited queries are c satisfying $\mathsf{F}(pk, c^*, c) = 1$. We say that a (detectable) PKE scheme Π is ATK secure if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{ATK}}_{\Pi,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{ATK}}_{\Pi,\mathcal{A}}(k) = 1] - 1/2|$ is negligible.

For a detectable PKE scheme Π (with predicate F) and an adversary $\mathcal{A}$, consider the unpredictability experiment $\mathsf{Expt}^{\mathtt{UNP}}_{\Pi,\mathcal{A}}(k)$ described in Fig. 1 (left-bottom). We say that a detectable PKE scheme Π is *unpredictable* if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{UNP}}_{\Pi,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{UNP}}_{\Pi,\mathcal{A}}(k) = 1]$ is negligible.

2.3 (Detectable) Key Encapsulation Mechanism

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

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

where Decap is a deterministic algorithm, (pk, sk) is a public/secret key pair that defines a session-key space $\mathcal{K}$, and c is a ciphertext of a session-key $K \in \mathcal{K}$ under pk. We say that a KEM satisfies *correctness* if for all $k \in \mathbb{N}$, all keys (pk, sk) output from $\mathsf{KKG}(1^k)$ and all ciphertext/session-key pairs (c, K) output from $\mathsf{Encap}(pk)$, it holds that $\mathsf{Decap}(sk, c) = K$.

[1] This requirement is not explicitly defined in [16], but is actually necessary for DCCA security to be meaningful. Without this requirement, DCCA security is unachievable, as an adversary can submit the challenge ciphertext to the decryption oracle. For our purpose, it is convenient to explicitly require this for F.

We also define a KEM-analogue of detectable PKE, which we call *detectable KEM*, as a KEM that has an efficiently computable predicate F whose interface is exactly the same as that of detectable PKE.

Security Notions. Here we review the definition of $\mathtt{CCA}$ *security* for a KEM, and the definitions of $\mathtt{DCCA}$ *security* and *unpredictability* for a detectable KEM.

Let $\mathtt{ATK} \in \{\mathtt{CCA}, \mathtt{DCCA}\}$. For a (detectable) KEM Γ and an adversary $\mathcal{A}$, consider the $\mathtt{ATK}$ experiment $\mathsf{Expt}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k)$ described in Fig. 1 (center-top). In the experiment, $\mathcal{A}$ is not allowed to submit the "prohibited" queries that are defined in the same way as those for the PKE case. We say that a (detectable) KEM Γ is $\mathtt{ATK}$ secure if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k) := 2 \cdot | \Pr[\mathsf{Expt}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k) = 1] - 1/2|$ is negligible.

For a detectable KEM Γ (with predicate F) and an adversary $\mathcal{A}$, consider the unpredictability experiment $\mathsf{Expt}^{\mathtt{UNP}}_{\Gamma,\mathcal{A}}(k)$ described in Fig. 1 (center-bottom). We say that a detectable KEM Γ is *unpredictable* if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{UNP}}_{\Gamma,\mathcal{A}}(k) = 1]$ is negligible.

2.4 Non-malleable Codes

Here, we recall the definition of *non-malleable codes* [14].

A code $\mathcal{C}$ with message length $\kappa = \kappa(k)$ and codeword length $n = n(k)$ (called also an (n, κ)-code) consists of the two PPTAs (E, D): E is the encoding algorithm that takes 1^k and a message $m \in \{0,1\}^\kappa$ as input, and outputs a codeword $c \in \{0,1\}^n$.; D takes 1^k and c as input, and outputs $m \in \{0,1\}^\kappa$ or the special symbol $\bot$ indicating that c is invalid. We require for all $k \in \mathbb{N}$ and all messages $m \in \{0,1\}^\kappa$, it holds that $\mathsf{D}(1^k, \mathsf{E}(1^k, m)) = m$.

Non-malleability. Non-malleability for codes, formalized by Dziembowski et al. [14], is defined with respect to a class of tampering functions $\mathcal{F}$. Intuitively, non-malleability guarantees that if an encoding c of a message m is modified into $c' = f(c)$ by a function $f \in \mathcal{F}$, then the decoded value m' of c' is either the original message m itself, or a completely unrelated message (or $\bot$). Here we recall the indistinguishability-based definition which is most convenient for us to work with, which is called the "alternative-non-malleability" in [13, Definition A.1]. It was shown in [13] that this definition is equivalent to the original simulation-based definition for codes whose message length κ is superlogarithmic in k.

Let $n, \kappa : \mathbb{N} \to \mathbb{N}$ be positive polynomials of k such that $n(k) \geq \kappa(k)$. For an (n, κ)-code $\mathcal{C} = (\mathsf{E}, \mathsf{D})$, a class of functions $\mathcal{F} = \{\mathcal{F}_k : \{0,1\}^k \to \{0,1\}^k\}_{k \in \mathbb{N}}$, and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, consider the $\mathcal{F}\text{-}\mathtt{NM}$ experiment $\mathsf{Expt}^{\mathcal{F}\text{-}\mathtt{NM}}_{\mathcal{C},\mathcal{A}}(k)$ described in Fig. 1 (right). In the experiment, "same" is the special symbol indicating that the decoded message m' was either m_0 or m_1, and it is required that $f \in \mathcal{F}_n$ and $|m_0| = |m_1| = \kappa(k)$. We say that $\mathcal{C}$ is *non-malleable with respect to the function class* $\mathcal{F}$ ($\mathcal{F}$-non-malleable, for short) if for all PPTAs[2] $\mathcal{A}$, $\mathsf{Adv}^{\mathcal{F}\text{-}\mathtt{NM}}_{\mathcal{C},\mathcal{A}}(k) := 2 \cdot | \Pr[\mathsf{Expt}^{\mathcal{F}\text{-}\mathtt{NM}}_{\mathcal{C},\mathcal{A}}(k) = 1] - 1/2|$ is negligible. We also say that $\mathcal{C}$ is an $\mathcal{F}$-non-malleable code.

Classes of Tampering Functions. In this paper, we consider the following classes of functions.

Composition of "Bitwise Tampering" and "Bit-Level Permutation" $\mathcal{P}$ [2, 1]: Let $\mathtt{set}, \mathtt{reset}, \mathtt{forward}, \mathtt{toggle} : \{0,1\} \to \{0,1\}$ be the functions over a bit, defined by $\mathtt{set}(x) := 1$,

[2] The original definition [14] considered security against computationally unbounded adversaries. In this paper, however, we only need security against PPTAs.

$\texttt{reset}(x) := 0$, $\texttt{forward}(x) := x$, and $\texttt{toggle}(x) := 1 - x$. We define $\mathcal{F}_{\texttt{BIT}} := \{\texttt{set}, \texttt{reset}, \texttt{forward}, \texttt{toggle}\}$.

Let $\mathcal{P} = \{\mathcal{P}_n\}_{n \in \mathbb{N}}$ be the class of functions which first perform "bitwise tampering" to an input, followed by a "bit-level permutation." Namely, $\mathcal{P}_n$ is the set of all functions $f : \{0,1\}^n \to \{0,1\}^n$ that can be described by using n bitwise-tampering functions $f_1, \dots, f_n \in \mathcal{F}_{\texttt{BIT}}$ and a permutation $\pi : [n] \to [n]$, as follows:

$$x = (x_1 \| \dots \| x_n) \overset{f}{\mapsto} \Big(f_{\pi^{-1}(1)}(x_{\pi^{-1}(1)}) \; \| \; \dots \; \| \; f_{\pi^{-1}(n)}(x_{\pi^{-1}(n)}) \Big).$$

"Bit-Fixing" or "Quoting an Input without Duplicated Positions" $\mathcal{Q}$: Let $\texttt{one} : \{0,1\}^n \to \{0,1\}$ and $\texttt{zero} : \{0,1\}^n \to \{0,1\}$ be the constant functions that output 1 and 0 for any n-bit inputs, respectively. Furthermore, for $j \in [n]$, let $\texttt{quote}^j : \{0,1\}^n \to \{0,1\}$ be the "quoting" function that always outputs the j-th bit of its input.

Let $\mathcal{Q} = \{\mathcal{Q}_n\}_{n \in \mathbb{N}}$ be the class of functions each of whose output bits is either a "fixed value" or "quoting the input without duplicated positions." More formally, $\mathcal{Q}_n$ is the set of all functions $f : \{0,1\}^n \to \{0,1\}^n$ that can be decomposed to n functions $f_1, \dots, f_n : \{0,1\}^n \to \{0,1\}$ so that $f(x) = (f_1(x) \| \dots \| f_n(x))$ for all $x \in \{0,1\}^n$, and furthermore it holds that for every $i \in [n]$:

$$f_i \in \{\texttt{one}, \texttt{zero}\} \cup \Big(\{\texttt{quote}^j\}_{j \in [n]} \backslash \{f_j\}_{j \in [i-1]} \Big). \qquad (1)$$

Note that the above guarantees that there exist no indices $i, i', j \in [n]$ such that $f_i = f_{i'} = \texttt{quote}^j$ and $i \neq i'$. We call this condition the *no duplicated quoting condition.*

Agrawal et al. [2] showed the following elegant result, which is crucial for the efficiency of our proposed KEM:

Lemma 1. *([2]) There exists an explicit (n,k)-code such that (1) it is $\mathcal{P}$-non-malleable, and (2) its "rate", defined by k/n, asymptotically approaches to 1 as k increases (and hence $n = k + o(k)$).*

Furthermore, the following is implicitly used by Agrawal et al. [1], and also is useful for our purpose. Although it is almost straightforward from the definitions of $\mathcal{P}$ and $\mathcal{Q}$, we show its formal proof in Appendix E.1 for self-containment.

Lemma 2. *For all $n \in \mathbb{N}$, $\mathcal{Q}_n \subseteq \mathcal{P}_n$. (This holds even if $\mathcal{F}_{\texttt{BIT}}$ does not contain* toggle*.) Hence, any $\mathcal{P}$-non-malleable code is also $\mathcal{Q}$-non-malleable.*

3 New Security Notions for Detectable PKE and KEM

In this section, we introduce new security notions for detectable PKE: wNM-DCCA *security* and wRNM-DCCA *security* in Section 3.1, and *randomness-inextractability* in Section 3.2. We also show some useful facts regarding the new security notions in Section 3.3.

We also define wNM-DCCA security and randomness-inextractability for detectable KEMs. Since their definitions are straightforward KEM-analogues of those for detectable PKE in this section, we provide them in Appendix B.

3.1 "Weak" Non-malleability under DCCA and Its "Replayable" Variant

Here, we define a "weak" form of non-malleability against DCCA for detectable PKE, which we call wNM-DCCA *security*, that captures the intuition that a DCCA adversary who works in the DCCA experiment cannot come up with a ciphertext that is "meaningfully related" to the challenge ciphertext. Recall that the original definitions of non-malleability for PKE [12, 3, 24] ensure that an adversary cannot come up with even a vector of ciphertexts that are "meaningfully related" to the challenge ciphertext, while our notion here only requires that it cannot come up with only a single related ciphertext. Technically, following the formalizations in [3, 24, 22], we formalize wNM-DCCA security by modifying the original DCCA experiment (in which originally the usage of the decryption oracle is restricted according the predicate F of detectable PKE), so that at the end of the experiment an adversary is allowed to make a single "unrestricted" decryption query, regardless of F. Thus, it is like "1-bounded" CCA security [7], albeit an adversary has additionally access to the DCCA decryption oracle. Myers and Shelat [22] defined a security notion for PKE-to-PKE constructions called "q-wise-non-malleability under UCCA." Our definition of wNM-DCCA security is a detectable-PKE-analogue of their 1-wise-non-malleability.

We also define a weaker variant of wNM-DCCA security, in the security experiment of which the final "unrestricted" decryption query (and only this query) is answered like a decryption query in the "replayable" CCA experiment [4], namely, if the decryption result is one of the challenge plaintexts that an adversary uses, then the adversary is only informed so and is not given the actual decryption result. Due to the lack of a better name, we call it wRNM-DCCA security (where R stands for "**R**eplayable").

Fomally, for a detectable PKE scheme $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{F})$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, we define the wNM-DCCA experiment $\mathsf{Expt}^{\mathtt{wNM\text{-}DCCA}}_{\Pi,\mathcal{A}}(k)$ and the wRNM-DCCA experiment $\mathsf{Expt}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi,\mathcal{A}}(k)$ as described in Fig. 2 (left and center, respectively). In both of the experiments, it is required that $|m_0| = |m_1|$, and as in the DCCA experiment, $\mathcal{A}_2$ is not allowed to submit a decryption query c satisfying $\mathsf{F}(pk, c^*, c) = 1$ to the decryption oracle. The adversary's final "unrestricted" decryption query is captured by the ciphertext c' that is finally output by $\mathcal{A}_2$, and naturally it is required that $c' \neq c^*$. *However, we allow c' to be such that* $\mathsf{F}(pk, c^*, c') = 1$. In the wRNM-DCCA experiment, "same" is the special symbol (which is distinguished from $\bot$) that indicates that $\mathsf{Dec}(sk, c') \in \{m_0, m_1\}$.

Definition 1. *We say that a detectable PKE scheme Π is* wNM-DCCA *secure if for all PPTAs $\mathcal{A}$,* $\mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\Pi,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{wNM\text{-}DCCA}}_{\Pi,\mathcal{A}}(k) = 1] - 1/2|$ *is negligible. We define* wRNM-DCCA *security analogously.*

3.2 Randomness-Inextractability

Here we introduce another security notion for detectable PKE that we call *randomness-inextractability.* Roughly, this security notion ensures that given the challenge ciphertext c^* (which is an encryption of a plaintext of an adversary's choice), an adversary cannot come up with a pair (m', r') of a plaintext and a randomness such that $\mathsf{F}(pk, c^*, \mathsf{Enc}(pk, m'; r')) = 1$. If the predicate $\mathsf{F}(pk, c^*, c')$ tests the equality ($c^* \overset{?}{=} c'$), then this notion exactly demands that the randomness used in c^* cannot be recovered, and hence we use the name "randomness-inextractability" (although we allow more general predicates for F).

Formally, for a detectable PKE scheme $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{F})$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, consider the R-Inext experiment described in Fig. 2 (right).

```
$\mathsf{Expt}^{\mathtt{wNM\text{-}DCCA}}_{\Pi,\mathcal{A}}(k)$ :
    $(pk, sk) \leftarrow \mathsf{PKG}(1^k)$
    $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(pk)$
    $b \leftarrow \{0,1\}$
    $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
    $(c', \mathsf{st}') \leftarrow \mathcal{A}_2^{\mathsf{Dec}(sk,\cdot)}(\mathsf{st}, c^*)$
    $m' \leftarrow \mathsf{Dec}(sk, c')$
    $b' \leftarrow \mathcal{A}_3(\mathsf{st}', m')$
    Return $(b' \stackrel{?}{=} b)$.
```

```
$\mathsf{Expt}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi,\mathcal{A}}(k)$ :
    $(pk, sk) \leftarrow \mathsf{PKG}(1^k)$
    $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(pk)$
    $b \leftarrow \{0,1\}$
    $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
    $(c', \mathsf{st}') \leftarrow \mathcal{A}_2^{\mathsf{Dec}(sk,\cdot)}(\mathsf{st}, c^*)$
    $m' \leftarrow \mathsf{Dec}(sk, c')$
    If $m' \in \{m_0, m_1\}$ then $m' \leftarrow \mathsf{same}$
    $b' \leftarrow \mathcal{A}_3(\mathsf{st}', m')$
    Return $(b' \stackrel{?}{=} b)$.
```

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

Fig. 2. Security experiments for `wNM-DCCA` security (left), `wRNM-DCCA` security (center), and randomness-inextractability (right). In the `wNM`/`wRNM-CCA` experiments, the decryption oracle for $\mathcal{A}_2$ has the same restriction as in the `DCCA` experiment.

Definition 2. *We say that a detectable PKE scheme Π satisfies* randomness-inextractability *if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{R\text{-}Inext}}_{\Pi,\mathcal{A}}(k) = 1]$ is negligible.*

Remark. We could have defined the randomness-inextractability experiment so that we let an adversary choose its challenge message m after given a public key pk. This makes the security stronger. However, we do not need this stronger variant for our results.

3.3 Useful Facts

Stretching a Session-Key. As in the case of ordinary KEMs, for a detectable KEM, session-keys can be stretched by using a PRG. More formally, let $\Gamma = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{F})$ be a detectable KEM whose session-key space is $\{0,1\}^k$. Let $\mathsf{G} : \{0,1\}^k \to \{0,1\}^\ell$ be a PRG with $\ell = \ell(k) > k$, where for convenience we define $\mathsf{G}(\bot) := \bot$. (The formal definition of a PRG can be found in Appendix A.3.) Then, consider the detectable KEM $\Gamma' = (\mathsf{KKG}, \mathsf{Encap}', \mathsf{Decap}', \mathsf{F})$ whose session-key space is $\{0,1\}^\ell$, which is naturally constructed by combining Γ and G: $\mathsf{Encap}'(pk)$ runs $(c, K) \leftarrow \mathsf{Encap}(pk)$ and outputs a ciphertext/session key pair $(c, \mathsf{G}(K))$.; We define $\mathsf{Decap}'(sk, c) := \mathsf{G}(\mathsf{Decap}(sk, c))$. The following is straightforward, and thus its proof is omitted.

Lemma 3. *If the detectable KEM Γ satisfies randomness-inextractability (resp. unpredictability), then so does the detectable KEM Γ'. Furthermore, if Γ is* `DCCA` *(resp.* `wNM-DCCA`*) secure and* G *is a PRG, then Γ' is* `DCCA` *(resp.* `wNM-DCCA`*) secure.*

Hybrid Encryption. For a detectable PKE scheme, a straightforward application of hybrid encryption preserves `w(R)NM-DCCA` security and randomness-inextractability, when combined with a `CCA` secure SKE scheme. Since a `CCA` secure SKE scheme with "zero" ciphertext overhead can be realized from a strong pseudorandom permutation [25] (which in turn can be realized based on any one-way function), the ciphertext overhead of a detectable PKE scheme with `w(R)NM-DCCA` security and randomness-inextractability, can be as small as the ciphertext size of the scheme for encrypting a random session-key (usually a k-bit string).

Formally, let $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{F})$ be a detectable PKE scheme where the randomness space of Enc is $\{0,1\}^\ell$, and let $E = (\mathsf{SEnc}, \mathsf{SDec})$ be a deterministic SKE scheme (i.e. its encryption algorithm SEnc is deterministic). (The formal definition of SKE can be found in Appendix A.1.) Then, we naturally construct the detectable PKE scheme $\Pi_{\mathtt{HYB}} = (\mathsf{PKG}_{\mathtt{HYB}}, \mathsf{Enc}_{\mathtt{HYB}}, \mathsf{Dec}_{\mathtt{HYB}}, \mathsf{F}_{\mathtt{HYB}})$ via

$\mathsf{PKG}_{\mathtt{HYB}}(1^k)$:	$\mathsf{Enc}_{\mathtt{HYB}}(pk, m; R)$:	$\mathsf{Dec}_{\mathtt{HYB}}(sk, C)$:	$\mathsf{F}_{\mathtt{HYB}}(pk, C^*, C')$:
$(pk, sk) \leftarrow \mathsf{PKG}(1^k)$. Return (pk, sk).	Parse R as $(r, K) \in \{0,1\}^\ell \times \{0,1\}^k$. $c \leftarrow \mathsf{Enc}(pk, K; r)$ $\widehat{c} \leftarrow \mathsf{SEnc}(K, m)$ Return $C \leftarrow (c, \widehat{c})$.	$(c, \widehat{c}) \leftarrow C$ $K \leftarrow \mathsf{Dec}(sk, c)$ If $K = \bot$ then return $\bot$. Return $m \leftarrow \mathsf{SDec}(K, \widehat{c})$.	$(c^*, \widehat{c}^*) \leftarrow C^*$ $(c', \widehat{c}') \leftarrow C'$ Return $\mathsf{F}(pk, c^*, c')$.

Fig. 3. Hybrid encryption $\Pi_{\mathtt{HYB}}$ for detectable PKE.

hybrid encryption, as in Fig. 3. (In the figure, we describe the randomness of $\mathsf{Enc}_{\mathtt{HYB}}$ explicitly so that it is convenient to consider its randomness-inextractability.) The randomness space of $\mathsf{Enc}_{\mathtt{HYB}}$ is $\{0,1\}^{\ell+k}$. Regarding the security of the hybrid encryption construction, the following lemma is straightforward to see. However, since this hybrid encryption is used in the "outer" PKE scheme in our final construction to expand its plaintext space, we provide the formal proof in Appendix E.2 to make the paper self-contained.

Lemma 4. *If the detectable PKE scheme Π is* wNM-DCCA *secure (resp.* wRNM-DCCA *secure) and the SKE scheme E is* CCA *secure, then the detectable PKE scheme $\Pi_{\mathtt{HYB}}$ in Fig. 3 is* wNM-DCCA *secure (resp.* wRNM-DCCA *secure). Furthermore, if Π satisfies randomness-inextractability (resp. unpredictability), then so does $\Pi_{\mathtt{HYB}}$.*

From wRNM-DCCA *Security to* wNM-DCCA *Security.* Canetti, Krawczyk, and Nielsen [4] showed how to convert a "replayable" CCA secure PKE scheme into an ordinary CCA secure KEM, using a message authentication code (MAC), with almost no overhead. This method can be used for converting a wRNM-DCCA secure detectable PKE scheme into a wNM-DCCA secure detectable KEM. For self-containment, we review this transformation in Appendix E.3.

On the Non-triviality of Randomness-Inextractability. One might wonder whether there is an implication from randomness-inextractability to unpredictability and/or vice versa (especially in case if a detectable PKE scheme already satisfies wNM-DCCA security). We show that this is *not* the case, for both directions. Specifically, via artificial counterexamples, we can show the following lemma that shows the non-triviality of these notions, which we formally show in Appendix D.1.

Lemma 5. *A detectable PKE scheme satisfying* wNM-DCCA *security and unpredictability simultaneously does not necessarily satisfy randomness-inextractability. Furthermore, a detectable PKE scheme satisfying* wNM-DCCA *security and randomness-inextractability simultaneously does not necessarily satisfy unpredictability.*

4 Chosen Ciphertext Security of the Double-Layered Construction

In this section, we show our main result: two new CCA security proofs for the "double-layered" construction $\Gamma_{\mathtt{DL}}$ (of a KEM) constructed based on the "inner" detectable KEM $\Gamma_{\mathtt{in}}$ and the "outer" detectable PKE scheme $\Pi_{\mathtt{out}}$. We also show a partial evidence that we need to rely on "non-malleability" that we defined in the previous section.

The Double-Layered Construction. Let $\Pi_{\mathtt{out}} = (\mathsf{PKG}_{\mathtt{out}}, \mathsf{Enc}_{\mathtt{out}}, \mathsf{Dec}_{\mathtt{out}}, \mathsf{F}_{\mathtt{out}})$ be a detectable PKE scheme. We assume the plaintext space of $\Pi_{\mathtt{out}}$ to be $\{0,1\}^n$ (where $n = n(k)$ is determined below), and the randomness space of $\mathsf{Enc}_{\mathtt{out}}$ to be $\{0,1\}^\ell$ for some positive polynomial $\ell = \ell(k)$. Let $\Gamma_{\mathtt{in}} = (\mathsf{KKG}_{\mathtt{in}}, \mathsf{Encap}_{\mathtt{in}}, \mathsf{Decap}_{\mathtt{in}}, \mathsf{F}_{\mathtt{in}})$ be a detectable KEM such that the ciphertext length is n bit, and

```
$\mathsf{KKG}_{\mathtt{DL}}(1^k)$ :
    $(pk_{\mathrm{in}}, sk_{\mathrm{in}}) \leftarrow \mathsf{KKG}_{\mathtt{in}}(1^k)$
    $(pk_{\mathrm{out}}, sk_{\mathrm{out}}) \leftarrow \mathsf{PKG}_{\mathtt{out}}(1^k)$
    $PK \leftarrow (pk_{\mathrm{in}}, pk_{\mathrm{out}})$
    $SK \leftarrow (sk_{\mathrm{in}}, sk_{\mathrm{out}}, PK)$
    Return $(PK, SK)$.
```

```
$\mathsf{Encap}_{\mathtt{DL}}(PK)$ :
    $(pk_{\mathrm{in}}, pk_{\mathrm{out}}) \leftarrow PK$
    $(c_{\mathrm{in}}, \alpha) \leftarrow \mathsf{Encap}_{\mathtt{in}}(pk_{\mathrm{in}})$
    Parse $\alpha$ as $(r, K) \in \{0,1\}^\ell \times \{0,1\}^k$.
    $c \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathrm{out}}, c_{\mathrm{in}}; r)$
    Return $(c, K)$.
```

```
$\mathsf{Decap}_{\mathtt{DL}}(SK, c)$ :
    $(sk_{\mathrm{in}}, sk_{\mathrm{out}}, PK) \leftarrow SK$
    $(pk_{\mathrm{in}}, pk_{\mathrm{out}}) \leftarrow PK$
    $c_{\mathrm{in}} \leftarrow \mathsf{Dec}_{\mathtt{out}}(sk_{\mathrm{out}}, c)$
    If $c_{\mathrm{in}} = \bot$ then return $\bot$.
    $\alpha \leftarrow \mathsf{Decap}_{\mathtt{in}}(sk_{\mathrm{in}}, c_{\mathrm{in}})$
    If $\alpha = \bot$ then return $\bot$.
    Parse $\alpha$ as $(r, K) \in \{0,1\}^\ell \times \{0,1\}^k$.
    If $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathrm{out}}, c_{\mathrm{in}}; r) = c$
            then return $K$ else return $\bot$.
```

Fig. 4. The double-layered KEM construction $\mathit{\Gamma}_{\mathtt{DL}}$ from a detectable PKE scheme $\mathit{\Pi}_{\mathtt{out}}$ and a detectable KEM $\mathit{\Gamma}_{\mathtt{in}}$.

the session-key space is $\{0,1\}^{\ell+k}$. Then we construct the "double-layered" KEM $\mathit{\Gamma}_{\mathtt{DL}} = (\mathsf{KKG}_{\mathtt{DL}}, \mathsf{Encap}_{\mathtt{DL}}, \mathsf{Decap}_{\mathtt{DL}})$ as in Fig. 4. For convenience, we occasionally call $\mathit{\Gamma}_{\mathtt{in}}$ the *inner* KEM and $\mathit{\Pi}_{\mathtt{out}}$ the *outer* PKE scheme.

Our First Security Proof. The $\mathtt{CCA}$ security of $\mathit{\Gamma}_{\mathtt{DL}}$ can be shown as follows.

Theorem 1. *Assume that the "outer" PKE scheme $\mathit{\Pi}_{\mathtt{out}}$ is a detectable PKE scheme satisfying* $\mathtt{wRNM\text{-}DCCA}$ *security and randomness-inextractability, and the "inner" KEM $\mathit{\Gamma}_{\mathtt{in}}$ is a detectable KEM satisfying* $\mathtt{DCCA}$ *security and unpredictability. Then, the KEM $\mathit{\Gamma}_{\mathtt{DL}}$ in Fig. 4 is* $\mathtt{CCA}$ *secure.*

The formal proof is given in Appendix D.2. The structure of the proof is similar to the security proofs for the constructions by Hohenberger et al. [16] and by Matsuda and Hanaoka [21]. However, the details differ due to the difference in the construction and the used assumptions.

We explain the ideas for the proof of Theorem 1. (Here, the values with asterisk (*) represent those related to the challenge ciphertext c^*.) As the first step, note that since a session-key K of $\mathit{\Gamma}_{\mathtt{DL}}$ is part of a session-key $\alpha = (r \| K)$ of the $\mathtt{DCCA}$ secure inner KEM $\mathit{\Gamma}_{\mathtt{in}}$, unless a $\mathtt{CCA}$ adversary $\mathcal{A}$ submits a decapsulation query c that simultaneously satisfies (1) $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \neq \bot$ and (2) $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$, $\mathcal{A}$ has no chance in distinguishing the real session-key K^*_1 from a random K^*_0. Following [16, 21], we call this type of decapsulation query a *dangerous* query. If the probability that $\mathcal{A}$ comes up with a dangerous query is negligible, then we can finish the proof. Furthermore, observe that since $\mathit{\Gamma}_{\mathtt{in}}$ satisfies unpredictability, if we can ensure that the information of the inner ciphertext $c^*_{\mathtt{in}}$ is hidden from $\mathcal{A}$'s view, then the probability that $\mathcal{A}$ comes up with a dangerous query is negligible.

To show that the probability that $\mathcal{A}$ comes up with a dangerous query in the original security game is negligibly close to that in the security game in which $\mathcal{A}$'s view does not contain $c^*_{\mathtt{in}}$ at all (and hence we can invoke the unpredictability of $\mathit{\Gamma}_{\mathtt{in}}$), we rely on the security properties of the outer PKE scheme $\mathit{\Pi}_{\mathtt{out}}$ to gradually change the security game for $\mathcal{A}$ so that in the final game, c^* as well as other values in $\mathcal{A}$'s view contain no information on $c^*_{\mathtt{in}}$. Note that in the actual encapsulation algorithm $\mathsf{Encap}_{\mathtt{DL}}$, the randomness r used for outer encryption is also a part of the session-key α of the inner KEM. Thus, once we invoke the $\mathtt{DCCA}$ security of the inner KEM $\mathit{\Gamma}_{\mathtt{in}}$ (which we have already done as the first step), not only the real session-key K^*_1 but also the randomness r^* used to generate the challenge ciphertext c^* are made uniformly random values, which enables us to rely on the security properties of $\mathit{\Pi}_{\mathtt{out}}$ from that point on.

Now, intuitively, the $\mathtt{DCCA}$ security (which is implied by $\mathtt{wRNM\text{-}DCCA}$ security) of $\mathit{\Pi}_{\mathtt{out}}$ guarantees that $c^*_{\mathtt{in}}$ is hidden from $\mathcal{A}$'s view as long as $\mathcal{A}$ only submits a decapsulation query c such that $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 0$. However, $\mathcal{A}$ is free to choose its own decapsulation query, and may submit c

such that $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$. As mentioned in Section 1.2, this is another type of "dangerous" query, in the sense that the condition $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ prevents us from relying on the DCCA security of the outer PKE scheme $\Pi_{\mathtt{out}}$. To distinguish this from the above mentioned type of dangerous queries with respect to the inner KEM, let us use the names "*inner-dangerous* queries" and "*outer-dangerous* queries" which are associated with the inner KEM and the outer PKE scheme, respectively.

In the full proof, we will show that the randomness-inextractability of the outer PKE scheme allows us to reject decapsulation queries c satisfying $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$, without being noticed by $\mathcal{A}$. Intuitively, this is possible because in order for $\mathcal{A}$ to notice the difference between a security game in which a decapsulation query c with $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ is not rejected and a security game in which such c is rejected, $\mathcal{A}$ has to come up with a "valid" query c satisfying $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ and $\mathsf{Decap}_{\mathtt{DL}}(SK, c) \neq \bot$. However, the latter condition implies $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \neq \bot$, $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c_{\mathtt{in}}) = (r \| K) \neq \bot$, and $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c_{\mathtt{in}}; r) = c$, among which the combination of $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ and $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c_{\mathtt{in}}; r) = c$ is exactly the condition of violating randomness-inextractability, and thus such a valid query c must be hard to find.

If we can safely reject an outer-dangerous query, one might wonder why we need non-malleability for the outer PKE scheme, and why ordinary DCCA security is not sufficient. The reason is that although DCCA security of $\Pi_{\mathtt{out}}$ intuitively ensures that $\mathcal{A}$ cannot "see" the inner challenge ciphertext $c^*_{\mathtt{in}}$, it does not prevent $\mathcal{A}$ from coming up with an inner-dangerous decapsulation query c such that $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$. From the viewpoint of the security proof, we may be able to come up with a DCCA adversary (a reduction algorithm) for $\Pi_{\mathtt{out}}$ that perfectly simulates the security game (in which queries c with $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ are rejected) for $\mathcal{A}$. However, such DCCA adversary cannot check if $\mathcal{A}$'s query satisfying $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ is an inner-dangerous query due to the restriction on the decryption oracle.

This is the place where the non-malleability of the outer PKE scheme comes into play. Note that an inner ciphertext is a "plaintext" of the outer PKE scheme, and the notion of "inner-dangerous queries" is a "meaningful relation" between $c^*_{\mathtt{in}}$ and another inner ciphertext. Therefore, the wRNM-DCCA security of $\Pi_{\mathtt{out}}$ ensures that $\mathcal{A}$ cannot come up with even a single inner-dangerous query c, as long as $\mathcal{A}$ can only observe the decapsulation results of queries c' satisfying $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c') = 0$. From the viewpoint of the security proof, if a reduction algorithm is a wRNM-DCCA adversary for $\Pi_{\mathtt{out}}$, it can check if $\mathcal{A}$'s query c is inner-dangerous by its final "unrestricted" decryption query, even if $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ holds. This enables us to finally show that the probability that $\mathcal{A}$ comes up with an inner-dangerous query in the original security game, is negligibly close to the probability that $\mathcal{A}$ does so in the game in which $\mathcal{A}$'s view does not contain the information on $c^*_{\mathtt{in}}$.

Hence, combining all the security properties of the building blocks leads to CCA security. However, the explanation so far hides some technical subtleties that arise due to the "replayable-CCA"-like nature of wRNM-DCCA security, and the treatment of the cases where $\mathcal{A}$'s decapsulation query c satisfies $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c^*_{\mathtt{in}}$, etc. For the details, see Appendix D.2.

Our Second Proof. We show an alternative security proof for the double-layered construction based on slightly different assumptions on the building blocks.

Theorem 2. *Assume that the "outer" PKE scheme $\Pi_{\mathtt{out}}$ is a detectable PKE scheme satisfying* DCCA *security and randomness-inextractability, and the "inner" KEM $\Gamma_{\mathtt{in}}$ is a detectable KEM satisfying* wNM-DCCA *security and unpredictability. Then, the KEM $\Gamma_{\mathtt{DL}}$ in Fig. 4 is* CCA *secure.*

The formal proof is given in Appendix D.3. Recall that Myers and Shelat's original double-layered construction uses an "unquoted" CCA (UCCA) secure construction of a PKE scheme for the outer PKE scheme and a construction of a KEM which is "1-wise-non-malleable under UCCA" for the inner KEM, where UCCA security and its non-malleable variant are security notions considered for PKE-to-PKE constructions (i.e. constructions that use another PKE scheme as a building block). Recall also that $\mathtt{DCCA}$ security is an abstraction of UCCA security [16], from a security notion for a PKE-to-PKE construction to that of a wider notion of detectable PKE. Analogously, our definition of $\mathtt{wNM\text{-}DCCA}$ security can be seen as an abstraction of Myers and Shelat's "1-wise non-malleability under UCCA". Furthermore, we can easily see that the actual instantiations of the inner KEM and the outer PKE scheme used in the original Myers-Shelat construction [22], when respectively seen as a detectable KEM and a detectable PKE scheme, satisfy unpredictability and randomness-inextractability. Therefore, Theorem 2 can be seen as a generalization of Myers and Shelat's result, stated using the language of detectable PKE/KEM.

The structure of the proof of Theorem 2 is similar to our first proof. However, there are several subtle but crucial differences. In particular, the definitions of "inner/outer-dangerous queries" are different from those used in the proof of Theorem 1, and correspondingly we consider a different ordering of the sequence of games for this proof. Furthermore, the role of the "non-malleability" in this proof and that of the proof of Theorem 1 are different. Informally speaking, in this proof, the $\mathtt{wNM\text{-}DCCA}$ security of the inner detectable KEM $\mathit{\Gamma}_{\mathtt{in}}$ is used to ensure that the probability that a $\mathtt{CCA}$ adversary comes up with an outer-dangerous query is not noticeably different between the games in which we invoke (the indistinguishability property of) the $\mathtt{DCCA}$ security of the inner KEM. For the details, see Appendix D.3.

Can We Avoid $\mathtt{w(R)NM\text{-}DCCA}$ *Security?* Both of our security proofs for the $\mathtt{CCA}$ security of the double-layered construction require either the inner detectable KEM or the outer detectable PKE scheme to be "non-malleable" under $\mathtt{DCCA}$.

Looking ahead, in the next section, we will see that the simplest "bitwise-encrypt" construction based on $\mathtt{CCA}$ secure 1-bit PKE satisfies $\mathtt{DCCA}$ security, unpredictability, and randomness-inextractability. Thus, a natural question would be whether we can prove the $\mathtt{CCA}$ security of the double-layered construction without using the non-malleability notions for both of the building blocks (and instead only requiring $\mathtt{DCCA}$ security). If such a security proof were possible, then one can use the bitwise-encrypt-based construction both for the inner KEM and the outer PKE scheme, and the resulting $\mathtt{CCA}$ secure KEM would be fairly simple. (If we appropriately use the known techniques and results on hybrid encryption, then the key and ciphertext sizes of the resulting KEM could be made only twice as large as those of the bitwise-encrypt-based KEM construction that works for the $\mathtt{CPA}$ and non-adaptive $\mathtt{CCA}$ (CCA1) settings.)

Unfortunately, however, we show that such a security proof is impossible, as there is a counterexample.

Theorem 3. *Assume there exists a detectable PKE scheme which is* $\mathtt{DCCA}$ *secure and unpredictable. Then, there exist a detectable KEM* $\mathit{\Gamma}_{\mathtt{in}}$ *and a detectable PKE scheme* $\mathit{\Pi}_{\mathtt{out}}$ *such that the following simultaneously hold: (1)* $\mathit{\Gamma}_{\mathtt{in}}$ *is* $\mathtt{DCCA}$ *secure and unpredictable. (2)* $\mathit{\Pi}_{\mathtt{out}}$ *is* $\mathtt{DCCA}$ *secure and randomness-inextractable. (3) The double-layered KEM* $\mathit{\Gamma}_{\mathtt{DL}}$ *constructed using* $\mathit{\Gamma}_{\mathtt{in}}$ *as the inner KEM and* $\mathit{\Pi}_{\mathtt{out}}$ *as the outer PKE scheme, is not* $\mathtt{CCA}$ *secure (in fact, not secure in the sense of one-wayness under* 1*-bounded* $\mathtt{CCA}$*).*

The formal proof is given in Appendix D.4. Our counterexample is based on an observation that the combination of DCCA security, unpredictability, and randomness-inextractability, does not rule out a double-layered KEM with the following property: A ciphertext C is of the form $C = (c_1, c_2)$ and the corresponding session-key K is of the form $K = (K_1, K_2)$, and furthermore it is "blockwise" consistent, meaning that each pair (c_i, K_i) is individually consistent as a ciphertext/session-key pair of the double-layered construction. Thus, the decapsulation result of the "swapped" ciphertext $\widehat{C} = (c_2, c_1)$ is the "swapped" session-key $\widehat{K} = (K_2, K_1)$. Such a KEM is clearly malleable, and its one-wayness is broken by just a single decapsulation query. For the details, see Appendix D.4.

5 Concrete Instantiations of Building Blocks

In this section, we show how to construct a detectable PKE scheme, which we call "encode-then-bitwise-encrypt" (EtBE) construction, that uses a CCA secure 1-bit PKE scheme and a $\mathcal{Q}$-non-malleable code as building blocks and simultaneously satisfies wRNM-DCCA security and randomness-inextractability. Since it is much easier to understand it if we first review the simple "bitwise-encrypt" construction, we first review it in Section 5.1 together with its security properties, and then we show the EtBE construction in Section 5.2.

5.1 Bitwise-Encrypt Construction

Here, we show that the simple "bitwise-encrypt" construction of a detectable PKE scheme based on a 1-bit PKE scheme, in which each bit of a plaintext is encrypted in a bit-by-bit fashion by the underlying 1-bit scheme, can be shown to satisfy randomness-inextractability, DCCA security, and unpredictability, if the underlying 1-bit PKE scheme is CCA secure.

Let $\Pi_1 = (\mathsf{PKG}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ be a 1-bit PKE scheme, and the randomness space of whose encryption algorithm Enc_1 is $\{0,1\}^\ell$ (where $\ell = \ell(k)$ is some positive polynomial). Then, for a polynomial $n = n(k) > 0$, consider the "bitwise-encrypt" construction $\Pi^n_{\mathtt{BE}} = (\mathsf{PKG}^n_{\mathtt{BE}} := \mathsf{PKG}_1, \mathsf{Enc}^n_{\mathtt{BE}}, \mathsf{Dec}^n_{\mathtt{BE}}, \mathsf{F}^n_{\mathtt{BE}})$ of an n-bit detectable PKE scheme described in Fig. 5 (left). The key generation algorithm $\mathsf{PKG}^n_{\mathtt{BE}}$ is actually PKG_1 itself, and we do not show it in the figure. The randomness space of $\mathsf{Enc}_{\mathtt{BE}}$ is $\{0,1\}^{\ell \cdot n}$. In the figure, we make the randomness used by $\mathsf{Enc}^n_{\mathtt{BE}}$ explicit so that it is convenient to consider its randomness-inextractability.

The following result was shown by Hohenberger et al. [16]:

Lemma 6. *([16]) Let $n = n(k) > 0$ be a polynomial. If the 1-bit PKE scheme Π_1 is* CCA *secure, then the detectable PKE scheme $\Pi^n_{\mathtt{BE}}$ scheme satisfies* DCCA *security and unpredictability.*

We show a similar statement regarding randomness-inextractability.

Lemma 7. *Let $n = n(k) > 0$ be a polynomial. If the PKE scheme Π_1 is* CCA *secure, then the detectable PKE scheme $\Pi^n_{\mathtt{BE}}$ satisfies randomness-inextractability.*

The formal proof is given in Appendix D.5, and here we explain an intuition why the lemma is true, which is quite straightforward: Suppose an adversary $\mathcal{A}$, given a public key pk and the challenge ciphertext $C^* = (c^*_1, \dots, c^*_n)$ and access to the decryption oracle, succeeds in outputting a plaintext $m' = (m'_1 \| \dots \| m'_n)$ and a randomness $r' = (r'_1, \dots, r'_n)$ such that $\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C') = 1$ with $C' = (c'_1, \dots, c'_n) = \mathsf{Enc}^n_{\mathtt{BE}}(pk, m'; r')$. Then, by definition, there must be a position $i \in [n]$ such that $c^*_i = c'_j$ holds for some $j \in [n]$, where $c'_a = \mathsf{Enc}_1(pk, m'_a; r'_a)$ for each $a \in [n]$. Note that such $\mathcal{A}$ is in fact "extracting" the randomness used for generating c^*_i. Note also that extracting

```
$\mathsf{Enc}^n_{\mathtt{BE}}(pk, m; r)$ :
    Parse $r$ as $(r_1, \ldots, r_n) \in (\{0,1\}^{\ell})^n$.
    View $m$ as $(m_1 \| \ldots \| m_n) \in \{0,1\}^n$.
    $\forall i \in [n] : c_i \leftarrow \mathsf{Enc}_1(pk, m_i; r_i)$
    Return $C \leftarrow (c_1, \ldots, c_n)$.
$\mathsf{Dec}^n_{\mathtt{BE}}(sk, C)$ :
    $(c_1, \ldots, c_n) \leftarrow C$
    $\forall i \in [n] : m_i \leftarrow \mathsf{Dec}_1(sk, c_i)$
    If $\exists i \in [n] : m_i = \bot$ then return $\bot$.
    Return $m \leftarrow (m_1 \| \ldots \| m_n)$.
$\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C')$ :
    $(c^*_1, \ldots, c^*_n) \leftarrow C^*$;   $(c'_i, \ldots, c'_n) \leftarrow C'$
    If $\exists i, j \in [n] : c^*_i = c'_j$
                then return 1 else return 0.
```

```
$\mathsf{Enc}_{\mathtt{EtBE}}(pk, m; R)$ :
    Parse $R$ as $(r, \widehat{r}) \in \{0,1\}^{\ell \cdot n} \times \{0,1\}^{\widehat{\ell}}$.
    $s = (s_1 \| \ldots \| s_n) \leftarrow \mathsf{E}(1^k, m; \widehat{r})$
    $C = (c_1, \ldots, c_n) \leftarrow \mathsf{Enc}^n_{\mathtt{BE}}(pk, s; r)$
    If $\mathtt{DUPCHK}(C) = 0$ then return $C$ else return $\bot$.$^{\dagger}$
$\mathsf{Dec}_{\mathtt{EtBE}}(sk, C)$ :
    If $\mathtt{DUPCHK}(C) = 1$ then return $\bot$.
    $s \leftarrow \mathsf{Dec}^n_{\mathtt{BE}}(sk, C)$
    If $s = \bot$ then return $\bot$.
    Return $m \leftarrow \mathsf{D}(1^k, s)$.
$\mathsf{F}_{\mathtt{EtBE}}(pk, C^*, C')$ :
    If (a) $\wedge$ (b) then return return 1 else return 0:
      (a) $\mathtt{DUPCHK}(C^*) = \mathtt{DUPCHK}(C') = 0$
      (b) $\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C') = 1$
```

Fig. 5. The "bitwise-encrypt" (n-bit) construction $\Pi^n_{\mathtt{BE}}$ (left), and the "encode-then-bitwise-encrypt" (EtBE) construction $\Pi_{\mathtt{EtBE}}$ (right), both based on a 1-bit PKE scheme Π_1. The key generation algorithms for $\Pi^n_{\mathtt{BE}}$ and $\Pi_{\mathtt{EtBE}}$ are the key generation algorithm PKG_1 of the underlying scheme Π_1. † Regarding the case in which $\mathsf{Enc}_{\mathtt{EtBE}}$ returns $\bot$, see the explanation in the text.

a randomness used for generating a ciphertext is a harder task than breaking indistinguishability. Thus, it is easy to construct another CCA adversary (a reduction algorithm) $\mathcal{B}$ for Π_1 that initially guesses the position i such that $c^*_i = c'_j$ holds with some j, embeds $\mathcal{B}$'s challenge ciphertext into the i-th position of the challenge ciphertext for $\mathcal{A}$, and has the CCA advantage at least $1/n$ times that of $\mathcal{A}$'s advantage in breaking randomness-inextractability.

5.2 Encode-then-Bitwise-Encrypt Construction

Here, we show the construction of detectable PKE that we call *"Encode-then-Bitwise-Encrypt"* (EtBE), which simultaneously achieves wRNM-DCCA security and randomness-inextractability, based on the security properties of the bitwise-encrypt construction (which are in turn based on the underlying CCA secure 1-bit scheme) and a $\mathcal{Q}$-non-malleable code. Our construction is actually a direct "PKE"-analogue of the transformation of a CCA secure 1-bit commitment scheme into a non-malleable string commitment scheme by Agrawal et al. [1]. We adapt their construction into the (detectable) PKE setting.

Let $\mathcal{C} = (\mathsf{E}, \mathsf{D})$ be a code with message length k and codeword length $n = n(k) \geq k$. Let $\Pi_1 = (\mathsf{PKG}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ be a 1-bit PKE scheme. Let $\Pi^n_{\mathtt{BE}} = (\mathsf{PKG}^n_{\mathtt{BE}} = \mathsf{PKG}_1, \mathsf{Enc}^n_{\mathtt{BE}}, \mathsf{Dec}^n_{\mathtt{BE}}, \mathsf{F}^n_{\mathtt{BE}})$ be the bitwise-encrypt construction based on Π_1. For convenience, we introduce the procedure "$\mathtt{DUPCHK}(\cdot)$" which takes a ciphertext $C = (c_1, \ldots, c_n)$ of $\Pi^n_{\mathtt{BE}}$ as input, and returns 1 if there exist distinct $i, j \in [n]$ such that $c_i = c_j$, and returns 0 otherwise. (That is, $\mathtt{DUPCHK}(C)$ checks a duplication in the component ciphertexts $(c_i)_{i \in [n]}$.)

Using $\mathcal{C}$, $\Pi^n_{\mathtt{BE}}$ (and Π_1), and $\mathtt{DUPCHK}$, the EtBE construction $\Pi_{\mathtt{EtBE}} = (\mathsf{PKG}_{\mathtt{EtBE}} := \mathsf{PKG}_1, \mathsf{Enc}_{\mathtt{EtBE}}, \mathsf{Dec}_{\mathtt{EtBE}}, \mathsf{F}_{\mathtt{EtBE}})$ is constructed as in Fig. 5 (right). Like $\Pi^n_{\mathtt{BE}}$, the key generation algorithm $\mathsf{PKG}_{\mathtt{EtBE}}$ is PKG_1 itself, and we do not show it in the figure. The plaintext space of $\Pi_{\mathtt{EtBE}}$ is $\{0,1\}^k$.

On the Correctness of $\Pi_{\mathtt{EtBE}}$. Note that the encryption algorithm $\mathsf{Enc}_{\mathtt{EtBE}}$ returns $\bot$ if it happens to be the case that $\mathtt{DUPCHK}(C) = 1$. This check is to ensure that a valid ciphertext does not have "duplicated" components, which is required due to our use of a $\mathcal{Q}$-non-malleable code whose non-malleability can only take care of a “non-duplicated” quoting. Since the probability (over the randomness of $\mathsf{Enc}_{\mathtt{EtBE}}$) that $\mathsf{Enc}_{\mathtt{EtBE}}$ outputs $\bot$ is not zero, our construction $\mathit{\Pi}_{\mathtt{EtBE}}$ does not satisfy correctness in a strict sense. (The exactly same problem arises in the construction of string commitments in [1].) However, it is easy to show that if $\mathit{\Pi}_1$ satisfies $\mathtt{CCA}$ security (or even $\mathtt{CPA}$ security), the probability of $\mathsf{Enc}_{\mathtt{EtBE}}$ outputting $\bot$ is negligible, and thus it does not do any harm in practice. (In practice, for example, in case $\bot$ is output, one can re-execute $\mathsf{Enc}_{\mathtt{EtBE}}$ with a fresh randomness. The expected execution time of $\mathsf{Enc}_{\mathtt{EtBE}}$ is negligibly close to 1.) Furthermore, if one needs standard correctness, then instead of letting $\mathsf{Enc}_{\mathtt{EtBE}}$ output $\bot$ in case $\mathtt{DUPCHK}(C) = 1$, one can let it output a plaintext m (being encrypted) as an “irregular ciphertext”, so that if the decryption algorithm $\mathsf{Dec}_{\mathtt{EtBE}}$ takes an irregular ciphertext C as input, it outputs C as a “decryption result” of C. (In order to actually implement this, in case $\mathtt{DUPCHK}(C) = 1$ occurs, $m \in \{0,1\}^k$ needs to be padded to the length $n \cdot |c|$ of an ordinary ciphertext, and we furthermore need to put a prefix for every ciphertext that tells the decryption algorithm whether the received ciphertext should be treated as a normal ciphertext or an irregular one.) Such a modification also does no harm to the security properties of $\mathit{\Pi}_{\mathtt{EtBE}}$ (it only contributes to increasing an adversary’s advantage negligibly), thanks to the $\mathtt{CCA}$ security of the building block $\mathit{\Pi}_1$. For simplicity, in this paper we focus on the current construction of $\mathit{\Pi}_{\mathtt{EtBE}}$.

Security of $\mathit{\Pi}_{\mathtt{EtBE}}$. The security properties of the EtBE construction is guaranteed by the following lemmas.

Lemma 8. *Assume that $\mathit{\Pi}_1$ is $\mathtt{CCA}$ secure and $\mathcal{C}$ is a $\mathcal{Q}$-non-malleable code. Then, the detectable PKE scheme $\mathit{\Pi}_{\mathtt{EtBE}}$ in Fig. 5 (right) is $\mathtt{wRNM\text{-}DCCA}$ secure.*

Lemma 9. *If $\mathit{\Pi}_1$ is $\mathtt{CCA}$ secure, then the detectable PKE scheme $\mathit{\Pi}_{\mathtt{EtBE}}$ scheme in Fig. 5 (right) satisfies unpredictability and randomness-inextractabilty.*

The proof of Lemma 8 is given in Appendix D.6. The proof of Lemma 9 is straightforward given the unpredictability (Lemma 6) and randomness-inextractability (Lemma 7) of the bitwise-encrypt construction $\mathit{\Pi}^n_{\mathtt{BE}}$, and thus omitted.

The proof of Lemma 8 follows essentially the same story line as the security proof of the non-malleable string commitment by Agrawal et al. [1]. A high-level idea is as follows: In the $\mathtt{wRNM\text{-}DCCA}$ experiment, an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ is allowed to submit a single “unrestricted” decryption query $C' = (c'_1, \dots c'_n)$, which is captured by the ciphertext finally output by $\mathcal{A}_2$. In order for this query to be valid, however, C' has to satisfy $\mathtt{DUPCHK}(C') = 0$, which guarantees that C' does not have duplicated components. Thus, since each component is a ciphertext of the $\mathtt{CCA}$ secure scheme $\mathit{\Pi}_1$, the best $\mathcal{A}$ can do to generate C' that is “related” to the challenge ciphertext $C^* = (c^*_1, \dots, c^*_n)$ is to “quote” some of c^*_i’s into C' in such a way that no c^*_i appears more than once. However, such “quoting without duplicated positions” is exactly the function class $\mathcal{Q}$ with respect to which the code $\mathcal{C}$ is non-malleable. Specifically, the $\mathcal{Q}$-non-malleability of $\mathcal{C}$ guarantees that even if an adversary observes the decryption result of such C' that quotes some of components of C^* without duplicated positions, $\mathcal{A}$ gains essentially no information of the original content m_b of the encoding s^* encrypted in C^*, and hence no information of the challenge bit b. Actually, it might be the case that $\mathcal{A}$ succeeds in generating C' so that $\mathsf{Dec}^n_{\mathtt{BE}}(sk, C')$ is s^* itself (and hence its decoded value is exactly the challenge plaintext m_b). According to the rule of the $\mathtt{wRNM\text{-}DCCA}$ experiment, however, in such a case $\mathcal{A}$ is not given the actual decryption result $\mathsf{Dec}_{\mathtt{EtBE}}(sk, C')$ directly but is given the symbol same which only informs that the decryption result is either m_0 or m_1. Furthermore, all other queries without quoting do not leak the information of the challenge bit b because of the $\mathtt{DCCA}$ security of the bitwise-encrypt construction $\Pi^n_{\mathtt{BE}}$ (Lemma 6). These ideas lead to $\mathtt{wRNM\text{-}DCCA}$ security of $\Pi_{\mathtt{EtBE}}$. For the details, see Appendix D.6.

6 Full Description of Our 1-bit-to-Multi-bit Conversion

Given the results in the previous sections, we are now ready to describe our proposed 1-bit-to-multi-bit conversion, i.e. a $\mathtt{CCA}$ secure KEM from a $\mathtt{CCA}$ secure 1-bit PKE scheme. Let $\Pi_1 = (\mathsf{PKG}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ be a 1-bit PKE scheme whose public key size is "$|pk|$", the ciphertext size is "$|c|$", and the randomness space of whose encryption algorithm Enc_1 is $\{0,1\}^{\ell}$. Let $\mathcal{C} = (\mathsf{E}, \mathsf{D})$ be a $\mathcal{Q}$-non-malleable (n,k)-code with $n = n(k) \geq k$, and the randomness space of whose encoding algorithm E is $\{0,1\}^{\widehat{\ell}}$. Let $\ell' = n \cdot \ell + \widehat{\ell} + 2k$, and $\mathsf{G} : \{0,1\}^k \to \{0,1\}^{\ell'}$ be a PRG. Finally, let $E = (\mathsf{SEnc}, \mathsf{SDec})$ be a deterministic SKE scheme whose plaintext space is $\{0,1\}^{k \cdot |c|}$, and it has zero ciphertext overhead (i.e. its ciphertext size is the same as that of a plaintext).

From these building blocks, consider the following detectable KEM $\Gamma_{\mathtt{in}}$ and detectable PKE scheme $\Pi_{\mathtt{out}}$:

$\Gamma_{\mathtt{in}}$**:** Consider the bitwise-encrypt construction $\Pi^k_{\mathtt{BE}}$ (Fig. 5) based on the PKE scheme Π_1, and regard it as a detectable KEM by encrypting a random k-bit string as a session-key. For this detectable KEM, use the PRG G with the method explained in the first paragraph of Section 3.3 to stretch its session-key into ℓ' bits. $\Gamma_{\mathtt{in}}$ is the resultant KEM.
The public key size of $\Gamma_{\mathtt{in}}$ is $|pk|$, its ciphertext size is $k \cdot |c|$, and its session-key space is $\{0,1\}^{\ell'}$. Due to Lemmas 3 and 6, $\Gamma_{\mathtt{in}}$ satisfies $\mathtt{DCCA}$ security and unpredictability based on the $\mathtt{CCA}$ security of Π_1 and the security of G.

$\Pi_{\mathtt{out}}$**:** Consider the EtBE construction $\Pi_{\mathtt{EtBE}}$ based on the code $\mathcal{C}$ and the bitwise-encrypt construction $\Pi^n_{\mathtt{BE}}$ (which is in turn based on Π_1) (Fig. 5). Combine this detectable PKE scheme with the SKE scheme E by the method explained in the second paragraph of Section 3.3 (see Fig. 3). $\Pi_{\mathtt{out}}$ is the resultant PKE scheme.
The public key size of $\Pi_{\mathtt{out}}$ is $|pk|$, its ciphertext overhead (the difference between the total ciphertext size minus the plaintext size) is $n \cdot |c|$, its plaintext space is $\{0,1\}^{k \cdot |c|}$, and the randomness space of its encryption algorithm is $\{0,1\}^{\ell' - k}$. Due to Lemmas 4, 6, 7, 8, and 9, $\Pi_{\mathtt{out}}$ satisfies $\mathtt{wRNM\text{-}DCCA}$ security and randomness-inextractability, based on the $\mathtt{CCA}$ security of Π_1, $\mathcal{Q}$-non-malleability of $\mathcal{C}$, and the $\mathtt{CCA}$ security of E.

Our proposed KEM $\widetilde{\Gamma} = (\widetilde{\mathsf{KKG}}, \widetilde{\mathsf{Encap}}, \widetilde{\mathsf{Decap}})$ is then obtained from the double-layered construction $\Gamma_{\mathtt{DL}}$ in which the inner KEM is $\Gamma_{\mathtt{in}}$ and the outer PKE scheme is $\Pi_{\mathtt{out}}$ explained above. More concretely, the description of $\widetilde{\Gamma}$ is as in Fig. 6.

The public key size of $\widetilde{\Gamma}$ is $2 \cdot |pk|$, and its ciphertext size is $(n+k) \cdot |c|$ (where $\Gamma_{\mathtt{in}}$ contributes $k \cdot |c|$ and $\Pi_{\mathtt{out}}$ contributes $n \cdot |c|$). Using the $\mathcal{P}$-non-malleable code with "optimal rate" (Lemma 1) by Agrawal et al. [2] which also satisfies $\mathcal{Q}$-non-malleability by Lemma 2, we have $n = k + o(k)$. Thus, the ciphertext size of $\widetilde{\Gamma}$ can be made asymptotically $(2k + o(k)) \cdot |c|$.

The following statement is obtained as a corollary of the combination of Theorem 1 and Lemmas 1, 2, 3, 4, 6, 7, 8, and 9.

Theorem 4. *Assume that the PKE scheme Π_1 is $\mathtt{CCA}$ secure, $\mathcal{C}$ is a $\mathcal{Q}$-non-malleable code, G is a PRG, and the SKE scheme E is $\mathtt{CCA}$ secure. Then, the KEM $\widetilde{\Gamma}$ in Fig. 6 is $\mathtt{CCA}$ secure.*

```
$\widetilde{\mathsf{KKG}}(1^k)$ :
    $(pk_{\mathtt{in}}, sk_{\mathtt{in}}) \leftarrow \mathsf{PKG}_1(1^k)$
    $(pk_{\mathsf{out}}, sk_{\mathsf{out}}) \leftarrow \mathsf{PKG}_1(1^k)$
    $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathsf{out}})$
    $SK \leftarrow (sk_{\mathtt{in}}, sk_{\mathsf{out}}, PK)$
    Return $(PK, SK)$.
```

```
$\widetilde{\mathsf{Encap}}(PK)$ :
    $(pk_{\mathtt{in}}, pk_{\mathsf{out}}) \leftarrow PK$
    $K_{\mathtt{in}} = (K_{\mathtt{in}}^{(1)} \| \dots \| K_{\mathtt{in}}^{(k)}) \leftarrow \{0,1\}^k$
    $\forall i \in [k] : c_{\mathtt{in}}^{(i)} \leftarrow \mathsf{Enc}_1(pk_{\mathtt{in}}, K_{\mathtt{in}}^{(i)})$
    $\alpha \leftarrow \mathsf{G}(K_{\mathtt{in}})$
    Parse $\alpha$ as $(r_1, \dots, r_n, \widehat{r}, K_{\mathsf{out}}, K)$
            $\in (\{0,1\}^{\ell})^n \times \{0,1\}^{\widehat{\ell}} \times (\{0,1\}^k)^2$.
    $s = (s_1 \| \dots \| s_n) \leftarrow \mathsf{E}(1^k, K_{\mathsf{out}}; \widehat{r})$
    $\forall i \in [n] : c_i \leftarrow \mathsf{Enc}_1(pk_{\mathsf{out}}, s_i; r_i)$
    If $\mathtt{DUPCHK}((c_i)_{i \in [n]}) = 1$ then return $\bot$.
    $\widehat{c} \leftarrow \mathsf{SEnc}(K_{\mathsf{out}}, (c_{\mathtt{in}}^{(1)} \| \dots \| c_{\mathtt{in}}^{(k)}))$
    $C \leftarrow (c_1, \dots, c_n, \widehat{c})$
    Return $(C, K)$.
```

```
$\widetilde{\mathsf{Decap}}(SK, C)$ :
    $(sk_{\mathtt{in}}, sk_{\mathsf{out}}, PK) \leftarrow SK$
    $(pk_{\mathtt{in}}, pk_{\mathsf{out}}) \leftarrow PK$;  $(c_1, \dots, c_n, \widehat{c}) \leftarrow C$
    If $\mathtt{DUPCHK}((c_i)_{i \in [n]}) = 1$ then return $\bot$.
    $\forall i \in [n] : s_i \leftarrow \mathsf{Dec}_1(sk_{\mathsf{out}}, c_i)$
    If $\exists i \in [n] : s_i = \bot$ then return $\bot$.
    $K_{\mathsf{out}} \leftarrow \mathsf{D}(1^k, s = (s_1 \| \dots \| s_n))$
    If $K_{\mathsf{out}} = \bot$ then return $\bot$.
    $(c_{\mathtt{in}}^{(1)} \| \dots \| c_{\mathtt{in}}^{(k)}) \leftarrow \mathsf{SDec}(K_{\mathsf{out}}, \widehat{c})$
    If $\mathsf{SDec}$ has returned $\bot$ then return $\bot$.
    $\forall i \in [k] : K_{\mathtt{in}}^{(i)} \leftarrow \mathsf{Dec}_1(sk_{\mathtt{in}}, c_{\mathtt{in}}^{(i)})$
    If $\exists i \in [k] : K_{\mathtt{in}}^{(i)} = \bot$ then return $\bot$.
    $\alpha \leftarrow \mathsf{G}(K_{\mathtt{in}} = (K_{\mathtt{in}}^{(1)} \| \dots \| K_{\mathtt{in}}^{(k)}))$
    Parse $\alpha$ as $(r_1, \dots, r_n, \widehat{r}, K'_{\mathsf{out}}, K)$
            $\in (\{0,1\}^{\ell})^n \times \{0,1\}^{\widehat{\ell}} \times (\{0,1\}^k)^2$.
    If (a) $\wedge$ (b) $\wedge$ (c) then return $K$ else return $\bot$:
      (a) $\forall i \in [n] : \mathsf{Enc}_1(pk_{\mathsf{out}}, s_i; r_i) = c_i$
      (b) $\mathsf{E}(1^k, K'_{\mathsf{out}}; \widehat{r}) = s$
      (c) $\mathsf{SEnc}(K'_{\mathsf{out}}, (c_{\mathtt{in}}^{(1)} \| \dots \| c_{\mathtt{in}}^{(k)})) = \widehat{c}$
```

Fig. 6. The proposed "1-bit-to-multi-bit" construction (KEM) $\widetilde{\Gamma}$.

2-bit-to-multi-bit Construction with a Single Key Pair. Note that our proposed 1-bit-to-multi-bit conversion $\widetilde{\Gamma}$ uses two key pairs of the underlying 1-bit scheme Π_1. It turns out that if we can use a 2-bit PKE scheme as a building block instead of a 1-bit scheme, then we can construct a CCA secure KEM that uses only one key pair of the underlying 2-bit scheme, with a very similar way to $\widetilde{\Gamma}$. We detail it in Appendix C.

The idea of this 2-bit-to-multi-bit conversion is to use the additional 1-bit of the plaintext space as the "indicator bit" that indicates whether each component ciphertext is generated for the inner layer or the outer layer. That is, each inner ciphertext $c_{\mathtt{in}}^{(i)}$ is an encryption of $(1 \| K_{\mathtt{in}}^{(i)})$, and each outer ciphertext c_i is an encryption of $(0 \| s_i)$, and in the decapsulation algorithm, we check whether the component ciphertexts $\{c_i\}_{i \in [n]}$ and $\{c_{\mathtt{in}}^{(i)}\}_{i \in [k]}$ have appropriate indicator bits ("1" for the inner layer and "0" for the outer layer). This additional indicator bit and its check prevent a quoting of an inner ciphertext into the outer layer and vice versa, and thus make the encryption/decryption operations for the inner layer and those of the outer layer virtually independent, as if each layer has an individual key pair. This enables us to conduct the security proof in essentially the same way as that of $\widetilde{\Gamma}$. For the details, see Appendix C.

7 Comparison

Table 1 compares the public key size and ciphertext size of the existing "1-bit-to-multi-bit" constructions that achieve CCA security (or related security). Specifically, in the table, "MS" represents the construction by Myers and Shelat [22].; "HLW" represents the construction by Hohenberger et al. [16] which uses a CPA secure PKE scheme, a 1-bounded CCA secure [7] PKE scheme, and a detectable PKE scheme satisfying DCCA security and unpredictability. We assume that for the 1-bounded CCA secure scheme, the construction by Dodis and Fiore [10, Appendix C] is used, which constructs such a scheme from a CPA secure scheme and a one-time signature scheme, and we also assume that its detectable scheme and the CPA secure scheme are realized by the bitwise-encrypt

Table 1. Comparison among the 1-bit-to-multi-bit constructions for CCA (and related) security. In the columns "PK Size" and "Ciphertext Size", $|pk|$ and $|c|$ denote the public key size and the ciphertext size of the underlying 1-bit PKE scheme $\mathit{\Pi}_1$, respectively, and $|vk|$ and $|\sigma|$ denote the size of a verification key and that of a signature of the one-time signature scheme used as a building block, respectively. The column "Sec. on $\mathit{\Pi}_1$" shows the assumption on the security of the underlying 1-bit PKE scheme required to show the CCA (or the related) security of the entire construction. Here, "SDA" and "NM-SDA" denote "(indistinguishability against) self-destruct CCA" [6] and "non-malleability against SDA" [5], respectively. The column "Add. Bld. Blk." shows the additional building blocks (used in each construction) that can be realized only from the existence of a one-way function. Here, "Sig" stands for a one-time signature scheme. (†) As explained in Introduction, CMTV [6] and CTDV [5] only achieve SDA security and NM-SDA security, respectively, which are both implied by ordinary CCA security but are strictly weaker than it.

Scheme	PK Size	Ciphertext Size	Sec. of $\mathit{\Pi}_1$	Add. Bld. Blk.
MS [22]	$(20k^2+1)\lvert pk\rvert$	$(10k^3\lvert c\rvert+\lvert vk\rvert+\lvert\sigma\rvert)\lvert c\rvert$	CCA	Sig., PRG
HLW [16]	$(2k+2)\lvert pk\rvert$	$(k^2+3k)\lvert c\rvert+\lvert vk\rvert+\lvert\sigma\rvert+6k$	DCCA & UNP	Sig., PRG, SKE
MH [21]	$(2k+2)\lvert pk\rvert$	$(k^2+2k)\lvert c\rvert+\lvert vk\rvert+\lvert\sigma\rvert$	DCCA & UNP	Sig., PRG, SKE
CMTV† [6]	$\approx k\lvert pk\rvert$	$\approx 5k\lvert c\rvert$	SDA	—
CDTV† [5]	$O(k)\lvert pk\rvert$	$O(k)\lvert c\rvert$	NM-SDA	—
Ours (§ 6)	$2\lvert pk\rvert$	$(2k+o(k))\lvert c\rvert$	CCA	PRG, SKE

construction $\mathit{\Pi}^k_{\mathtt{BE}}$. (If we need to encrypt a value longer than k-bit, then we assume that hybrid encryption is used everywhere possible by encrypting a k-bit random session-key and using it as a key for SKE (where the length of SKE ciphertexts are assumed to be the same as a plaintext [25]), which we do the same for the constructions explained below.); "MH" represents the construction by Matsuda and Hanaoka [21], which can be seen as an efficient version of HLW [16] due to hybrid encryption techniques, and we assume that the building blocks similar to HLW are used.; "CMTV" represents the construction by Coretti et al. [6], the size parameters of which are taken from the introduction of [6].; "CDTV" represents the construction by Coretti et al. [5], where the size parameters are estimated according to the explanations in [5, Sections 4.2 & 4.3].; "Ours" is the KEM $\widetilde{\mathit{\Gamma}}$ shown in Fig. 6 in Section 6.

As is clear from Table 1, if one starts from a CCA secure 1-bit PKE scheme (and assuming that building blocks implied by one-way functions are available for free), then "Ours" achieves asymptotically the best efficiency. Notably, the public size and the ciphertext size of "Ours" are asymptotically "optimal" in the sense that they are asymptotically the same as the bitwise-encrypt construction $\mathit{\Pi}^k_{\mathtt{BE}}$ that works as a 1-bit-to-multi-bit conversion for the CPA and non-adaptive CCA (CCA1) settings. Note also that all the previous constructions that achieve ordinary CCA security have the public key size $\mathit{\Omega}(k)\cdot|pk|$, and the ciphertext size $\mathit{\Omega}(k^2)\cdot|c|$.

We note that, as mentioned in Section 1.3, CMTV [6] and CDTV [5] achieve only indistinguishability under self-destruct CCA (SDA) and non-malleability under self-destruct CCA (NM-SDA), respectively, which are both implied by ordinary CCA security but are strictly weaker than it. Nonetheless, "Ours" actually achieves better asymptotic efficiency than them.

However, for fairness we note that our construction requires CCA security for the underlying 1-bit PKE scheme $\mathit{\Pi}_1$, while HLW [16] and MH [21] only require DCCA security and unpredictability, and the constructions CMTV [6] and CDTV [5] only require SDA and NM-SDA security for $\mathit{\Pi}_1$, respectively, and thus there is a tradeoff among the assumptions on the building block $\mathit{\Pi}_1$.

8 On the Necessity of Two Key Pairs

Our positive results on the 1-/2-bit-to-multi-bit constructions for CCA security raise an interesting question in terms of the number of public keys: *Is it necessary to use two key pairs in* 1*-bit-to-*

multi-bit constructions for $\mathtt{CCA}$ *security?* Motivated by this question, in this section we consider the one-key variant of our proposed KEM $\widetilde{\Gamma}$, and study its security. Specifically, consider a variant of the KEM $\widetilde{\Gamma}$ in Fig. 6 in which the key generation algorithm only generates one key pair (pk, sk) and then pk is used for both $pk_{\mathtt{in}}$ and $pk_{\mathtt{out}}$, and sk is used for both $sk_{\mathtt{in}}$ and $sk_{\mathtt{out}}$. Let $\widetilde{\Gamma}' = (\widetilde{\mathsf{KKG}}', \widetilde{\mathsf{Encap}}', \widetilde{\mathsf{Decap}}')$ be this "one-key-pair" version of the KEM $\widetilde{\Gamma}$.

Unfortunately, we will show that this KEM $\widetilde{\Gamma}'$ is vulnerable to a $\mathtt{CCA}$ attack. Hence, our negative result in this section shows that using two key pairs of the underlying 1-bit scheme is essential for our proposed construction $\widetilde{\Gamma}$. This also contrasts strikingly with our 2-bit-to-multi-bit construction explained in Section 6 (and Appendix C) that only requires one key pair. Although the negative result on the one-key pair variant does not rule out the possibility of 1-bit-to-multi-bit constructions that use only one-key pair in general, it does show a necessity of different techniques and ideas for fully answering the optimal number of keys needed for 1-bit-to-multi-bit conversion for $\mathtt{CCA}$ security. We leave it as an open problem to clarify whether we can achieve a 1-bit-to-multi-bit conversion for $\mathtt{CCA}$ security that uses only one key pair.

Theorem 5. *If* G *is a PRG, then the KEM* $\widetilde{\Gamma}'$ *is not* $\mathtt{CCA}$ *secure.*

The formal proof is given in Appendix D.7. Here, we explain a proof overview. Our key observations for the $\mathtt{CCA}$ attack on $\widetilde{\Gamma}'$ are: (1) One can use the decapsulation oracle $\widetilde{\mathsf{Decap}}'(sk, \cdot)$ to decrypt any "honestly generated" ciphertext c' of the underlying 1-bit PKE scheme that is in the range of $\mathsf{Enc}_1(pk, \cdot)$, and (2) the "outer" ciphertexts $\{c_i^*\}_{i \in [n]}$ and the "inner" ciphertexts $\{c_{\mathtt{in}}^{*(i)}\}_{i \in [k]}$ in the challenge ciphertext $C^* = (c_1^*, \dots, c_n^*, \widehat{c}^* = \mathsf{SEnc}(K_{\mathtt{out}}, (c_{\mathtt{in}}^{*(1)} \| \dots \| c_{\mathtt{in}}^{*(k)})))$ are *all* generated honestly. These (1) and (2) imply that the following simple adversary $\mathcal{A}$ works: Given the challenge ciphertext C^*, $\mathcal{A}$ decrypts the outer ciphertexts $\{c_i^*\}_{i \in [n]}$ to recover s^* with the help of the decapsulation oracle $\widetilde{\mathsf{Decap}}'(sk, \cdot)$ and the above mentioned property (1), decodes s^* to obtain $K_{\mathtt{out}}^*$, decrypts $\widehat{c}^*$ using $K_{\mathtt{out}}^*$ to recover the inner ciphertexts $\{c_{\mathtt{in}}^{*(i)}\}_{i \in [k]}$, decrypts them to recover $K_{\mathtt{in}}^*$ with the help of the decapsulation oracle as above, and finally recovers the session-key $K^* = \mathsf{LSB}_k(\mathsf{G}(K_{\mathtt{in}}^*))$. (Here, $\mathsf{LSB}_k(\cdot)$ denotes the least significant k-bits of the input.)

Therefore, our task is reduced to showing how one uses the decapsulation oracle $\widetilde{\mathsf{Decap}}'(sk, \cdot)$ to decrypt honestly generated ciphertexts of Π_1. Let c' be an honestly generated ciphertext in the range of $\mathsf{Enc}_1(pk, \cdot)$, and let $K_{\mathtt{in}}^\star \in \{0,1\}^{k-1}$ be a string satisfying $\mathsf{LSB}_k(\mathsf{G}(1 \| K_{\mathtt{in}}^\star)) \neq \mathsf{LSB}_k(\mathsf{G}(0 \| K_{\mathtt{in}}^\star))$. Then, consider the procedure of generating a ciphertext/session-key pair (C, K) in the same manner as $\widetilde{\mathsf{Encap}}'(pk)$, except that c' is used as the first inner ciphertext $c_{\mathtt{in}}^{(1)}$, $K_{\mathtt{in}} = (1 \| K^\star)$ is used for computing α, and the rest is unchanged from $\widetilde{\mathsf{Encap}}'(pk)$. This is as if we generate a ciphertext C and a session-key K so that we treat c' as an encryption of 1 and embed it into the position of $c_{\mathtt{in}}^{(1)}$ in C. Indeed, we can show that if c' is an encryption of 1, then $\widetilde{\mathsf{Decap}}'(sk, C) = K$ holds, while if c' is an encryption of 0, then $\widetilde{\mathsf{Decap}}'(sk, C) = K$ never occurs, due to the property of $K_{\mathtt{in}}^\star$. Put differently, it holds that $\mathsf{Dec}_1(sk, c') = (\widetilde{\mathsf{Decap}}'(sk, C) \overset{?}{=} K)$. Thus, the decapsulation oracle $\widetilde{\mathsf{Decap}}'(sk, \cdot)$, together with the above method of "embedding" c', can be used to decrypt c'.

However, there is one problem: There could be a PRG G such that $\mathsf{LSB}_k(\mathsf{G}(1 \| s)) = \mathsf{LSB}_k(\mathsf{G}(0 \| s))$ holds for all $s \in \{0,1\}^{k-1}$ (e.g. a PRG that ignores the first bit). If such G is used, we cannot mount the above attack. Thus, in the actual proof we use a more sophisticated (and more complicated) method for embedding c' into a ciphertext/session-key pair so that essentially the same method works for any PRG G. For more details, see Appendix D.7.

Acknowledgement

The authors would like to thank the members of the study group "Shin-Akarui-Angou-Benkyou-Kai," and the anonymous reviewers of ASIACRYPT 2015 for their helpful comments and suggestions.

References

1. S. Agrawal, D. Gupta, H.K. Maji, O. Pandey, and M. Prabhakaran. Explicit non-malleable codes against bit-wise tampering and permutations. In *Proc. of CRYPTO 2015(1)*, volume 9215 of *LNCS*, pages 538–557. Springer, 2015.
2. S. Agrawal, D. Gupta, H.K. Maji, O. Pandey, and M. Prabhakaran. A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In *Proc. of TCC 2015(1)*, volume 9014 of *LNCS*, pages 375–397. Springer, 2015.
3. 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.
4. 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.
5. S. Coretti, Y. Dodis, B. Tackmann, and D. Venturi. Non-malleable encryption: Simpler, shorter, stronger, 2015. To appear in TCC 2016-A. http://eprint.iacr.org/2015/772.
6. S. Coretti, U. Maurer, B. Tackmann, and D. Venturi. From single-bit to multi-bit public-key encryption via non-malleable codes. In *Proc. of TCC 2015(1)*, volume 9014 of *LNCS*, pages 532–560. Springer, 2015.
7. R. Cramer, G. Hanaoka, D. Hofheinz, H. Imai, E. Kiltz, R. Pass, A. Shelat, and V. Vaikuntanathan. Bounded CCA2-secure encryption. In *Proc. of ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 502–518. Springer, 2007.
8. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Computing*, 33(1):167–226, 2003.
9. D. Dachman-Soled, G. Fuchsbauer, P. Hohassel, and A. O'Neill. Enhanced chosen-ciphertext security and applications. In *Proc. of PKC 2014*, volume 8383 of *LNCS*, pages 329–344. Springer, 2014.
10. Y. Dodis and D. Fiore. Interactive encryption and message authentication, 2013. Full version of [11]. http://eprint.iacr.org/2013/817.
11. Y. Dodis and D. Fiore. Interactive encryption and message authentication. In *Proc. of SCN 2014*, volume 8642 of *LNCS*, pages 494–513. Springer, 2014.
12. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proc. of STOC 1991*, pages 542–552. ACM, 1991.
13. S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes, 2009. Full version of [14]. http://eprint.iacr.org/2009/608.
14. S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. In *Proc. of ICS 2010*, pages 434–452. Tsinghua University Press, 2010.
15. S. Faust, P. Mukherjee, J.B. Nielsen, and D. Venturi. Continuous non-malleable codes. In *Proc. of TCC 2014*, volume 8349 of *LNCS*, pages 465–488. Springer, 2014.
16. 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.
17. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *Proc. of TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, 2006.
18. F. Kitagawa, T. Matsuda, G. Hanaoka, and K. Tanaka. Completeness of single-bit projection-KDM security for public key encryption. In *Proc. of CT-RSA 2015*, volume 9048 of *LNCS*, pages 201–219. Springer, 2015.
19. H. Lin and S. Tessaro. Amplification of chosen-ciphertext security. In *Proc. of EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 503–519. Springer, 2013.
20. 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.
21. 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.
22. S. Myers and A. Shelat. Bit encryption is complete. In *Proc. of FOCS 2009*, pages 607–616. IEEE Computer Society, 2009.

23. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proc. of STOC 1990*, pages 427–437. ACM, 1990.
24. 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.
25. D.H. Phan and D. Pointcheval. About the security of ciphers (semantic security and pseudo-random permutations). In *Proc. of SAC 2004*, volume 3357 of *LNCS*, pages 182–197. Springer, 2005.

A Standard Cryptographic Primitives and a Basic Fact on Probability

A.1 Symmetric Key Encryption

A symmetric key encryption (SKE) scheme E consists of the two PPTAs $(\mathsf{SEnc}, \mathsf{SDec})$ with the following interface:

Encryption:	**Decryption:**
$c \leftarrow \mathsf{SEnc}(K, m)$	m (or $\bot$) $\leftarrow \mathsf{SDec}(K, c)$

where SDec is a deterministic algorithm, c is a ciphertext of a plaintext m under a key $K \in \{0,1\}^k$, and $k \in \mathbb{N}$ is a security parameter.

We say that a SKE scheme satisfies *correctness* if for all $k \in \mathbb{N}$, all $K \in \{0,1\}^k$, and all plaintexts m, it holds that $\mathsf{SDec}(K, \mathsf{SEnc}(K, m)) = m$.

$\mathtt{CCA}$ *Security.* Here, we review the $\mathtt{CCA}$ security of a SKE scheme. For a SKE scheme E and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, consider the $\mathtt{CCA}$ experiment $\mathsf{Expt}^{\mathtt{CCA}}_{E,\mathcal{A}}(k)$ that is defined as follows:

$$\mathsf{Expt}^{\mathtt{CCA}}_{E,\mathcal{A}}(k) : [\ (m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(1^k);\ K \leftarrow \{0,1\}^k;\ b \leftarrow \{0,1\};\ c^* \leftarrow \mathsf{SEnc}(K, m_b);$$
$$b' \leftarrow \mathcal{A}_2^{\mathsf{SDec}(K,\cdot)}(\mathsf{st}, c^*);\ \text{Return } (b' \stackrel{?}{=} b).\ ],$$

where it is required that $|m_0| = |m_1|$ holds, and $\mathcal{A}_2$ is not allowed to submit c^* to its decryption oracle. We say that a SKE scheme E is $\mathtt{CCA}$ secure if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{CCA}}_{E,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{CCA}}_{E,\mathcal{A}}(k) = 1] - 1/2|$ is negligible.

A.2 Message Authentication Code

A message authentication code (MAC) $\mathcal{M}$ consists of the two PPTAs $(\mathsf{Mac}, \mathsf{MVer})$ with the following interface:

MAC-Tag Generation:	**Verification:**
$\tau \leftarrow \mathsf{Mac}(K, m)$	$\top$ or $\bot \leftarrow \mathsf{MVer}(K, m, \tau)$

where MVer is a deterministic algorithm, τ is a MAC-tag of a message m under a key $K \in \{0,1\}^k$, $\top$ (resp. $\bot$) is the symbol indicating that τ is a valid (resp. invalid) MAC tag on m under K, and $k \in \mathbb{N}$ is a security parameter.

We say that a MAC satisfies *correctness* if for all $k \in \mathbb{N}$, all keys $K \in \{0,1\}^k$, and all messages m, it holds that $\mathsf{MVer}(K, m, \mathsf{Mac}(K, m)) = \top$.

Strong One-Time Security. Here, we recall the definition of strong one-time unforgeability (SOT security) of a MAC. For a MAC $\mathcal{M}$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, consider the SOT experiment $\mathsf{Expt}^{\mathtt{SOT}}_{\mathcal{M},\mathcal{A}}(k)$ that is defined as follows:

$$\begin{aligned}\mathsf{Expt}^{\mathtt{SOT}}_{\mathcal{M},\mathcal{A}}(k) : [\ &(m, \mathsf{st}) \leftarrow \mathcal{A}_1(1^k);\ K \leftarrow \{0,1\}^k;\ \tau \leftarrow \mathsf{Mac}(K, m);\ (m', \tau') \leftarrow \mathcal{A}_2(\mathsf{st}, \tau);\\ &\text{If } \mathsf{MVer}(K, m', \tau') = \top \wedge (m', \tau') \neq (m, \tau) \text{ then return 1 else return 0. }],\end{aligned}$$

We say that a MAC $\mathcal{M}$ is SOT secure if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{SOT}}_{\mathcal{M},\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{SOT}}_{\mathcal{M},\mathcal{A}}(k) = 1]$ is negligible.

A.3 Pseudorandom Generator

Let $\ell = \ell(k) > k$ be a polynomial and let $\mathsf{G} : \{0,1\}^* \rightarrow \{0,1\}^*$ be a function such that $|\mathsf{G}(x)| = \ell(|x|)$ holds for all strings $x \in \{0,1\}^*$. We say that G is a pseudorandom generator (PRG) if (1) it is efficiently computable, and (2) for all PPTAs $\mathcal{A}$, the following advantage function $\mathsf{Adv}^{\mathtt{PRG}}_{\mathsf{G},\mathcal{A}}(k)$ is negligible:

$$\mathsf{Adv}^{\mathtt{PRG}}_{\mathsf{G},\mathcal{A}}(K) := \Big| \Pr_{x \leftarrow \{0,1\}^k}[\mathcal{A}(1^k, \mathsf{G}(x)) = 1] - \Pr_{y \leftarrow \{0,1\}^\ell}[\mathcal{A}(1^k, y) = 1] \Big|.$$

A.4 Basic Fact on Probability

We will use the following simple fact on probability.

Fact 1 *Let* S *and* B *be events defined over the same probability space. Then, it holds that*

$$\Big|\Pr[\mathsf{S}] - \frac{1}{2}\Big| \leq \Big|\Pr[\mathsf{S} \wedge \overline{\mathsf{B}}] + \frac{1}{2}\Pr[\mathsf{B}] - \frac{1}{2}\Big| + \frac{1}{2}\Pr[\mathsf{B}]. \qquad (2)$$

Proof of Fact 1. If $\Pr[\mathsf{B}] = 0$, then the right hand side of Equation (2) is exactly $|\Pr[\mathsf{S}] - \frac{1}{2}|$, and thus the inequality trivially holds. Otherwise (i.e. $\Pr[\mathsf{B}] > 0$), using the triangle inequality, we have

$$\begin{aligned}\Big|\Pr[\mathsf{S}] - \frac{1}{2}\Big| &= \Big|\Pr[\mathsf{S} \wedge \overline{\mathsf{B}}] + \Pr[\mathsf{S}|\mathsf{B}] \cdot \Pr[\mathsf{B}] - \frac{1}{2}\Big|\\ &\leq \Big|\Pr[\mathsf{S} \wedge \overline{\mathsf{B}}] + \frac{1}{2}\Pr[\mathsf{B}] - \frac{1}{2}\Big| + \Pr[\mathsf{B}] \cdot \Big|\Pr[\mathsf{S}|\mathsf{B}] - \frac{1}{2}\Big|.\end{aligned}$$

Then, the proof finishes by noting that we have $|\Pr[\mathsf{S}|\mathsf{B}] - \frac{1}{2}| \leq \frac{1}{2}$. □ (**Fact 1**)

B wNM-DCCA Security and Randomness-Inextractability for Detectable KEM

wNM-DCCA *Security.* For a detectable KEM $\Gamma = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{F})$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define the wNM-DCCA experiment $\mathsf{Expt}^{\mathtt{wNM\text{-}DCCA}}_{\Gamma,\mathcal{A}}(k)$ as described in Fig. 7 (left). As in an adversary $\mathcal{A}$ in the DCCA experiment for KEMs, $\mathcal{A}_1$ in wNM-DCCA experiment is not allowed to submit a decapsulation query c satisfying $\mathsf{F}(pk, c^*, c) = 1$ to the decapsulation oracle. The adversary's final "unrestricted" decapsulation query is captured by the ciphertext c' that is finally output by $\mathcal{A}_1$. It is required that $c' \neq c^*$. *However, we allow* c' *to be such that* $\mathsf{F}(pk, c^*, c') = 1$.

Definition 3. *We say that a detectable KEM* Γ *is* wNM-DCCA *secure if for all PPTAs* $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\Gamma,\mathcal{A}}(k)$ $:= 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{wNM\text{-}DCCA}}_{\Gamma,\mathcal{A}}(k) = 1] - 1/2|$ *is negligible.*

```
$\mathsf{Expt}^{\mathtt{wNM\text{-}DCCA}}_{\Gamma,\mathcal{A}}(k)$ :
    $(pk, sk) \leftarrow \mathsf{KKG}(1^k)$
    $(c^*, K_1^*) \leftarrow \mathsf{Encap}(pk)$
    $K_0^* \leftarrow \mathcal{K}$
    $b \leftarrow \{0,1\}$
    $(c', \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Decap}(sk,\cdot)}(pk, c^*, K_b^*)$
    $K' \leftarrow \mathsf{Decap}(sk, c')$
    $b' \leftarrow \mathcal{A}_2(\mathsf{st}, K')$
    Return $(b' \overset{?}{=} b)$
```

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

Fig. 7. Security experiments for `wNM-DCCA` security (left), and randomness-inextractability (right) for detectable KEMs.

Randomness-Inextractability. For a detectable KEM $\Gamma = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{F})$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, consider the `R-Inext` experiment $\mathsf{Expt}^{\mathtt{R\text{-}Inext}}_{\Gamma,\mathcal{A}}(k)$ as described in Fig. 7 (right).

Definition 4. *We say that a detectable KEM Γ satisfies* randomness-inextractability *if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Gamma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{R\text{-}Inext}}_{\Gamma,\mathcal{A}}(k) = 1]$ is negligible.*

C 2-bit-to-Multi-bit Conversion with a Single Key Pair

Here, we give our 2-bit-to-multi-bit conversion that uses only one key pair of the building block 2-bit PKE scheme. The idea is to use the additional 1-bit of the plaintext space as an "indicator bit" that indicates whether each of component ciphertexts (i.e. $c_{\mathtt{in}}^{(i)}$'s and c_i's) is generated for the inner layer or the outer layer.

Let $\Pi_2 = (\mathsf{PKG}_2, \mathsf{Enc}_2, \mathsf{Dec}_2)$ be a 2-bit PKE scheme whose ciphertext length is "$|c_2|$" and the randomness space of whose encryption algorithm Enc_2 is $\{0,1\}^{\ell}$. Let $\mathcal{C} = (\mathsf{E}, \mathsf{D})$ be a $\mathcal{Q}$-non-malleable (n,k)-code with $n = n(k) \geq k$, and the randomness space of whose encoding algorithm E is $\{0,1\}^{\widehat{\ell}}$. Let $\ell' = n \cdot \ell + \widehat{\ell} + 2k$, and $\mathsf{G} : \{0,1\}^k \rightarrow \{0,1\}^{\ell'}$ be a PRG. Finally, let $E = (\mathsf{SEnc}, \mathsf{SDec})$ be a deterministic SKE scheme whose plaintext space is $\{0,1\}^{k \cdot |c_2|}$. Then, we construct a KEM $\widehat{\Gamma} = (\widehat{\mathsf{KKG}}, \widehat{\mathsf{Encap}}, \widehat{\mathsf{Decap}})$ as in Fig. 8.

The `CCA` security of the KEM $\widehat{\Gamma}$ is guaranteed in essentially the same way as that of $\widetilde{\Gamma}$, as follows.

Theorem 6. *Assume that the PKE scheme Π_2 is `CCA` secure, $\mathcal{C}$ is a $\mathcal{Q}$-non-malleable code, G is a PRG, and the SKE scheme E is `CCA` secure. Then, the KEM $\widehat{\Gamma}$ in Fig. 8 is `CCA` secure.*

The proof can be shown in essentially the same way as that of the 1-bit-to-multi-bit construction $\widetilde{\Gamma}$ via a sequence of games that takes into account all the steps in the security proof of the (suitable analogues of the) double-layered construction (Theorem 1), the bitwise-encrypt construction $\Pi^n_{\mathtt{BE}}$ (Lemmas 6 and 7), and the EtBE construction $\Pi_{\mathtt{EtBE}}$ (Lemmas 8 and 9) and the miscellaneous methods explained in Section 3.3. Since it is redundant and tedious given all the proofs used for Theorem 4 given in this paper, we omit it.

Note that the consistency of each of the "indicator" bits encrypted together with the actual contents (s_i or $K_{\mathtt{in}}^{(i)}$) by the 2-bit scheme, is checked in the decapsulation algorithm $\widehat{\mathsf{Decap}}$. This check prevents a quoting of an inner ciphertext into the outer layer and vice versa, and thus makes the encryption/decryption operations for the inner layer and those of the outer layer virtually independent, as if each layer has an individual key pair, which enables us to conduct the security

```
$\widehat{\mathsf{KKG}}(1^k)$ :
    $(pk, sk) \leftarrow \mathsf{PKG}_2(1^k)$
    $SK \leftarrow (sk, pk)$
    Return $(pk, SK)$.
```

```
$\widehat{\mathsf{Encap}}(pk)$ :
    $K_{\mathtt{in}} = (K_{\mathtt{in}}^{(1)} \| \dots \| K_{\mathtt{in}}^{(k)}) \leftarrow \{0,1\}^k$
    $\forall i \in [k] : c_{\mathtt{in}}^{(i)} \leftarrow \mathsf{Enc}_2(pk, (1 \| K_{\mathtt{in}}^{(i)}))$
    $\alpha \leftarrow \mathsf{G}(K_{\mathtt{in}})$
    Parse $\alpha$ as $(r_1, \dots, r_n, \widehat{r}, K_{\mathtt{out}}, K)$
            $\in (\{0,1\}^{\ell})^n \times \{0,1\}^{\widehat{\ell}} \times (\{0,1\}^k)^2$.
    $s = (s_1 \| \dots \| s_n) \leftarrow \mathsf{E}(1^k, K_{\mathtt{out}}; \widehat{r})$
    $\forall i \in [n] : c_i \leftarrow \mathsf{Enc}_2(pk, (0 \| s_i); r_i)$
    If $\mathtt{DUPCHK}((c_i)_{i \in [n]}) = 1$ then return $\bot$.
    $\widehat{c} \leftarrow \mathsf{SEnc}(K_{\mathtt{out}}, (c_{\mathtt{in}}^{(1)} \| \dots \| c_{\mathtt{in}}^{(k)}))$
    $C \leftarrow (c_1, \dots, c_n, \widehat{c})$
    Return $(C, K)$.
```

```
$\widehat{\mathsf{Decap}}(SK, C)$ :
    $(sk, pk) \leftarrow SK$
    $(c_1, \dots, c_n, \widehat{c}) \leftarrow C$
    If $\mathtt{DUPCHK}((c_i)_{i \in [n]}) = 1$ then return $\bot$.
    $\forall i \in [n] : (\beta_i \| s_i) \leftarrow \mathsf{Dec}_2(sk, c_i)$
    If $\exists i \in [n] : s_i \notin \{0,1\} \lor \beta_i \neq 0$ then return $\bot$.
    $K_{\mathtt{out}} \leftarrow \mathsf{D}(1^k, s = (s_1 \| \dots \| s_n))$
    If $K_{\mathtt{out}} = \bot$ then return $\bot$.
    $(c_{\mathtt{in}}^{(1)} \| \dots \| c_{\mathtt{in}}^{(k)}) \leftarrow \mathsf{SDec}(K_{\mathtt{out}}, \widehat{c})$
    If $\mathsf{SDec}$ has returned $\bot$ then return $\bot$.
    $\forall i \in [k] : (\beta'_i \| K_{\mathtt{in}}^{(i)}) \leftarrow \mathsf{Dec}_2(sk, c_{\mathtt{in}}^{(i)})$
    If $\exists i \in [k] : K_{\mathtt{in}}^{(i)} \notin \{0,1\} \lor \beta'_i \neq 1$ then return $\bot$.
    $K_{\mathtt{in}} \leftarrow (K_{\mathtt{in}}^{(1)} \| \dots \| K_{\mathtt{in}}^{(k)})$
    $\alpha \leftarrow \mathsf{G}(K_{\mathtt{in}})$
    Parse $\alpha$ as $(r_1, \dots, r_n, \widehat{r}, K'_{\mathtt{out}}, K)$
            $\in (\{0,1\}^{\ell})^n \times \{0,1\}^{\widehat{\ell}} \times (\{0,1\}^k)^2$.
    If (a) $\land$ (b) $\land$ (c) then return $K$ else return $\bot$:
      (a) $\forall i \in [n] : \mathsf{Enc}_2(pk, (0 \| s^{(i)}); r_i) = c_i$
      (b) $\mathsf{E}(1^k, K'_{\mathtt{out}}; \widehat{r}) = s$
      (c) $\mathsf{SEnc}(K'_{\mathtt{out}}, (c_{\mathtt{in}}^{(1)} \| \dots \| c_{\mathtt{in}}^{(k)})) = \widehat{c}$
```

Fig. 8. The "2-bit-to-multi-bit" construction with optimal key size: The construction of the KEM $\widehat{\Gamma}$.

proof in essentially the same way as that of $\widetilde{\Gamma}$. Put differently from the viewpoint of the security proof, even if an adversary submits a decryption query with "cross-layer" quoting, i.e. quoting one (or more) of the inner layer components $\{c_{\mathtt{in}}^{*(i)}\}_{i \in [k]}$ of the challenge ciphertext C^* as one (or more) of the outer layer components $\{c_i\}_{i \in [n]}$ in an adversary's decryption query C (and vice versa), the reduction algorithms can simply just reject them. Note that the reduction algorithms can always know all the inner layer component ciphertexts $\{c_{\mathtt{in}}^{*(i)}\}_{i \in [k]}$ and outer layer component ciphertexts $\{c_i^*\}_{i \in [n]}$ used for the challenge ciphertext C^* (but not necessarily the decryption results of them), which enables the reduction algorithms to always find such "cross-layer" quoting queries.

Understanding the Proposed Constructions via Tag-Based Encryption. We note that our proposed 1- and 2-bit-to-multi-bit constructions can also be understood via CCA secure tag-based encryption (TBE) [20, 17] both of whose tag space and plaintext space are just 1-bit. (Recall that TBE is PKE in which the encryption and decryption algorithms take a tag tag as an additional input, and the correctness holds if we use the same tag for both encryption and decryption. This primitive is sometimes also called "PKE with lables", "label-based PKE", etc.) The idea is to use the 1-bit tag as the "indicator" bit in $\widehat{\Gamma}$.

Specifically, let us denote a TBE scheme with 1-bit tag-space and 1-bit plaintext space by $\mathcal{T}_1$. Then consider a modification of our 2-bit-to-multi-bit construction $\widehat{\Gamma}$, as follows: Replace the key generation algorithm PKG_2 with the key generation algorithm of $\mathcal{T}_1$.; Replace "$\mathsf{Enc}_2(pk, (1 \| K_{\mathtt{in}}^{(i)}))$" in $\widehat{\mathsf{Encap}}$ with an encryption of $K_{\mathtt{in}}^{(i)}$ using the tag "1" by the encryption algorithm of $\mathcal{T}_1$, and correspondingly replace "$\mathsf{Dec}_2(sk, c_{\mathtt{in}}^{(i)})$" in $\widehat{\mathsf{Decap}}$ with decryption of $c_{\mathtt{in}}^{(i)}$ using the same tag "1" by the decryption algorithm of $\mathcal{T}_1$.; Likewise, replace "$\mathsf{Enc}_2(pk, (0 \| s_i); r_i)$" in $\widehat{\mathsf{Encap}}$ with encryption of s_i using the tag "0" by the encryption algorithm of $\mathcal{T}_1$, and correspondingly, replace "$\mathsf{Dec}_2(sk, c_i)$" in $\widehat{\mathsf{Decap}}$ with decryption of c_i using the tag "0".

Let us denote by $\Gamma_{\mathcal{T}}$ this modified version of $\widehat{\Gamma}$ using the TBE scheme $\mathcal{T}_1$. Then, we can prove that the KEM $\Gamma_{\mathcal{T}}$ is $\mathtt{CCA}$ secure under the assumption that the TBE scheme $\mathcal{T}_1$ is $\mathtt{CCA}$ secure (which is the version called "full CCA" in [17]) and with the same assumptions on $\mathcal{C}$, E, and G). (The proof is again very similar to those for $\widetilde{\Gamma}$ and $\widehat{\Gamma}$.)

Furthermore, as explained in [17], given a $\mathtt{CCA}$ secure PKE scheme that can encrypt 2-bit or longer plaintexts, one can construct a $\mathtt{CCA}$ secure TBE scheme with 1-bit tag-space and 1-bit plaintext space by encrypting $(\mathsf{tag}\|m)$ by the encryption algorithm of the 2-bit PKE scheme, where $\mathsf{tag} \in \{0,1\}$ and $m \in \{0,1\}$ are a tag and a plaintext, respectively, and in the decryption algorithm, whether the first bit of the decryption result and the tag that is input to the decryption algorithm match. Now, notice that if we instantiate $\mathcal{T}_1$ in $\Gamma_{\mathcal{T}}$ with this TBE scheme based on 2-bit PKE, we obtain the proposed 2-bit-to-multi-bit construction $\widehat{\Gamma}$ itself.

The PKE-to-TBE construction explained in [17] is applicable only for PKE schemes that can encrypt messages whose length is the sum of a tag and a plaintext (to be encrypt by TBE), and hence, we cannot use it to obtain a TBE scheme with 1-bit tag space and 1-bit plaintext space, from a 1-bit PKE scheme. However, there is a simple construction of such a TBE scheme even from a 1-bit PKE scheme. We can just generate two independent key pairs (pk_0, sk_0) and (pk_1, sk_i) of the 1-bit PKE scheme, and regard (pk_0, pk_1) (resp. (sk_0, sk_1)) as a public (resp. secret) key of TBE, and in the encryption (resp. decryption), if a tag is $\mathsf{tag} \in \{0,1\}$, we just use pk_{tag} (resp. sk_{tag}) to encrypt a 1-bit plaintext (resp. decrypt a ciphertext). It is straightforward to see that if the 1-bit PKE scheme is $\mathtt{CCA}$ secure, then the TBE scheme obtained in this way is $\mathtt{CCA}$ secure. Now, if we instantiate the TBE scheme $\mathcal{T}_1$ in $\Gamma_{\mathcal{T}}$ with this TBE scheme based on 1-bit PKE, then we obtain the proposed 1-bit-to-multi-bit construction $\widetilde{\Gamma}$ itself.

D Proofs for Main Results

D.1 Proof of Lemma 5: Non-triviality of Randomness-Inextractability

***Proof for* "$\mathtt{wNM\text{-}DCCA} + \mathtt{UNP} \not\Rightarrow \mathtt{R\text{-}Inext}$."** Here, we show that if there exists a detectable PKE scheme that satisfies $\mathtt{wNM\text{-}DCCA}$ security and unpredictability, then there exists a detectable PKE scheme that satisfies $\mathtt{wNM\text{-}DCCA}$ security and unpredictability, but does not satisfy randomness-inextractability.

Note that if there exists a detectable PKE scheme satisfying ($\mathtt{wNM\text{-}}$)$\mathtt{DCCA}$ security and unpredictability (which is guaranteed by assumption), then due to the result by Hohenberger et al. [16], there exists a $\mathtt{CCA}$ secure PKE scheme $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ whose plaintext space is $\{0,1\}^{k+1}$. Then, using this PKE scheme Π as a building block, we construct the "separating" PKE scheme $\Pi_{\mathsf{SEP1}} = (\mathsf{PKG}_{\mathsf{SEP1}}, \mathsf{Enc}_{\mathsf{SEP1}}, \mathsf{Dec}_{\mathsf{SEP1}}, \mathsf{F}_{\mathsf{SEP1}})$ whose plaintext space is $\{0,1\}$, as described in Fig 9 (left).

Note that the construction in Fig. 9 (left) is in fact exactly the "transformation" of a $\mathtt{CCA}$ secure PKE scheme into a $\mathtt{CCA}$ secure tag-based encryption (TBE) scheme explained by Kiltz [17], except that the tag tag is always chosen uniformly at random from $\{0,1\}^k$. (Here, we consider the $\mathtt{CCA}$ security of a TBE scheme in which an adversary is allowed to submit a tag/ciphertext pair (tag, c) as a decryption query, as long as it is different (as a pair) from the challenge tag/ciphertext pair (tag^*, c^*).) Therefore, it is straightforward from [17] that if Π is $\mathtt{CCA}$ secure, then Π_{SEP1} is $\mathtt{wNM\text{-}DCCA}$ secure (in particular, the final "unrestricted" decryption query in the $\mathtt{wNM\text{-}DCCA}$ experiment can be handled by the $\mathtt{CCA}$ security of the tag-based encryption scheme).

```
$\mathsf{PKG}_{\mathsf{SEP1}}(1^k)$ :
    Return $(pk, sk) \leftarrow \mathsf{PKG}(1^k)$.
$\mathsf{Enc}_{\mathsf{SEP1}}(pk, m)$ :
    $\mathsf{tag} \leftarrow \{0,1\}^k$
    $c \leftarrow \mathsf{Enc}(pk, (\mathsf{tag}\|m))$
    Return $C \leftarrow (\mathsf{tag}, c)$.
$\mathsf{Dec}_{\mathsf{SEP1}}(sk, C)$ :
    $(\mathsf{tag}, c) \leftarrow C$
    $(\mathsf{tag}'\|m) \leftarrow \mathsf{Dec}(sk, c)$
    If $\mathsf{Dec}$ has returned $\bot$ or $\mathsf{tag}' \neq \mathsf{tag}$ then return $\bot$.
    Return $m$.
$\mathsf{F}_{\mathsf{SEP1}}(pk, C^*, C')$ :
    $(\mathsf{tag}^*, c^*) \leftarrow C^*$
    $(\mathsf{tag}', c') \leftarrow C'$
    Return $(\mathsf{tag}^* \overset{?}{=} \mathsf{tag}')$.
```

```
$\mathsf{PKG}_{\mathsf{SEP2}}(1^k)$ :
    Return $(pk, sk) \leftarrow \mathsf{PKG}(1^k)$.
$\mathsf{Enc}_{\mathsf{SEP2}}(pk, m)$ :
    $c \leftarrow \mathsf{Enc}(pk, m)$
    Return $C \leftarrow (0\|c)$.
$\mathsf{Dec}_{\mathsf{SEP2}}(sk, C)$ :
    Parse $C$ as $(\gamma\|c)$ s.t. $|\gamma| = 1$.
    If $\gamma = 1$ then return $\bot$.
    Return $m \leftarrow \mathsf{Dec}(sk, c)$.
$\mathsf{F}_{\mathsf{SEP2}}(pk, C^*, C')$ :
    Parse $C^*$ as $(\gamma^*\|c^*)$ s.t. $|\gamma^*| = 1$.
    Parse $C'$ as $(\gamma'\|c')$ s.t. $|\gamma'| = 1$.
    If $\gamma^* \neq \gamma'$ or $\mathsf{F}(pk, c^*, c') = 1$
            then return 1 else return 0.
```

Fig. 9. The "separating" detectable PKE schemes used to show the separations of security notions. The scheme Π_{SEP1} that separates randomness-inextractability from the combination of `wNM-DCCA` security and unpredictability (left), and the scheme Π_{SEP2} that separates unpredictability from the combination of `wNM-DCCA` security and randomness-inextractability (right).

Furthermore, it is also easy to see that Π_{SEP1} is information-theoretically unpredictable. This is because the first component tag^* is chosen uniformly at random, and thus even if an adversary is computationally unbounded, it can output a ciphertext $C' = (\mathsf{tag}', c)$ such that $\mathsf{F}_{\mathsf{SEP1}}(pk, C^*, C') = 1$ (i.e. $\mathsf{tag}^* = \mathsf{tag}'$) for an unseen ciphertext $C^* = (\mathsf{tag}^*, c^*)$ only with negligible probability, which implies that Π_{SEP1} unconditionally satisfies unpredictability.

Finally, we show that Π_{SEP1} does not satisfy randomness-inextractability. Specifically, consider an adversary $\mathcal{A}$ that first submits any plaintext m to the experiment, and then is given a public key pk and the challenge ciphertext $C^* = (\mathsf{tag}^*, c^*)$, where by definition c^* is an encryption of $(\mathsf{tag}^*\|m)$ generated by $c^* \leftarrow \mathsf{Enc}(pk, (\mathsf{tag}^*\|m))$. Now, $\mathcal{A}$ picks a randomness $r' = (\mathsf{tag}^*, r)$ where r is any randomness in the randomness space of Enc, and terminates with output (m, r'). Note that we have $C' = \mathsf{Enc}_{\mathsf{SEP1}}(pk, m; r' = (\mathsf{tag}^*, r)) = (\mathsf{tag}^*\|\mathsf{Enc}(pk, (\mathsf{tag}^*\|m); r)) = (\mathsf{tag}^*\|c')$ for some c', and thus $\mathsf{F}_{\mathsf{SEP1}}(pk, C^*, C') = 1$ holds. That is, $\mathcal{A}$ has maximum advantage in breaking the randomness-inextractability of Π_{SEP1}, meaning that Π_{SEP1} does not satisfy randomness-inextractability.

Proof for "`wNM-DCCA` + `R-Inext` $\not\Rightarrow$ `UNP`." Here, we show that if there exists a detectable PKE scheme that satisfies `wNM-DCCA` security and randomness-inextractability, then there exists a PKE scheme that satisfies `wNM-DCCA` security and randomness-inextractability, but does not satisfy unpredictability.

Let $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{F})$ be a detectable PKE scheme that satisfies `wNM-DCCA` security and randomness-inextractability that is guaranteed to exist by assumption. Then, consider the "separating" scheme $\Pi_{\mathsf{SEP2}} = (\mathsf{PKG}_{\mathsf{SEP2}}, \mathsf{Enc}_{\mathsf{SEP2}}, \mathsf{Dec}_{\mathsf{SEP2}}, \mathsf{F}_{\mathsf{SEP2}})$ as described in Fig. 9 (right).

Firstly, it is not hard to see that Π_{SEP2} preserves the `wNM-DCCA` security of the underlying detectable PKE scheme Π. Specifically, using a `wNM-DCCA` adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ as a building block, we can straightforwardly construct a reduction algorithm $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$ that attacks the `wNM-DCCA` security of the building block scheme Π, such that $\mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\Pi,\mathcal{B}}(k) = \mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\Pi_{\mathsf{SEP2}},\mathcal{A}}(k)$. In particular, the challenge ciphertext C^* for $\mathcal{A}$ is always of the form $C^* = (0\|c^*)$, and thus the allowable set of decryption queries $C = (\gamma\|c)$ by the second stage $\mathcal{A}_2$ of the `wNM-DCCA` adversary

$\mathcal{A}$, are those satisfying $\gamma = 0$ and $\mathsf{F}(pk, c^*, c) = 0$ simultaneously. However, such ciphertexts can be easily handled by the reduction algorithm $\mathcal{B}$ by $\mathcal{B}_2$'s own decryption oracle. The final "unrestricted" decryption query output by $\mathcal{A}_2$ can also be dealt with straightforwardly by the final decryption query allowed for the second stage $\mathcal{B}_2$ of the reduction algorithm.

Secondly, it is also not hard to see that Π_{SEP2} preserves the randomness-inextractability of the underlying scheme Π. Specifically, recall that randomness-inextractability is always about ciphertexts generated "honestly" via the encryption algorithm $\mathsf{Enc}_{\mathsf{SEP2}}$. Therefore, to break the randomness-inextractability of Π_{SEP2}, an adversary cannot use the condition $\gamma' \neq \gamma^* = 0$, and thus it has to essentially break the randomness-inextractability of the underlying scheme Π, which is hard by assumption.

Finally, we note that it is easy to break the unpredictability of Π_{SEP2}. Specifically, consider an adversary $\mathcal{A}$ that outputs $C' = (1\|c)$ with any c (which need not even be in the range of $\mathsf{Enc}_{\mathsf{SEP2}}$) and any plaintext m. Then, since a ciphertext generated "honestly" by $\mathsf{Enc}_{\mathsf{SEP2}}$ always has the prefix 0 (and this is the case in the unpredictability experiment), the prefix of C^* and that of C' are distinct. This implies $\mathsf{F}_{\mathsf{SEP2}}(pk, C^*, C') = 1$, and thus the adversary $\mathcal{A}$ has maximum advantage in breaking the unpredictability of Π_{SEP2}. Hence, Π_{SEP2} does not satisfy unpredictability. □ (**Lemma 5**)

D.2 Proof of Theorem 1: Our First Proof of CCA Security of $\Gamma_{\mathtt{DL}}$

We will show that for any PPTA $\mathcal{A}$ that attacks the $\mathtt{CCA}$ security of the KEM $\Gamma_{\mathtt{DL}}$ and makes $Q = Q(k) > 0$ decapsulation queries, there exist PPTAs $\mathcal{B}_{\mathtt{in1}}$, $\mathcal{B}_{\mathtt{in2}}$, $\mathcal{B}_{\mathtt{out}}$, $\mathcal{B}'_{\mathtt{out}}$, and $\mathcal{B}'_{\mathtt{in}}$, such that

$$\begin{aligned}\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k) \leq Q \cdot 2^{-(n-1)} + 2 \cdot \mathsf{Adv}^{\mathtt{DCCA}}_{\Gamma_{\mathtt{in}},\mathcal{B}_{\mathtt{in1}}}(k) + \mathsf{Adv}^{\mathtt{DCCA}}_{\Gamma_{\mathtt{in}},\mathcal{B}_{\mathtt{in2}}}(k) + \mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi_{\mathtt{out}},\mathcal{B}_{\mathtt{out}}}(k)\\ + Q \cdot \mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{out}},\mathcal{B}'_{\mathtt{out}}}(k) + Q^2 \cdot \mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{in}},\mathcal{B}'_{\mathtt{in}}}(k), \quad (3)\end{aligned}$$

which, due to the assumptions on the building blocks $\Gamma_{\mathtt{in}}$ and $\Pi_{\mathtt{out}}$, implies that $\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k)$ is negligible, and hence proves the theorem.

To this end, fix arbitrarily a PPTA adversary $\mathcal{A}$ that attacks the $\mathtt{CCA}$ security of the KEM $\Gamma_{\mathtt{DL}}$ and makes in total $Q > 0$ decapsulation queries. Consider the following sequence of games: (Here, the values with asterisk (*) represent those related to $\mathcal{A}$'s challenge ciphertext.)

Game 1: This is the experiment $\mathsf{Expt}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k)$ itself. For defining the subsequent games, we make Game 1 pick a randomness $d^* \in \{0,1\}^n$, which we call the "dummy" plaintext, uniformly at random at the beginning of the game. (The value d^* does not appear in $\mathcal{A}$'s view in Game 1, and thus does not affect $\mathcal{A}$'s behavior at all.)

Game 2: Same as Game 1, except that all decapsulation queries c satisfying $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) \in \{c^*_{\mathtt{in}}, d^*\}$ are answered with $\bot$.

Game 3: Same as Game 2, except that $r^* \in \{0,1\}^\ell$ and $K^*_1 \in \{0,1\}^k$ are picked uniformly at random, independently of $c^*_{\mathtt{in}}$. More precisely, the steps "$(c^*_{\mathtt{in}}, \alpha^*) \leftarrow \mathsf{Encap}_{\mathtt{in}}(pk_{\mathtt{in}})$; Parse α^* as $(r^*, K^*_1) \in \{0,1\}^\ell \times \{0,1\}^k$" in Game 2 are replaced with the steps "$(c^*_{\mathtt{in}}, \alpha^*) \leftarrow \mathsf{Encap}_{\mathtt{in}}(pk_{\mathtt{in}})$; $r^* \leftarrow \{0,1\}^\ell$; $K^*_1 \leftarrow \{0,1\}^k$," and we do not use α^* (corresponding to $c^*_{\mathtt{in}}$) at all.

Game 4: Same as Game 3, except that all decapsulation queries c satisfying $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) \in \{c^*_{\mathtt{in}}, d^*\}$ *or* $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ are answered with $\bot$.

Game 5: Same as Game 4, except that the information of $c^*_{\mathtt{in}}$ is erased from c^*, and instead the "dummy" plaintext d^* is encrypted. More precisely, the step "$c^* \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}})$" in Game 4 is replaced with the step "$c^* \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, d^*)$."

Table 2. An overview of the games used in the proof of Theorem 1.

Game	c^*	r^* and K_1^*	$\mathcal{A}$'s decapsulation query c is answered with
1	$\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$	$(r^* \\| K_1^*) = \alpha^*$	$\mathsf{Decap}_{\mathtt{DL}}(SK, c)$
2	$\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$	$(r^* \\| K_1^*) = \alpha^*$	$\begin{cases} \bot & \text{if } \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) \in \{c^*_{\mathtt{in}}, d^*\} \\ \mathsf{Decap}_{\mathtt{DL}}(SK, c) & \text{otherwise} \end{cases}$
3	$\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$	$r^* \leftarrow \{0,1\}^\ell$ $K_1^* \leftarrow \{0,1\}^k$	$\begin{cases} \bot & \text{if } \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) \in \{c^*_{\mathtt{in}}, d^*\} \\ \mathsf{Decap}_{\mathtt{DL}}(SK, c) & \text{otherwise} \end{cases}$
4	$\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$	$r^* \leftarrow \{0,1\}^\ell$ $K_1^* \leftarrow \{0,1\}^k$	$\begin{cases} \bot & \text{if } \begin{array}{l} \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) \in \{c^*_{\mathtt{in}}, d^*\} \\ \quad \text{or } \mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1 \end{array} \\ \mathsf{Decap}_{\mathtt{DL}}(SK, c) & \text{otherwise} \end{cases}$
5	$\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, d^*; r^*)$	$r^* \leftarrow \{0,1\}^\ell$ $K_1^* \leftarrow \{0,1\}^k$	$\begin{cases} \bot & \text{if } \begin{array}{l} \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) \in \{c^*_{\mathtt{in}}, d^*\} \\ \quad \text{or } \mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1 \end{array} \\ \mathsf{Decap}_{\mathtt{DL}}(SK, c) & \text{otherwise} \end{cases}$

The above completes the description of the games. Table 2 is an overview of the differences among the games in terms of how the challenge ciphertext c^* and the values (r^*, K_1^*) are generated, and how the decapsulation oracle works.

We say that a decapsulation query c submitted by $\mathcal{A}$ in the above games is *inner-dangerous* if c satisfies the following two conditions:

(1) $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, d^*, \bot\}$, and
(2) $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$.

Furthermore, we say that a decapsulation query c submitted by $\mathcal{A}$ in the above games is *outer-dangerous* if c satisfies the following four conditions[3]:

(1) $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$,
(2) $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, d^*, \bot\}$,
(3) $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c_{\mathtt{in}}) = (r \| K) \neq \bot$, and
(4) $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c_{\mathtt{in}}; r) = c$.

For $i \in [5]$, we define the following events in Game i:

S_i**:** $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs).
In_i**:** $\mathcal{A}$ submits at least one inner-dangerous decapsulation query.
$\mathsf{In}_i^{(j)}$ **where** $j \in [Q]$**:** $\mathcal{A}$'s j-th decapsulation query is inner-dangerous.

By definitions of the games and events, we can show the following upperbound on $\mathcal{A}$'s $\mathtt{CCA}$ advantage $\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k)$:

[3] Looking ahead, the definition of an outer-dangerous query will be used only in the proof of Claim 5, but we believe that it is helpful to introduce the definition here to compare it with the above definition of an inner-dangerous query, and also with the definitions of inner/outer queries used in the proof of Theorem 2 (in Appendix D.3).

Claim 1 *$\mathcal{A}$'s* $\mathtt{CCA}$ *advantage* $\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k)$ *can be upperbounded as follows:*

$$\begin{aligned}\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k) \le 2\cdot\Big|\Pr[\mathsf{S}_1]-\Pr[\mathsf{S}_2]\Big| + 2\cdot\Big|\Pr[\mathsf{S}_2\wedge\overline{\mathsf{In}_2}] - \Pr[\mathsf{S}_3\wedge\overline{\mathsf{In}_3}] + \frac{1}{2}(\Pr[\mathsf{In}_2]-\Pr[\mathsf{In}_3])\Big| \\ + 2\cdot\Big|\Pr[\mathsf{S}_3\wedge\overline{\mathsf{In}_3}] + \frac{1}{2}\Pr[\mathsf{In}_3] - \frac{1}{2}\Big| + \sum_{i\in\{2,3\}}\Big|\Pr[\mathsf{In}_i]-\Pr[\mathsf{In}_{i+1}]\Big| \\ + \Big|\sum_{j\in[Q]}\Big(\Pr[\mathsf{In}_4^{(j)}]-\Pr[\mathsf{In}_5^{(j)}]\Big)\Big| + \sum_{j\in[Q]}\Pr[\mathsf{In}_5^{(j)}]. \quad (4)\end{aligned}$$

Proof of Claim 1. By the triangle inequality, we have

$$\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k) = 2\cdot\Big|\Pr[\mathsf{S}_1]-\frac{1}{2}\Big| \le 2\cdot\Big|\Pr[\mathsf{S}_1]-\Pr[\mathsf{S}_2]\Big| + 2\cdot\Big|\Pr[\mathsf{S}_2]-\frac{1}{2}\Big|.$$

Furthermore, by Fact 1 (stated in Appendix A.4) and the triangle inequality, we have

$$\begin{aligned}\Big|\Pr[\mathsf{S}_2]-\frac{1}{2}\Big| &\le \Big|\Pr[\mathsf{S}_2\wedge\overline{\mathsf{In}_2}] + \frac{1}{2}\Pr[\mathsf{In}_2] - \frac{1}{2}\Big| + \frac{1}{2}\Pr[\mathsf{In}_2] \\ &\le \Big|\Pr[\mathsf{S}_2\wedge\overline{\mathsf{In}_2}] - \Pr[\mathsf{S}_3\wedge\overline{\mathsf{In}_3}] + \frac{1}{2}(\Pr[\mathsf{In}_2]-\Pr[\mathsf{In}_3])\Big| \\ &\quad + \Big|\Pr[\mathsf{S}_3\wedge\overline{\mathsf{In}_3}] + \frac{1}{2}\Pr[\mathsf{In}_3] - \frac{1}{2}\Big| + \frac{1}{2}\sum_{i\in\{2,3\}}\Big|\Pr[\mathsf{In}_i]-\Pr[\mathsf{In}_{i+1}]\Big| + \frac{1}{2}\Pr[\mathsf{In}_4].\end{aligned}$$

Finally, noting that $\Pr[\mathsf{In}_4] = \Pr[\bigvee_{j\in[Q]}\mathsf{In}_4^{(j)}]$, and applying the union bound and the triangle inequality, we estimate the upperbound of $\Pr[\mathsf{In}_4]$ as follows:

$$\Pr[\mathsf{In}_4] \le \sum_{j\in[Q]}\Pr[\mathsf{In}_4^{(j)}] \le \Big|\sum_{j\in[Q]}\Big(\Pr[\mathsf{In}_4^{(j)}]-\Pr[\mathsf{In}_5^{(j)}]\Big)\Big| + \sum_{j\in[Q]}\Pr[\mathsf{In}_5^{(j)}].$$

Combining all the inequalities yields the claim. □ (**Claim 1**)

In the following, we show upperbounds of the terms that appear in the right hand side of Equation (4).

Claim 2 $|\Pr[\mathsf{S}_1]-\Pr[\mathsf{S}_2]| \le Q\cdot 2^{-n}$.

Proof of Claim 2. Note that the difference between Game 1 and Game 2 is only in how a decapsulation query c satisfying $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}},c)\in\{c^*_{\mathtt{in}},d^*\}$ is answered. (Such a query is answered with $\mathsf{Decap}_{\mathtt{DL}}(SK,c)$ in Game 1, while it is answered with $\bot$ in Game 2.) We first show that if $\mathcal{A}$'s decapsulation query c satisfies $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}},c)=c^*_{\mathtt{in}}$, then c is also answered with $\bot$ in Game 1 as well. To see this, let c be any decapsulation query satisfying $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}},c)=c^*_{\mathtt{in}}$. Recall that in order for this query to satisfy $\mathsf{Decap}_{\mathtt{DL}}(SK,c)\ne\bot$, it must additionally satisfy $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}},c^*_{\mathtt{in}};r^*)=c$, because we have $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}},c_{\mathtt{in}})=\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}},c^*_{\mathtt{in}})=(r^*\|K_1^*)$. However, the only ciphertext c satisfying this additional condition is the challenge ciphertext c^* itself, while according to the rule of the $\mathtt{CCA}$ experiment, any of $\mathcal{A}$'s decapsulation queries c must satisfy $c\ne c^*$. Therefore, any decapsulation query c satisfying $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}},c)=c^*_{\mathtt{in}}$ is answered with $\bot$ in Game 1 as well.

Therefore, Game 1 and Game 2 proceed identically unless $\mathcal{A}$ submits at least one decapsulation query c satisfying $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ and $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = d^*$. Hence, $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]|$ can be upperbounded by the probability that $\mathcal{A}$ submits such a query in Game 1. However, recall that $d^* \in \{0,1\}^n$ is chosen uniformly at random, and is information-theoretically hidden from $\mathcal{A}$ in Game 1. Therefore, the probability that $\mathcal{A}$ submits a query c satisfying $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = d^*$ in Game 1 is upperbounded by $Q \cdot 2^{-n}$. This completes the proof of the claim. □ (**Claim 2**)

Claim 3 *For any constants $p, q \in [0,1]$, there exists a PPTA $\mathcal{B}_{\mathtt{in}}$ such that $\mathsf{Adv}^{\mathtt{DCCA}}_{\Gamma_{\mathtt{in}},\mathcal{B}_{\mathtt{in}}}(k) = |p \cdot (\Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}]) + q \cdot (\Pr[\mathsf{In}_2] - \Pr[\mathsf{In}_3])|$.*

Proof of Claim 3. Fix arbitrarily $p, q \in [0,1]$. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}_{\mathtt{in}}$ that attacks the DCCA security of the detectable KEM $\Gamma_{\mathtt{in}}$ with the claimed advantage. The description of $\mathcal{B}_{\mathtt{in}}$ is as follows:

$\mathcal{B}_{\mathtt{in}}^{\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}},\cdot)}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, \alpha^*_\gamma)$**:** (where $\gamma \in \{0,1\}$ is $\mathcal{B}_{\mathtt{in}}$'s challenge bit in the DCCA experiment) $\mathcal{B}_{\mathtt{in}}$ first picks two coins $b_p, b_q \in \{0,1\}$ such that $b_p = 1$ (resp. $b_q = 1$) holds with probability p (resp. q). Next, $\mathcal{B}_{\mathtt{in}}$ parses α^*_γ as $(r^*, K^*_1) \in \{0,1\}^\ell \times \{0,1\}^k$, picks $K^*_0 \in \{0,1\}^k$ and $b \in \{0,1\}$ uniformly at random, and then runs $(pk_{\mathtt{out}}, sk_{\mathtt{out}}) \leftarrow \mathsf{PKG}_{\mathtt{out}}(1^k)$ and $c^* \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$. Then $\mathcal{B}_{\mathtt{in}}$ picks $d^* \in \{0,1\}^n$ uniformly at random, sets $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathtt{out}})$, and runs $\mathcal{A}(PK, c^*, K^*_b)$. $\mathcal{B}_{\mathtt{in}}$ answers $\mathcal{A}$'s decapsulation queries c as follows: $\mathcal{B}_{\mathtt{in}}$ runs $c_{\mathtt{in}} \leftarrow \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c)$. If $c_{\mathtt{in}} \in \{c^*_{\mathtt{in}}, d^*, \perp\}$, then $\mathcal{B}_{\mathtt{in}}$ returns $\perp$ to $\mathcal{A}$. Otherwise (i.e. $c_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, d^*, \perp\}$), if $\mathsf{F}_{\mathtt{in}}(sk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$, then $\mathcal{B}_{\mathtt{in}}$ has detected that this query is inner-dangerous, and terminates with output $\gamma' \leftarrow b_q$. Otherwise, $\mathcal{B}_{\mathtt{in}}$ performs the remaining procedure of $\mathsf{Decap}_{\mathtt{DL}}(SK, c)$ using $\mathcal{B}_{\mathtt{in}}$'s own decapsulation oracle as a substitute for $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, \cdot)$, and returns the decapsulation result to $\mathcal{A}$.
When $\mathcal{A}$ terminates with output its guess bit b', $\mathcal{B}_{\mathtt{in}}$ sets $\gamma' \leftarrow 1$ if both $b' = b$ and $b_p = 1$ hold, otherwise (i.e. $b' \neq b$ or $b_p = 0$) $\mathcal{B}_{\mathtt{in}}$ sets $\gamma' \leftarrow 0$, and terminates with output γ'.

The above completes the description of $\mathcal{B}_{\mathtt{in}}$. Note that $\mathcal{B}_{\mathtt{in}}$ never submits a prohibited decapsulation query $c_{\mathtt{in}}$ satisfying $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$ to its decapsulation oracle. Let $\mathsf{In}_{\mathcal{B}}$ be the event that $\mathcal{A}$ submits an inner-dangerous decapsulation query in the experiment simulated by $\mathcal{B}$.

Consider the case when $\gamma = 1$. It is easy to see that in this case, $\mathcal{B}_{\mathtt{in}}$ simulates Game 2 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b, *until the point $\mathcal{A}$ submits an inner-dangerous query (i.e. a query c satisfying $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, d^*, \perp\}$ and $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$).* In particular, the value α^*_1 associated with $c^*_{\mathtt{in}}$ is used as (r^*, K^*_1) as is done in Game 2. All other values (PK, c^*, and the answers to decapsulation queries) are distributed identically to those of Game 2. Furthermore, $\mathcal{B}_{\mathtt{in}}$ can detect whether $\mathcal{A}$'s query is inner-dangerous by using $sk_{\mathtt{out}}$, $\mathsf{F}_{\mathtt{out}}$, c^*, $c^*_{\mathtt{in}}$, d^*, and $\mathsf{F}_{\mathtt{in}}$. These imply that $\Pr[b' = b \wedge \overline{\mathsf{In}_{\mathcal{B}}} | \gamma = 1] = \Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}]$ and $\Pr[\mathsf{In}_{\mathcal{B}} | \gamma = 1] = \Pr[\mathsf{In}_2]$. Recall that $\mathcal{B}_{\mathtt{in}}$ outputs $\gamma' = 1$ only if either (1) $\mathcal{A}$ succeeds in guessing b without making any inner-dangerous queries (i.e. $b' = b \wedge \overline{\mathsf{In}_{\mathcal{B}}}$ occurs) and $b_p = 1$, or (2) $\mathcal{A}$ makes an inner-dangerous query ($\mathsf{In}_{\mathcal{B}}$ occurs) and $b_q = 1$. Furthermore, the choice of the coins b_p and b_q is independent of the behavior of $\mathcal{A}$ and $\mathcal{B}_{\mathtt{in}}$'s challenge bit. These imply

$$\begin{aligned}\Pr[\gamma' = 1 | \gamma = 1] &= \Pr[b_p = 1 \wedge b' = b \wedge \overline{\mathsf{In}_{\mathcal{B}}} | \gamma = 1] + \Pr[b_q = 1 \wedge \mathsf{In}_{\mathcal{B}} | \gamma = 1] \\ &= p \cdot \Pr[b' = b \wedge \overline{\mathsf{In}_{\mathcal{B}}} | \gamma = 1] + q \cdot \Pr[\mathsf{In}_{\mathcal{B}} | \gamma = 1] \\ &= p \cdot \Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}] + q \cdot \Pr[\mathsf{In}_2].\end{aligned}$$

On the other hand, when $\gamma = 0$, $\mathcal{B}_{\mathtt{in}}$ simulates Game 3 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b, *until the point $\mathcal{A}$ submits an inner-dangerous query.* In particular, the uniformly chosen random value α_0^* (independent of $c_{\mathtt{in}}^*$) is used as (r^*, K_1^*), which is exactly how they are distributed in Game 3. The rest is unchanged from the case of $\gamma = 1$, and thus, with a similar argument to the above, we have $\Pr[\gamma' = 1 | \gamma = 0] = p \cdot \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] + q \cdot \Pr[\mathsf{In}_3]$.

We can calculate $\mathcal{B}_{\mathtt{in}}$'s $\mathtt{DCCA}$ advantage as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{DCCA}}_{\Gamma_{\mathtt{in}},\mathcal{B}_{\mathtt{in}}}(k) &= 2 \cdot \left|\Pr[\gamma' = \gamma] - \frac{1}{2}\right| = \left|\Pr[\gamma' = 1|\gamma = 1] - \Pr[\gamma' = 1|\gamma = 0]\right| \\ &= \left|p \cdot (\Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}]) + q \cdot (\Pr[\mathsf{In}_2] - \Pr[\mathsf{In}_3])\right|.\end{aligned}$$

□ (**Claim 3**)

Claim 4 $|\Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] + \frac{1}{2}\Pr[\mathsf{In}_3] - \frac{1}{2}| = 0.$

Proof of Claim 4. Note that in Game 3, the real session-key K_1^* is chosen uniformly at random, independently of $\mathcal{A}$'s challenge ciphertext. Furthermore, the behavior of the decapsulation oracle in Game 3 is independent of the challenge bit. Since the distribution of K_1^* and that of K_0^* are identical, $\mathcal{A}$'s view in Game 3 is distributed identically regardless of the challenge bit b. This guarantees that $\Pr[\mathsf{S}_3] = 1/2$, and that the event $\overline{\mathsf{In}_3}$ is independent of S_3. Hence, we have $\Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] = \frac{1}{2}\Pr[\overline{\mathsf{In}_3}]$, which implies the claim. □ (**Claim 4**)

Claim 5 *There exists a PPTA $\mathcal{B}_{\mathtt{out}}$ such that* $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi_{\mathtt{out}},\mathcal{B}_{\mathtt{out}}}(k) \geq |\Pr[\mathsf{In}_3] - \Pr[\mathsf{In}_4]|$.

Proof of Claim 5. For $i \in \{3, 4\}$, let Out_i be the event that in Game i, $\mathcal{A}$ submits at least one outer-dangerous decapsulation query.

Note that the difference between Game 3 and Game 4 is in how $\mathcal{A}$'s decapsulation query c satisfying the conditions $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$, $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \notin \{c_{\mathtt{in}}^*, d^*\}$, and $\mathsf{Decap}_{\mathtt{DL}}(SK, c) \neq \bot$, is answered. (Such a query is answered with $\bot$ in Game 4, while it is answered with non-$\bot$ in Game 3.) However, note that $\mathsf{Decap}_{\mathtt{DL}}(SK, c) \neq \bot$ implies $c_{\mathtt{in}} \neq \bot$, $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c_{\mathtt{in}}) = (r \| K) \neq \bot$, and $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c_{\mathtt{in}}; r) = c$. Thus, a decapsulation query c satisfying the above conditions in fact satisfies the conditions of an outer-dangerous query. This means that Game 3 and Game 4 proceed identically unless Out_3 or Out_4 occurs in the corresponding games, and thus we have

$$|\Pr[\mathsf{In}_3] - \Pr[\mathsf{In}_4]| \leq \Pr[\mathsf{Out}_3] = \Pr[\mathsf{Out}_4]. \tag{5}$$

Now, using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}_{\mathtt{out}}$ that attacks the randomness-inextractability of the detectable PKE scheme $\Pi_{\mathtt{out}}$ with advantage $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi_{\mathtt{out}},\mathcal{B}_{\mathtt{out}}}(k) = \Pr[\mathsf{Out}_4]$, which implies the claim. The description of $\mathcal{B}_{\mathtt{out}} = (\mathcal{B}_{\mathtt{out1}}, \mathcal{B}_{\mathtt{out2}})$ is as follows:

$\mathcal{B}_{\mathtt{out1}}(1^k)$**:** $\mathcal{B}_{\mathtt{out1}}$ first runs $(pk_{\mathtt{in}}, sk_{\mathtt{in}}) \leftarrow \mathsf{KKG}_{\mathtt{in}}(1^k)$ and $(c_{\mathtt{in}}^*, \alpha^*) \leftarrow \mathsf{Encap}_{\mathtt{in}}(pk_{\mathtt{in}})$. Then $\mathcal{B}_{\mathtt{out1}}$ sets $M \leftarrow c_{\mathtt{in}}^*$ and $\mathsf{st}_{\mathcal{B}} \leftarrow (\mathcal{B}_{\mathtt{out1}}$'s entire view), and terminates with output $(M, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathtt{out2}}^{\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}},\cdot)}(\mathsf{st}_{\mathcal{B}}, pk_{\mathtt{out}}, c^*)$**:** $\mathcal{B}_{\mathtt{out2}}$ sets $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathtt{out}})$, picks $K^* \in \{0,1\}^k$ and $d^* \in \{0,1\}^n$ uniformly at random, and runs $\mathcal{A}(PK, c^*, K^*)$.

$\mathcal{B}_{\mathtt{out2}}$ answers $\mathcal{A}$'s decapsulation queries as the decapsulation oracle in Game 4 does, which is possible because $\mathcal{B}_{\mathtt{out2}}$ possesses $sk_{\mathtt{in}}$ and has access to the decryption oracle $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, \cdot)$.

When $\mathcal{A}$ terminates, $\mathcal{B}_{\mathtt{out2}}$ checks whether $\mathcal{A}$ has submitted a decapsulation query c satisfying the conditions of an outer-dangerous query, i.e. $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$, $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, d^*, \perp\}$, $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c_{\mathtt{in}}) = (r \| K) \neq \perp$, and $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c_{\mathtt{in}}; r) = c$. (Note that $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c)$ can be performed by $\mathcal{B}_{\mathtt{out2}}$'s decryption oracle.) If such a query c is found, then $\mathcal{B}_{\mathtt{out2}}$ sets $M' \leftarrow c_{\mathtt{in}}$ and $R' \leftarrow r$, and terminates with output (M', R'). Otherwise, $\mathcal{B}_{\mathtt{out2}}$ gives up and aborts.

The above completes the description of $\mathcal{B}_{\mathtt{out}}$. Let $\mathsf{Out}_{\mathcal{B}}$ be the event that $\mathcal{A}$ submits an outer-dangerous decapsulation query in the experiment simulated by $\mathcal{B}_{\mathtt{out}}$.

It is easy to see that $\mathcal{B}_{\mathtt{out}}$ simulates Game 4 perfectly for $\mathcal{A}$. Therefore, the probability that $\mathsf{Out}_{\mathcal{B}}$ occurs in the experiment simulated by $\mathcal{B}_{\mathtt{out}}$ is identical to the probability that Out_4 occurs in Game 4, namely we have $\Pr[\mathsf{Out}_{\mathcal{B}}] = \Pr[\mathsf{Out}_4]$. Furthermore, whenever $\mathsf{Out}_{\mathcal{B}}$ occurs, $\mathcal{B}_{\mathtt{out2}}$'s output $(M', R') = (c_{\mathtt{in}}, r)$ satisfies the condition:

$$\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, M'; R')) = 1,$$

which is exactly the condition of violating the randomness-inextractability of $\Pi_{\mathtt{out}}$. Therefore, we have

$$\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi_{\mathtt{out}}, \mathcal{B}_{\mathtt{out}}}(k) = \Pr[\mathsf{Out}_{\mathcal{B}}] = \Pr[\mathsf{Out}_4].$$

Then, by Equation (5), we have $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi_{\mathtt{out}}, \mathcal{B}_{\mathtt{out}}}(k) \geq |\Pr[\mathsf{In}_3] - \Pr[\mathsf{In}_4]|$, as required. □ (**Claim 5**)

Claim 6 *There exists a PPTA* $\mathcal{B}'_{\mathtt{out}}$ *such that* $\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{out}}, \mathcal{B}'_{\mathtt{out}}}(k) = (1/Q) \cdot |\sum_{j \in [Q]} (\Pr[\mathsf{In}^{(j)}_4] - \Pr[\mathsf{In}^{(j)}_5])|$.

Proof of Claim 6. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}'_{\mathtt{out}}$ that attacks the `wRNM-DCCA` security of the detectable PKE scheme $\Pi_{\mathtt{out}}$ with the claimed advantage. The description of $\mathcal{B}'_{\mathtt{out}} = (\mathcal{B}'_{\mathtt{out1}}, \mathcal{B}'_{\mathtt{out2}}, \mathcal{B}'_{\mathtt{out3}})$ is as follows:

$\mathcal{B}'^{\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, \cdot)}_{\mathtt{out1}}(pk_{\mathtt{out}})$**:** $\mathcal{B}'_{\mathtt{out1}}$ first executes $(pk_{\mathtt{in}}, sk_{\mathtt{in}}) \leftarrow \mathsf{KKG}_{\mathtt{in}}(1^k)$ and $(c^*_{\mathtt{in}}, \alpha^*) \leftarrow \mathsf{Encap}_{\mathtt{in}}(pk_{\mathtt{in}})$, and then picks $d^* \in \{0,1\}^n$ uniformly at random. Then $\mathcal{B}'_{\mathtt{out1}}$ sets $M_0 \leftarrow d^*$, $M_1 \leftarrow c^*_{\mathtt{in}}$, and $\mathsf{st}_{\mathcal{B}} \leftarrow (\mathcal{B}'_{\mathtt{out1}}$'s entire view), and terminates with output $(M_0, M_1, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}'^{\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, \cdot)}_{\mathtt{out2}}(\mathsf{st}_{\mathcal{B}}, c^*)$**:** $\mathcal{B}'_{\mathtt{out2}}$ first picks $K^* \in \{0,1\}^k$ uniformly at random, sets $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathtt{out}})$, and runs $\mathcal{A}(PK, c^*, K^*)$.

$\mathcal{B}'_{\mathtt{out2}}$ answers $\mathcal{A}$'s decapsulation queries c as the decapsulation oracle in Game 4 does, which is possible because $\mathcal{B}'_{\mathtt{out2}}$ possesses $sk_{\mathtt{in}}$ and has access to the decryption oracle $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, \cdot)$. (Note that if $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$, then $\mathcal{B}'_{\mathtt{out2}}$ can immediately return $\perp$ to $\mathcal{A}$, which is the correct behavior of the decapsulation oracle in Game 4 (and Game 5), and thus need not use the oracle.)

When $\mathcal{A}$ terminates, $\mathcal{B}'_{\mathtt{out2}}$ picks $u \in [Q]$ uniformly at random. Let $c^{(u)}$ be the u-th decapsulation query submitted by $\mathcal{A}$. $\mathcal{B}'_{\mathtt{out2}}$ sets $\mathsf{st}'_{\mathcal{B}} \leftarrow (\mathcal{B}'_{\mathtt{out2}}$'s entire view), and terminates with output $(c^{(u)}, \mathsf{st}'_{\mathcal{B}})$. (Note that it could be the case that $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c^{(u)}) = 1$ holds.)

$\mathcal{B}'_{\mathtt{out3}}(\mathsf{st}'_{\mathcal{B}}, M')$**:** (where $M' = \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(u)})$ or $M' = \mathsf{same}$) If $M' \in \{\mathsf{same}, \perp\}$, then $\mathcal{B}'_{\mathtt{out3}}$ sets $\gamma' \leftarrow 0$. Otherwise, $\mathcal{B}'_{\mathtt{out3}}$ interprets M' as a ciphertext $c'_{\mathtt{in}}$ of the inner detectable KEM $\Gamma_{\mathtt{in}}$, and computes $\gamma' \leftarrow \mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c'_{\mathtt{in}})$. Finally, $\mathcal{B}'_{\mathtt{out3}}$ terminates with output γ'.

The above completes the description of $\mathcal{B}'_{\mathtt{out}}$. Note that $\mathcal{B}'_{\mathtt{out2}}$ never submits a prohibited decryption query c satisfying $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ to $\mathcal{B}'_{\mathtt{out2}}$'s own decryption oracle. Furthermore, since $\mathcal{A}$'s queries c always satisfy $c \neq c^*$, it is guaranteed that the ciphertext $c^{(u)}$ finally output by $\mathcal{B}'_{\mathtt{out2}}$ satisfies $c^{(u)} \neq c^*$.

Let $\gamma \in \{0,1\}$ be $\mathcal{B}'_{\mathtt{out}}$'s challenge bit, and for $j \in [Q]$, let $\mathsf{In}_{\mathcal{B}}^{(j)}$ be the event that $\mathcal{A}$'s j-th decapsulation query is an inner-dangerous query in the experiment simulated by $\mathcal{B}'_{\mathtt{out}}$. Notice that $\mathcal{B}'_{\mathtt{out3}}$ outputs 1 only when $\mathsf{In}_{\mathcal{B}}^{(u)}$ occurs. In particular, if $M' \in \{\mathsf{same}, \bot\}$, then it implies that $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(u)}) \in \{c^*_{\mathtt{in}}, d^*, \bot\}$ and thus $\mathsf{In}_{\mathcal{B}}^{(u)}$ has not occurred, in which case $\mathcal{B}'_{\mathtt{out3}}$ correspondingly outputs 0.

It is not hard to see that if $\gamma = 1$ (resp. $\gamma = 0$), then $\mathcal{B}'_{\mathtt{out}}$ simulates Game 4 (resp. Game 5) perfectly for $\mathcal{A}$. Specifically, c^* is an encryption of $c^*_{\mathtt{in}}$ if $\gamma = 1$, and is an encryption of the "dummy" plaintext $d^* \in \{0,1\}^n$ if $\gamma = 0$, which is exactly how $\mathcal{A}$'s challenge ciphertext in Game 4 and that in Game 5 are computed, respectively. Furthermore, $\mathcal{B}'_{\mathtt{out2}}$ answers $\mathcal{A}$'s decapsulation queries exactly as the decapsulation oracle in Game 4 (and Game 5) does, by using $\mathcal{B}'_{\mathtt{out2}}$'s decryption oracle, $\mathsf{F}_{\mathtt{out}}$ (which is publicly computable), and $sk_{\mathtt{in}}$, $c^*_{\mathtt{in}}$, and d^* (that are all known to $\mathcal{B}'_{\mathtt{out}}$). Note also that $u \in [Q]$ is chosen uniformly and independently of γ and $\mathcal{A}$'s behavior. These imply that for all $j \in [Q]$, we have $\Pr[\mathsf{In}_{\mathcal{B}}^{(u)} | u = j \wedge \gamma = 1] = \Pr[\mathsf{In}_4^{(j)}]$, $\Pr[\mathsf{In}_{\mathcal{B}}^{(u)} | u = j \wedge \gamma = 0] = \Pr[\mathsf{In}_5^{(j)}]$, and $\Pr[u = j | \gamma = 1] = \Pr[u = j | \gamma = 0] = 1/Q$.

Using the above, $\mathcal{B}'_{\mathtt{out}}$'s $\mathtt{wRNM\text{-}DCCA}$ advantage can be calculated as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{out}}, \mathcal{B}'_{\mathtt{out}}}(k) &= 2 \cdot \left| \Pr[\gamma' = \gamma] - \frac{1}{2} \right| = \Big| \Pr[\gamma' = 1 | \gamma = 1] - \Pr[\gamma' = 1 | \gamma = 0] \Big| \\
&= \Big| \Pr[\mathsf{In}_{\mathcal{B}}^{(u)} | \gamma = 1] - \Pr[\mathsf{In}_{\mathcal{B}}^{(u)} | \gamma = 0] \Big| \\
&= \Big| \sum_{j \in [Q]} \Big(\Pr[\mathsf{In}_{\mathcal{B}}^{(u)} | u = j \wedge \gamma = 1] \cdot \Pr[u = j | \gamma = 1] \\
&\qquad\qquad - \Pr[\mathsf{In}_{\mathcal{B}}^{(u)} | u = j \wedge \gamma = 0] \cdot \Pr[u = j | \gamma = 0] \Big) \Big| \\
&= \frac{1}{Q} \Big| \sum_{j \in [Q]} (\Pr[\mathsf{In}_4^{(j)}] - \Pr[\mathsf{In}_5^{(j)}]) \Big|.
\end{aligned}$$

□ (**Claim 6**)

Claim 7 *There exists a PPTA* $\mathcal{B}'_{\mathtt{in}}$ *such that* $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{in}}, \mathcal{B}'_{\mathtt{in}}}(k) \geq (1/Q^2) \cdot \sum_{j \in [Q]} \Pr[\mathsf{In}_5^{(j)}]$.

Proof of Claim 7. The proof of this claim is done by showing an adversary (reduction algorithm) $\mathcal{B}'_{\mathtt{in}}$ against the unpredictability of the detectable KEM $\Gamma_{\mathtt{in}}$, such that $\mathcal{B}'_{\mathtt{in}}$ simulates Game 5 for $\mathcal{A}$, picks one of $\mathcal{A}$'s decapsulation queries c randomly, uses its decryption result $c_{\mathtt{in}} = \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c)$ as $\mathcal{B}'_{\mathtt{in}}$'s final output in the unpredictability experiment, and hopes that the picked query was inner-dangerous (and satisfies $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$, where $c^*_{\mathtt{in}}$ is $\mathcal{B}'_{\mathtt{in}}$'s challenge ciphertext in the unpredictability experiment).

The reader may expect that such $\mathcal{B}'_{\mathtt{in}}$ can perfectly simulate Game 5 for $\mathcal{A}$ and should have unpredictability advantage $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{in}}, \mathcal{B}'_{\mathtt{in}}}(k) \geq (1/Q) \cdot \sum_{j \in [Q]} \Pr[\mathsf{In}^{(j)}]$ (and might wonder why the "loss" that we show in this claim is $1/Q^2$). However, we could not come up with such a proof, due to a technical subtlety that the behavior of the decapsulation oracle in Game 5 is dependent on $c^*_{\mathtt{in}}$, while the unpredictability adversary $\mathcal{B}'_{\mathtt{in}}$ cannot see its challenge ciphertext $c^*_{\mathtt{in}}$ during the unpredictability experiment.

Nonetheless, we show that such $\mathcal{B}'_{\mathtt{in}}$ can simulate Game 5 for $\mathcal{A}$ "almost" perfectly, and provide an analysis that guarantees that our $\mathcal{B}'_{\mathtt{in}}$ has at least the claimed unpredictability advantage. The description of $\mathcal{B}'_{\mathtt{in}}$ is as follows:

$\mathcal{B}'^{\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}},\cdot)}_{\mathtt{in}}(pk_{\mathtt{in}})$**:** $\mathcal{B}'_{\mathtt{in}}$ first picks $d^* \in \{0,1\}^n$ uniformly at random, and runs $(pk_{\mathtt{out}}, sk_{\mathtt{out}}) \leftarrow \mathsf{PKG}_{\mathtt{out}}(1^k)$ and $c^* \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, d^*)$. Next, $\mathcal{B}'_{\mathtt{in}}$ picks $K^* \in \{0,1\}^k$ uniformly at random, sets $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathtt{out}})$, and then runs $\mathcal{A}(PK, c^*, K^*)$.

$\mathcal{B}'_{\mathtt{in}}$ answers decapsulation queries c from $\mathcal{A}$ as follows:

1. If $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$, then return $\bot$ to $\mathcal{A}$.
2. Otherwise, compute $c_{\mathtt{in}} \leftarrow \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c)$ and return $\bot$ to $\mathcal{A}$ if $c_{\mathtt{in}} \in \{d^*, \bot\}$.[4]
3. Otherwise, submit $c_{\mathtt{in}}$ to $\mathcal{B}'_{\mathtt{in}}$'s decapsulation oracle, and receive the result α. If $\alpha = (r \| K) \neq \bot$ and $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}} c_{\mathtt{in}}; r) = c$ then return K, otherwise return $\bot$, to $\mathcal{A}$.

When $\mathcal{A}$ terminates, $\mathcal{B}'_{\mathtt{in}}$ picks $u \in [Q]$ uniformly at random, and proceeds as follows: Let $c^{(u)}$ be the u-th query submitted by $\mathcal{A}$. If $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(u)}) = c^{(u)}_{\mathtt{in}} \neq \bot$ holds, then $\mathcal{B}'_{\mathtt{in}}$ terminates with output $c^{(u)}_{\mathtt{in}}$. Otherwise, $\mathcal{B}'_{\mathtt{in}}$ gives up and aborts.

The above completes the description of $\mathcal{B}'_{\mathtt{in}}$.

Let $c^*_{\mathtt{in}}$ be the ciphertext generated by $\mathcal{B}'_{\mathtt{in}}$'s unpredictability experiment. Since the challenge ciphertext $c^*_{\mathtt{in}}$ is information-theoretically hidden from $\mathcal{B}'_{\mathtt{in}}$ in the unpredictability experiment, we can assume without loss of generality that $c^*_{\mathtt{in}}$ is generated at the beginning of the experiment. In order to analyze the unpredictability advantage of $\mathcal{B}'_{\mathtt{in}}$, we introduce several definitions of the events in the experiment simulated by $\mathcal{B}'_{\mathtt{in}}$: (Here, the subscript "$\mathcal{B}$" of each event indicates that it is an event defined in the experiment simulated by $\mathcal{B}'_{\mathtt{in}}$.)

$\mathsf{In}^{(j)}_{\mathcal{B}}$ **where** $j \in [Q]$**:** $\mathcal{A}$'s j-th query $c^{(j)}$ is *inner-dangerous with respect to* $c^*_{\mathtt{in}}$. Namely, it holds that $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(j)}) = c^{(j)}_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, d^*, \bot\}$ and $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c^{(j)}_{\mathtt{in}}) = 1$.

$\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}}$ **where** $j \in [Q]$**:** $\mathcal{A}$'s j-th query $c^{(j)}$ satisfies $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(j)}) = c^{(j)}_{\mathtt{in}} \neq \bot$ and $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c^{(j)}_{\mathtt{in}}) = 1$.

$\mathsf{C}_{\mathcal{B}}$**:** $\mathcal{A}$ submits at least one decapsulation query c satisfying $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c^*_{\mathtt{in}}$.

$\mathsf{C}^{(j)}_{\mathcal{B}}$ **where** $j \in [Q]$**:** $\mathcal{A}$'s j-th query $c^{(j)}$ satisfies $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(j)}) = c^*_{\mathtt{in}}$.

Firstly, by definition of the unpredictability experiment and our design of $\mathcal{B}'_{\mathtt{in}}$, we have $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{in}}, \mathcal{B}'_{\mathtt{in}}}(k) = \Pr[\widehat{\mathsf{In}}^{(u)}_{\mathcal{B}}]$. Note also that the index $u \in [Q]$ is chosen uniformly, independently of $\mathcal{A}$'s behavior. Hence, for every $j \in [Q]$, we have $\Pr[u = j] = 1/Q$.

Next, by the definitions of the events $\mathsf{In}^{(j)}_{\mathcal{B}}$, $\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}}$, and $\mathsf{C}^{(j)}_{\mathcal{B}}$, the event $\mathsf{In}^{(j)}_{\mathcal{B}}$ implies $\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}} \wedge \overline{\mathsf{C}^{(j)}_{\mathcal{B}}}$, and hence for all $j \in [Q]$, we have $\Pr[\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}} \wedge \overline{\mathsf{C}^{(j)}_{\mathcal{B}}}] \geq \Pr[\mathsf{In}^{(j)}_{\mathcal{B}}]$. Note also that the event $\mathsf{C}^{(j)}_{\mathcal{B}}$ implies $\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}}$, because $c^{(j)}_{\mathtt{in}} = c^*_{\mathtt{in}}$ implies $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c^{(j)}_{\mathtt{in}}) = 1$.[5] Hence, for all $j \in [Q]$, we have $\Pr[\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}} \wedge \mathsf{C}^{(j)}_{\mathcal{B}}] = \Pr[\mathsf{C}^{(j)}_{\mathcal{B}}]$. Using these, for every $j \in [Q]$, we have

$$\Pr[\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}}] = \Pr[\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}} \wedge \overline{\mathsf{C}^{(j)}_{\mathcal{B}}}] + \Pr[\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}} \wedge \mathsf{C}^{(j)}_{\mathcal{B}}] \geq \Pr[\mathsf{In}^{(j)}_{\mathcal{B}}] + \Pr[\mathsf{C}^{(j)}_{\mathcal{B}}]. \tag{6}$$

Finally, note that the distribution of the inner ciphertext $c^*_{\mathtt{in}}$ in Game 5 and the distribution of $\mathcal{B}'_{\mathtt{in}}$'s challenge ciphertext in the unpredictability experiment are identical. We therefore analyze the

[4] Here, if $\mathcal{B}'_{\mathtt{in}}$ could check whether $c_{\mathtt{in}} \in \{c^*_{\mathtt{in}}, d^*, \bot\}$ holds, then $\mathcal{B}'_{\mathtt{in}}$'s simulation of the decapsulation oracle in Game 5 was perfect.

[5] Recall that we require $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c, c) = 1$ as a functional requirement of a detectable PKE scheme.

unpredictability advantage of $\mathcal{B}'_{\mathtt{in}}$ by understanding that $\mathcal{B}'_{\mathtt{in}}$ simulates Game 5 for $\mathcal{A}$ as if the inner ciphertext $c^*_{\mathtt{in}}$ for $\mathcal{A}$ in Game 5 is $\mathcal{B}'_{\mathtt{in}}$'s challenge ciphertext generated in $\mathcal{B}'_{\mathtt{in}}$'s unpredictability experiment (and the notion of an inner-dangerous query is with respect to this $c^*_{\mathtt{in}}$). Seeing in this way, it is not hard to see that $\mathcal{B}'_{\mathtt{in}}$ perfectly simulates Game 5 for $\mathcal{A}$ *until the point* $\mathcal{A}$ *submits a decapsulation query that causes the event* $\mathsf{C}_{\mathcal{B}}$. Therefore, letting C_5 be the event that $\mathcal{A}$ submits at least one decapsulation query c such that $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c^*_{\mathtt{in}}$ in Game 5, we have

$$\forall j \in [Q] \ : \ \Pr[\mathsf{In}^{(j)}_{\mathcal{B}} \wedge \overline{\mathsf{C}_{\mathcal{B}}}] = \Pr[\mathsf{In}^{(j)}_5 \wedge \overline{\mathsf{C}_5}] \qquad \text{and} \qquad \Pr[\mathsf{C}_{\mathcal{B}}] = \Pr[\mathsf{C}_5]. \tag{7}$$

Armed with these, $\mathcal{B}'_{\mathtt{in}}$'s unpredictability advantage can be estimated as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{in}},\mathcal{B}'_{\mathtt{in}}}(k) &= \Pr[\widehat{\mathsf{In}}^{(u)}_{\mathcal{B}}] = \sum_{j\in[Q]} \Pr[\widehat{\mathsf{In}}^{(u)}_{\mathcal{B}} | u = j] \cdot \Pr[u = j] = \frac{1}{Q} \sum_{j\in[Q]} \Pr[\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}}] \\
&\geq \frac{1}{Q} \sum_{j\in[Q]} \Big(\Pr[\mathsf{In}^{(j)}_{\mathcal{B}}] + \Pr[\mathsf{C}^{(j)}_{\mathcal{B}}] \Big) && \text{(by Equation (6))} \\
&\geq \frac{1}{Q} \sum_{j\in[Q]} \Pr[\mathsf{In}^{(j)}_{\mathcal{B}}] + \frac{1}{Q} \Pr[\mathsf{C}_{\mathcal{B}}] && \text{(by the union bound)} \\
&\geq \frac{1}{Q} \sum_{j\in[Q]} \Pr[\mathsf{In}^{(j)}_{\mathcal{B}} \wedge \overline{\mathsf{C}_{\mathcal{B}}}] + \frac{1}{Q} \Pr[\mathsf{C}_{\mathcal{B}}] \\
&= \frac{1}{Q} \sum_{j\in[Q]} \Pr[\mathsf{In}^{(j)}_5 \wedge \overline{\mathsf{C}_5}] + \frac{1}{Q} \Pr[\mathsf{C}_5] && \text{(by Equation (7))} \\
&\geq \frac{1}{Q^2} \sum_{j\in[Q]} \Pr[\mathsf{In}^{(j)}_5 \wedge \overline{\mathsf{C}_5}] + \frac{1}{Q} \Pr[\mathsf{C}_5] && \text{(multiply the first term by } \frac{1}{Q} \leq 1) \\
&= \frac{1}{Q^2} \sum_{j\in[Q]} \Big(\Pr[\mathsf{In}^{(j)}_5 \wedge \overline{\mathsf{C}_5}] + \Pr[\mathsf{C}_5] \Big) \\
&\geq \frac{1}{Q^2} \sum_{j\in[Q]} \Big(\Pr[\mathsf{In}^{(j)}_5 \wedge \overline{\mathsf{C}_5}] + \Pr[\mathsf{In}^{(j)}_5 \wedge \mathsf{C}_5] \Big) = \frac{1}{Q^2} \sum_{j\in[Q]} \Pr[\mathsf{In}^{(j)}_5],
\end{aligned}$$

which implies the claim. □ (**Claim 7**)

Claims 1 to 7 guarantee that there exist PPTAs $\mathcal{B}_{\mathtt{in1}}$, $\mathcal{B}_{\mathtt{in2}}$, $\mathcal{B}_{\mathtt{out}}$, $\mathcal{B}'_{\mathtt{out}}$, and $\mathcal{B}'_{\mathtt{in}}$, satisfying Equation (3), where the second and the third terms in the right hand side of Equation (3) are derived from Claim 3 in which we set $(p, q) = (1, 1/2)$ and $(p, q) = (0, 1)$, respectively. Recall that the choice of $\mathcal{A}$ was arbitrarily, and thus for any PPTA $\mathcal{A}$, we can show a negligible upperbound on $\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k)$. Therefore, $\Gamma_{\mathtt{DL}}$ is $\mathtt{CCA}$ secure. □ (**Theorem 1**)

D.3 Proof of Theorem 2: Our Second Proof of CCA Security of $\Gamma_{\mathtt{DL}}$

Before going into the actual proof, we note that although the proof of Theorem 2 would seem structurally similar to that of Theorem 1, there are several subtle but crucial differences. Most importantly, the definitions of "inner/outer-dangerous queries" in this proof are different from those in the proof of Theorem 1, and correspondingly we consider a different ordering of the sequence of

Table 3. An overview of the games used in the proof of Theorem 2.

Game	c^*	r^* and K_1^*	$\mathcal{A}$'s decapsulation query c is answered with
1	$\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$	$(r^* \| K_1^*) = \alpha^*$	$\mathsf{Decap}_{\mathtt{DL}}(SK, c)$
2	$\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$	$(r^* \| K_1^*) = \alpha^*$	$\begin{cases} \bot & \text{if } \mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1 \\ \mathsf{Decap}_{\mathtt{DL}}(SK, c) & \text{otherwise} \end{cases}$
3	$\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$	$r^* \leftarrow \{0,1\}^\ell$ $K_1^* \leftarrow \{0,1\}^k$	$\begin{cases} \bot & \text{if } \mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1 \\ \mathsf{Decap}_{\mathtt{DL}}(SK, c) & \text{otherwise} \end{cases}$
4	$\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, 0^n; r^*)$	$r^* \leftarrow \{0,1\}^\ell$ $K_1^* \leftarrow \{0,1\}^k$	$\begin{cases} \bot & \text{if } \mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1 \\ \mathsf{Decap}_{\mathtt{DL}}(SK, c) & \text{otherwise} \end{cases}$

games for the proof. Furthermore, the role of the "non-malleability" of the assumption in this proof and that of the proof of Theorem 1 are different. Informally speaking, in the proof of Theorem 2, the wNM-DCCA security of the inner detectable KEM $\mathit{\Gamma}_{\mathtt{in}}$ is used to ensure that the probability that a CCA adversary comes up with an outer-dangerous query is not noticeably different between the games in which we invoke (the indistinguishability property of) the DCCA security of the KEM.

Now we proceed to the proof of Theorem 2. We will show that for any PPTA $\mathcal{A}$ that attacks the CCA security of the KEM $\mathit{\Gamma}_{\mathtt{DL}}$ and makes $Q = Q(k) > 0$ decapsulation queries, there exist PPTAs $\mathcal{B}_{\mathtt{in}}$, $\mathcal{B}'_{\mathtt{in1}}$, $\mathcal{B}'_{\mathtt{in2}}$, $\mathcal{B}_{\mathtt{out}}$, $\mathcal{B}'_{\mathtt{out}}$, and $\mathcal{B}''_{\mathtt{in}}$, such that

$$\begin{aligned}\mathsf{Adv}^{\mathtt{CCA}}_{\mathit{\Gamma}_{\mathtt{DL}},\mathcal{A}}(k) \le 2Q \cdot \mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\mathit{\Gamma}_{\mathtt{in}},\mathcal{B}_{\mathtt{in}}}(k) + 2 \cdot \mathsf{Adv}^{\mathtt{DCCA}}_{\mathit{\Gamma}_{\mathtt{in}},\mathcal{B}'_{\mathtt{in1}}}(k) + 3 \cdot \mathsf{Adv}^{\mathtt{DCCA}}_{\mathit{\Gamma}_{\mathtt{in}},\mathcal{B}'_{\mathtt{in2}}}(k) \\ + 2Q \cdot \mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Pi}_{\mathtt{out}},\mathcal{B}_{\mathtt{out}}}(k) + 3 \cdot \mathsf{Adv}^{\mathtt{DCCA}}_{\mathit{\Pi}_{\mathtt{out}},\mathcal{B}'_{\mathtt{out}}}(k) + 3Q \cdot \mathsf{Adv}^{\mathtt{UNP}}_{\mathit{\Gamma}_{\mathtt{in}},\mathcal{B}''_{\mathtt{in}}}(k), \quad (8)\end{aligned}$$

which, due to the assumptions on the building blocks $\mathit{\Gamma}_{\mathtt{in}}$ and $\mathit{\Pi}_{\mathtt{out}}$, implies that $\mathsf{Adv}^{\mathtt{CCA}}_{\mathit{\Gamma}_{\mathtt{DL}},\mathcal{A}}(k)$ is negligible, and hence proves the theorem.

Let $\mathcal{A}$ be any PPTA adversary that attacks the CCA security of the KEM $\mathit{\Gamma}_{\mathtt{DL}}$ and makes in total $Q > 0$ decapsulation queries. Consider the following sequence of games: (Here, the values with asterisk (*) represent those related to $\mathcal{A}$'s challenge ciphertext.)

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

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

Game 3: Same as Game 2, except that $r^* \in \{0,1\}^\ell$ and $K_1^* \in \{0,1\}^k$ are picked uniformly at random, independently of $c^*_{\mathtt{in}}$. More precisely, the steps "$(c^*_{\mathtt{in}}, \alpha^*) \leftarrow \mathsf{Encap}_{\mathtt{in}}(pk_{\mathtt{in}})$; Parse α^* as $(r^*, K_1^*) \in \{0,1\}^\ell \times \{0,1\}^k$" in Game 2 are replaced with the steps "$(c^*_{\mathtt{in}}, \alpha^*) \leftarrow \mathsf{Encap}_{\mathtt{in}}(pk_{\mathtt{in}})$; $r^* \leftarrow \{0,1\}^\ell$; $K_1^* \leftarrow \{0,1\}^k$," and we do not use α^* (corresponding to $c^*_{\mathtt{in}}$) at all.

Game 4: Same as Game 3, except that the information of $c^*_{\mathtt{in}}$ is erased from c^*. More precisely, the step "$c^* \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}})$" in Game 3 is replaced with the step "$c^* \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, 0^n)$."

The above completes the description of the games. Table 3 is an overview of the differences among the games in terms of how the challenge ciphertext c^* and the values (r^*, K_1^*) are generated, and how the decapsulation oracle works.

We say that a decapsulation query c submitted by $\mathcal{A}$ in the above games is *inner-dangerous* if c satisfies the following three conditions:

(1) $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 0$,

(2) $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \neq \bot$, and
(3) $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$.

Furthermore, we say that a decapsulation query c submitted by $\mathcal{A}$ in the above games is *outer-dangerous* if c satisfies the following four conditions:

(1) $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$,
(2) $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, \bot\}$,
(3) $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c_{\mathtt{in}}) = (r \| K) \neq \bot$, and
(4) $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c_{\mathtt{in}}; r) = c$.

For $i \in [4]$, we define the following events in Game i:

S_i**:** $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs).
In_i**:** $\mathcal{A}$ submits at least one inner-dangerous decapsulation query.
$\mathsf{In}_i^{(j)}$ **where** $j \in [Q]$**:** $\mathcal{A}$'s j-th decapsulation query is inner-dangerous.
Out_i**:** $\mathcal{A}$ submits at least one outer-dangerous decapsulation query.
$\mathsf{Out}_i^{(j)}$ **where** $j \in [Q]$**:** $\mathcal{A}$'s j-th decapsulation query is outer-dangerous.

By definitions of the games and events, we can show the following upperbound on $\mathcal{A}$'s $\mathtt{CCA}$ advantage $\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k)$:

Claim 8 *$\mathcal{A}$'s $\mathtt{CCA}$ advantage $\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k)$ can be upperbounded as follows:*

$$\begin{aligned}\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k) \leq 2 \cdot \Big| \sum_{j \in [Q]} \Big(\Pr[\mathsf{Out}_2^{(j)} \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{Out}_3^{(j)} \wedge \overline{\mathsf{In}_3}] \Big) \Big| + 2 \cdot \sum_{j \in [Q]} \Pr[\mathsf{Out}_3^{(j)}] \\ + 2 \cdot \Big| \Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] + \frac{1}{2}(\Pr[\mathsf{In}_2] - \Pr[\mathsf{In}_3]) \Big| + 2 \cdot \Big| \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] + \frac{1}{2}\Pr[\mathsf{In}_3] - \frac{1}{2} \Big| \\ + 3 \cdot \sum_{i \in \{2,3\}} \Big| \Pr[\mathsf{In}_i] - \Pr[\mathsf{In}_{i+1}] \Big| + 3 \cdot \sum_{j \in [Q]} \Pr[\mathsf{In}_4^{(j)}]. \quad (9)\end{aligned}$$

Proof of Claim 8. By the triangle inequality, we have

$$\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k) = 2 \cdot \Big| \Pr[\mathsf{S}_1] - \frac{1}{2} \Big| \leq 2 \cdot \Big| \Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2] \Big| + 2 \cdot \Big| \Pr[\mathsf{S}_2] - \frac{1}{2} \Big|.$$

Regarding the first term $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]|$, notice that the difference between Game 1 and Game 2 is in how $\mathcal{A}$'s decapsulation queries c satisfying the conditions $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ and $\mathsf{Decap}_{\mathtt{DL}}(SK, c) \neq \bot$ is answered. (Such a query is answered with $\bot$ in Game 2, while it s answered with non-$\bot$ in Game 1.) We first show that if c additionally satisfies $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c^*_{\mathtt{in}}$, then c is answered with $\bot$ in Game 1 as well. To see this, recall that according to the validity check in $\mathsf{Decap}_{\mathtt{DL}}$, in order for a query c with $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c^*_{\mathtt{in}}$ to satisfy $\mathsf{Decap}_{\mathtt{DL}}(SK, c) \neq \bot$, it must hold that $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c^*_{\mathtt{in}}) = (r^* \| K^*_1)$ and $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*) = c$. However, the only ciphertext c satisfying the last condition is the challenge ciphertext c^* itself, but according to the rule of the $\mathtt{CCA}$ experiment, any of $\mathcal{A}$'s decapsulation query c must satisfy $c \neq c^*$. Therefore, any decapsulation query c satisfying $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c^*_{\mathtt{in}}$ is answered with $\bot$ in Game 1 as well. (In fact, this is true regardless of whether $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ holds.)

Hence, an actual difference between Game 1 and Game 2 occurs only if $\mathcal{A}$ submits a decapsulation query c satisfying $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$, $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} \neq c^*_{\mathtt{in}}$, and $\mathsf{Decap}_{\mathtt{DL}}(SK, c) \neq$

$\bot$. The last condition in particular implies $c_{\mathtt{in}} \neq \bot$, $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c_{\mathtt{in}}) = (r \| K) \neq \bot$, and $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c_{\mathtt{in}}; r) = c$, which, combined with the condition $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$, are exactly the conditions of outer-dangerous queries. This implies that the difference between $\Pr[\mathsf{S}_1]$ and $\Pr[\mathsf{S}_2]$ is upperbounded by the probability that $\mathcal{A}$ submits at least one outer-dangerous decapsulation query, i.e., we have

$$|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]| \leq \Pr[\mathsf{Out}_1] = \Pr[\mathsf{Out}_2].$$

We further proceed to estimating the upperbound of $\Pr[\mathsf{Out}_2]$ based on whether In_2 occurs, namely,

$$\Pr[\mathsf{Out}_2] = \Pr[\mathsf{Out}_2 \wedge \overline{\mathsf{In}_2}] + \Pr[\mathsf{Out}_2 \wedge \mathsf{In}_2] \leq \Pr[\mathsf{Out}_2 \wedge \overline{\mathsf{In}_2}] + \Pr[\mathsf{In}_2].$$

Note that by the definition of the events, for every $i \in [4]$ we have $\Pr[\mathsf{Out}_i \wedge \overline{\mathsf{In}_i}] = \Pr[\bigvee_{j \in [Q]}(\mathsf{Out}_i^{(j)} \wedge \overline{\mathsf{In}_i})]$, and for every $j \in [Q]$ we have $\Pr[\mathsf{Out}_i^{(j)} \wedge \overline{\mathsf{In}_i}] \leq \Pr[\mathsf{Out}_i^{(j)}]$. Using these and applying the union bound and the triangle inequality, we obtain the following upperbound of $\Pr[\mathsf{Out}_2 \wedge \overline{\mathsf{In}_2}]$:

$$\Pr[\mathsf{Out}_2 \wedge \overline{\mathsf{In}_2}] \leq \sum_{j \in [Q]} \Pr[\mathsf{Out}_2^{(j)} \wedge \overline{\mathsf{In}_2}] \leq \Big| \sum_{j \in [Q]} \Big(\Pr[\mathsf{Out}_2^{(j)} \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{Out}_3^{(j)} \wedge \overline{\mathsf{In}_3}] \Big) \Big| + \sum_{j \in [Q]} \Pr[\mathsf{Out}_3^{(j)}].$$

Regarding the term $|\Pr[\mathsf{S}_2] - 1/2|$, by Fact 1 (stated in Appendix A.4) and the triangle inequality, we have

$$\begin{aligned} \left| \Pr[\mathsf{S}_2] - \frac{1}{2} \right| &\leq \left| \Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}] + \frac{1}{2}\Pr[\mathsf{In}_2] - \frac{1}{2} \right| + \frac{1}{2}\Pr[\mathsf{In}_2] \\ &\leq \left| \Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] + \frac{1}{2}(\Pr[\mathsf{In}_2] - \Pr[\mathsf{In}_3]) \right| \\ &\quad + \left| \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] + \frac{1}{2}\Pr[\mathsf{In}_3] - \frac{1}{2} \right| + \frac{1}{2}\Pr[\mathsf{In}_2]. \end{aligned}$$

Finally, noting that $\Pr[\mathsf{In}_4] = \Pr[\bigvee_{j \in [Q]} \mathsf{In}_4^{(j)}]$, and applying the union bound and the triangle inequality, we can upperbound $\Pr[\mathsf{In}_2]$ as follows:

$$\Pr[\mathsf{In}_2] \leq \sum_{j \in \{2,3\}} \Big| \Pr[\mathsf{In}_i] - \Pr[\mathsf{In}_{i+1}] \Big| + \Pr[\mathsf{In}_4] \leq \sum_{i \in \{2,3\}} \Big| \Pr[\mathsf{In}_i] - \Pr[\mathsf{In}_{i+1}] \Big| + \sum_{j \in [Q]} \Pr[\mathsf{In}_4^{(j)}].$$

Combining all the inequalities yields the claim. (Notice that in total the coefficient of $\Pr[\mathsf{In}_2]$, and hence the coefficients of the last two terms of the right hand side of Equation (9), become 3.) □ (**Claim 8**)

In the following, we show upperbounds of the terms that appear in the right hand side of Equation (9).

Claim 9 *There exists a PPTA* $\mathcal{B}_{\mathtt{in}}$ *such that* $\mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\Gamma_{\mathtt{in}}, \mathcal{B}_{\mathtt{in}}}(k) = (1/Q) \cdot |\sum_{j \in [Q]}(\Pr[\mathsf{Out}_2^{(j)} \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{Out}_3^{(j)} \wedge \overline{\mathsf{In}_3}])|$.

Proof of Claim 9. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}_{\mathtt{in}}$ that attacks the $\mathtt{wNM\text{-}DCCA}$ security of the detectable KEM $\Gamma_{\mathtt{in}}$ with the claimed advantage. The description of $\mathcal{B}_{\mathtt{in}} = (\mathcal{B}_{\mathtt{in}1}, \mathcal{B}_{\mathtt{in}2})$ is as follows:

$\mathcal{B}_{\mathtt{in1}}^{\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}},\cdot)}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, \alpha^*_\gamma)$: (where $\gamma \in \{0,1\}$ is $\mathcal{B}_{\mathtt{in}}$'s challenge bit in the $\mathtt{wNM\text{-}DCCA}$ experiment) $\mathcal{B}_{\mathtt{in1}}$ first picks $u \in [Q]$ uniformly at random. Next, $\mathcal{B}_{\mathtt{in1}}$ parses α^*_γ as $(r^*, K^*_1) \in \{0,1\}^\ell \times \{0,1\}^k$, picks $K^*_0 \in \{0,1\}^k$ and $b \in \{0,1\}$ uniformly at random, and then runs $(pk_{\mathtt{out}}, sk_{\mathtt{out}}) \leftarrow \mathsf{PKG}_{\mathtt{out}}(1^k)$ and $c^* \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$. Then $\mathcal{B}_{\mathtt{in1}}$ sets $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathtt{out}})$, and runs $\mathcal{A}(PK, c^*, K^*_b)$.

$\mathcal{B}_{\mathtt{in1}}$ answers $\mathcal{A}$'s decapsulation queries c as follows:

1. If $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ or $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c) = c_{\mathtt{in}} = \bot$, then $\mathcal{B}_{\mathtt{in1}}$ returns $\bot$ to $\mathcal{A}$.
2. Otherwise (i.e. $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 0$ and $c_{\mathtt{in}} \neq \bot$), if $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$, then $\mathcal{B}_{\mathtt{in1}}$ has detected that this query is inner-dangerous, in which case $\mathcal{B}_{\mathtt{in1}}$ gives up, prepares the state information $\mathsf{st}_{\mathcal{B}}$ that tells $\mathcal{B}_{\mathtt{in2}}$ that $\mathcal{B}_{\mathtt{in1}}$ has given up, and terminates with output $(\bot, \mathsf{st}_{\mathcal{B}})$.
3. Otherwise (i.e. $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 0$), $\mathcal{B}_{\mathtt{in}}$ submits $c_{\mathtt{in}}$ to its own decapsulation oracle, receives the result α, performs the remaining procedure of $\mathsf{Decap}_{\mathtt{DL}}(SK, c)$, and returns the result to $\mathcal{A}$.

When $\mathcal{A}$ terminates and at this point $\mathcal{B}_{\mathtt{in1}}$ has not given up (i.e. $\mathcal{A}$ has not made an inner-dangerous query), $\mathcal{B}_{\mathtt{in1}}$ proceeds as follows: Let $c^{(u)}$ be the u-th decapsulation query submitted by $\mathcal{A}$, and let $c^{(u)}_{\mathtt{in}} = \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(u)})$ be the inner ciphertext corresponding to $c^{(u)}$ (which $\mathcal{B}_{\mathtt{in1}}$ can compute because it possesses $sk_{\mathtt{out}}$). If $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c^{(u)}) = 0$ or $c^{(u)}_{\mathtt{in}} \in \{c^*_{\mathtt{in}}, \bot\}$, then $\mathcal{B}_{\mathtt{in1}}$ gives up, and does the same procedure for "give up" as above. Otherwise (i.e. $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c^{(u)}) = 1$ and $c^{(u)}_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, \bot\}$), $\mathcal{B}_{\mathtt{in1}}$ sets $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathtt{in1}}$'s entire view), and terminates with output $(c^{(u)}_{\mathtt{in}}, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathtt{in2}}(\mathsf{st}_{\mathcal{B}}, \alpha')$: (where $\alpha' = \mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c^{(u)}_{\mathtt{in}})$) $\mathcal{B}_{\mathtt{in2}}$ first checks if $\mathcal{B}_{\mathtt{in1}}$ gave up (by looking at $\mathsf{st}_{\mathcal{B}}$) or $\alpha' = \bot$ holds, and sets $\gamma' \leftarrow 0$ if this is the case. Otherwise (i.e. $\mathcal{B}_{\mathtt{in1}}$ did not give up and $\alpha' \neq \bot$), $\mathcal{B}_{\mathtt{in2}}$ parses α' as $(r', K') \in \{0,1\}^\ell \times \{0,1\}^k$, and then sets $\gamma' \leftarrow (\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^{(u)}_{\mathtt{in}}; r') \overset{?}{=} c^{(u)})$. Finally, $\mathcal{B}_{\mathtt{in2}}$ terminates with output γ'.

The above completes the description of $\mathcal{B}_{\mathtt{in}}$. Note that $\mathcal{B}_{\mathtt{in1}}$ never submits a prohibited decapsulation query $c_{\mathtt{in}}$ satisfying $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$ to its own decapsulation oracle. Furthermore, due to our design of $\mathcal{B}_{\mathtt{in}}$, it is guaranteed that the ciphertext that is finally output by $\mathcal{B}_{\mathtt{in1}}$ is different from $c^*_{\mathtt{in}}$.

Let $\mathsf{In}_{\mathcal{B}}$ be the event that $\mathcal{A}$ submits an inner-dangerous decapsulation query in the experiment simulated by $\mathcal{B}$, and for $j \in [Q]$, let $\mathsf{Out}^{(j)}_{\mathcal{B}}$ be the event that $\mathcal{A}$'s j-th decapsulation query $c^{(j)}$ is outer-dangerous, namely, $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c^{(j)}) = 1$, $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(j)}) = c^{(j)}_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, \bot\}$, $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c^{(j)}_{\mathtt{in}}) = (r^{(j)} \| K^{(j)}) \neq \bot$, and $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^{(j)}_{\mathtt{in}}; r^{(j)}) = c^{(j)}$.

Note that $\mathcal{B}_{\mathtt{in2}}$ outputs 1 only if $\mathcal{A}$ does not submit any inner-dangerous query (i.e. $\overline{\mathsf{In}_{\mathcal{B}}}$ occurs) *and* $\mathsf{Out}^{(u)}_{\mathcal{B}}$ occurs. Note also that $u \in [Q]$ is chosen uniformly at random, independently of $\mathcal{B}_{\mathtt{in}}$'s challenge bit and $\mathcal{A}$'s behavior. Hence, for every $j \in [Q]$, we have $\Pr[u = j | \gamma = 1] = \Pr[u = j | \gamma =$

$0] = 1/Q$. Using these, we can calculate $\mathcal{B}_{\mathtt{in}}$'s $\mathtt{wNM\text{-}DCCA}$ advantage as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\Gamma_{\mathtt{in}},\mathcal{B}_{\mathtt{in}}}(k) &= 2 \cdot \left|\Pr[\gamma' = \gamma] - \frac{1}{2}\right| = \left|\Pr[\gamma' = 1|\gamma = 1] - \Pr[\gamma' = 1|\gamma = 0]\right| \\
&= \left|\Pr[\mathsf{Out}^{(u)}_{\mathcal{B}} \wedge \overline{\mathsf{In}_{\mathcal{B}}}|\gamma = 1] - \Pr[\mathsf{Out}^{(u)}_{\mathcal{B}} \wedge \overline{\mathsf{In}_{\mathcal{B}}}|\gamma = 0]\right| \\
&= \Big|\sum_{j\in[Q]}\Big(\Pr[\mathsf{Out}^{(u)}_{\mathcal{B}} \wedge \overline{\mathsf{In}_{\mathcal{B}}}|u = j \wedge \gamma = 1] \cdot \Pr[u = j|\gamma = 1] \\
&\qquad\qquad - \Pr[\mathsf{Out}^{(u)}_{\mathcal{B}} \wedge \overline{\mathsf{In}_{\mathcal{B}}}|u = j \wedge \gamma = 0] \cdot \Pr[u = j|\gamma = 0]\Big)\Big| \\
&= \frac{1}{Q}\Big|\sum_{j\in[Q]}\Big(\Pr[\mathsf{Out}^{(j)}_{\mathcal{B}} \wedge \overline{\mathsf{In}_{\mathcal{B}}}|\gamma = 1] - \Pr[\mathsf{Out}^{(j)}_{\mathcal{B}} \wedge \overline{\mathsf{In}_{\mathcal{B}}}|\gamma = 0]\Big)\Big|.
\end{aligned}$$

Consider the case when $\gamma = 1$. It is easy to see that in this case, $\mathcal{B}_{\mathtt{in}}$ simulates Game 2 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b *until the point $\mathcal{A}$ submits an inner-dangerous query.* In particular, the value α^*_1 associated with $c^*_{\mathtt{in}}$ is used as (r^*, K^*_1) as is done in Game 2. All other values (PK, c^*, and the answers to decapsulation queries) are distributed identically to those of Game 2. Furthermore, $\mathcal{B}_{\mathtt{in}}$ can detect whether $\mathcal{A}$'s query is inner-dangerous by using $sk_{\mathtt{out}}$, $\mathsf{F}_{\mathtt{out}}$, c^*, and $\mathsf{F}_{\mathtt{in}}$. These imply that for every $j \in [Q]$, we have $\Pr[\mathsf{Out}^{(j)}_{\mathcal{B}} \wedge \overline{\mathsf{In}_{\mathcal{B}}}|\gamma = 1] = \Pr[\mathsf{Out}^{(j)}_2 \wedge \overline{\mathsf{In}_2}]$.

On the other hand, when $\gamma = 0$, $\mathcal{B}_{\mathtt{in}}$ simulates Game 3 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b *until the point $\mathcal{A}$ submits an inner-dangerous query.* In particular, the uniformly chosen random value α^*_0 (independent of $c^*_{\mathtt{in}}$) is used as (r^*, K^*_1), which is exactly how they are distributed in Game 3. The rest is unchanged from the case of $\gamma = 1$, and thus, with a similar argument to the above, for every $j \in [Q]$, we have $\Pr[\mathsf{Out}^{(j)}_{\mathcal{B}} \wedge \overline{\mathsf{In}_{\mathcal{B}}}|\gamma = 0] = \Pr[\mathsf{Out}^{(j)}_3 \wedge \overline{\mathsf{In}_3}]$.

In summary, we have $\mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\Gamma_{\mathtt{in}},\mathcal{B}_{\mathtt{in}}}(k) = (1/Q) \cdot |\sum_{j\in[Q]}(\Pr[\mathsf{Out}^{(j)}_2 \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{Out}^{(j)}_3 \wedge \overline{\mathsf{In}_3}])|$, as required. □ (**Claim 9**)

Claim 10 *For any constants $p, q \in [0, 1]$, there exists a PPTA $\mathcal{B}'_{\mathtt{in}}$ such that* $\mathsf{Adv}^{\mathtt{DCCA}}_{\Gamma_{\mathtt{in}},\mathcal{B}'_{\mathtt{in}}}(k) = |p \cdot (\Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}]) + q \cdot (\Pr[\mathsf{In}_2] - \Pr[\mathsf{In}_3])|$.

Proof of Claim 10. Fix arbitrarily $p, q \in [0, 1]$. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}'_{\mathtt{in}}$ that attacks the $\mathtt{DCCA}$ security of the detectable KEM $\Gamma_{\mathtt{in}}$ with the claimed advantage. The description of $\mathcal{B}'_{\mathtt{in}}$ is as follows:

$\mathcal{B}'^{\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}},\cdot)}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, \alpha^*_\gamma)$**:** (where $\gamma \in \{0, 1\}$ is $\mathcal{B}'_{\mathtt{in}}$'s challenge bit in the $\mathtt{DCCA}$ experiment) $\mathcal{B}'_{\mathtt{in}}$ first picks two coins $b_p, b_q \in \{0, 1\}$ such that $b_p = 1$ (resp. $b_q = 1$) holds with probability p (resp. q). Next, $\mathcal{B}'_{\mathtt{in}}$ parses α^*_γ as $(r^*, K^*_1) \in \{0, 1\}^\ell \times \{0, 1\}^k$, picks $K^*_0 \in \{0, 1\}^k$ and $b \in \{0, 1\}$ uniformly at random, and then runs $(pk_{\mathtt{out}}, sk_{\mathtt{out}}) \leftarrow \mathsf{PKG}_{\mathtt{out}}(1^k)$ and $c^* \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*_{\mathtt{in}}; r^*)$. Then $\mathcal{B}'_{\mathtt{in}}$ sets $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathtt{out}})$, and runs $\mathcal{A}(PK, c^*, K^*_b)$.

$\mathcal{B}'_{\mathtt{in}}$ answers $\mathcal{A}$'s decapsulation queries c in the same way as $\mathcal{B}_{\mathtt{in1}}$ in the proof of Claim 9 does, except that if $\mathcal{B}'_{\mathtt{in}}$ detects that $\mathcal{A}$'s query is inner-dangerous, then $\mathcal{B}'_{\mathtt{in}}$ terminates with output $\gamma' \leftarrow b_q$.

When $\mathcal{A}$ terminates with output its guess bit b' (which means that $\mathcal{A}$ made no inner-dangerous query), $\mathcal{B}'_{\mathtt{in}}$ sets $\gamma' \leftarrow 1$ if both $b' = b$ and $b_p = 1$ hold, otherwise (i.e. $b' \neq b$ or $b_p = 0$) $\mathcal{B}'_{\mathtt{in}}$ sets $\gamma' \leftarrow 0$, and terminates with output γ'.

The above completes the description of $\mathcal{B}'_{\mathtt{in}}$. Note that $\mathcal{B}'_{\mathtt{in}}$ never submits a prohibited decapsulation query $c_{\mathtt{in}}$ satisfying $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c_{\mathtt{in}}) = 1$ to its decapsulation oracle. Let $\mathsf{In}_{\mathcal{B}}$ be the event that $\mathcal{A}$ submits an inner-dangerous decapsulation query in the experiment simulated by $\mathcal{B}'_{\mathtt{in}}$.

Consider the case when $\gamma = 1$. It is not hard to see that in this case, $\mathcal{B}'_{\mathtt{in}}$ simulates Game 2 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b *until the point* $\mathcal{A}$ *submits an inner-dangerous query.* In particular, the value α^*_1 associated with $c^*_{\mathtt{in}}$ is used as (r^*, K^*_1) as is done in Game 2. All other values (PK, c^*, and the answers to decapsulation queries) are distributed identically to those of Game 2. Furthermore, $\mathcal{B}'_{\mathtt{in}}$ can detect whether $\mathcal{A}$'s query is inner-dangerous by using $sk_{\mathtt{out}}$, $\mathsf{F}_{\mathtt{out}}$, c^*, $c^*_{\mathtt{in}}$, and $\mathsf{F}_{\mathtt{in}}$. These imply that $\Pr[b' = b \wedge \overline{\mathsf{In}_{\mathcal{B}}} | \gamma = 1] = \Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}]$ and $\Pr[\mathsf{In}_{\mathcal{B}} | \gamma = 1] = \Pr[\mathsf{In}_2]$. Recall that $\mathcal{B}'_{\mathtt{in}}$ outputs $\gamma' = 1$ only if either (1) $\mathcal{A}$ succeeds in guessing b without making any inner-dangerous queries (i.e. $b' = b \wedge \overline{\mathsf{In}_{\mathcal{B}}}$ occurs) and $b_p = 1$, or (2) $\mathcal{A}$ makes an inner-dangerous query ($\mathsf{In}_{\mathcal{B}}$ occurs) and $b_q = 1$. Furthermore, the choice of the coins b_p and b_q is independent of the behavior of $\mathcal{A}$ and $\mathcal{B}'_{\mathtt{in}}$'s challenge bit. These imply

$$\begin{aligned}\Pr[\gamma' = 1 | \gamma = 1] &= \Pr[b_p = 1 \wedge b' = b \wedge \overline{\mathsf{In}_{\mathcal{B}}} | \gamma = 1] + \Pr[b_q = 1 \wedge \mathsf{In}_{\mathcal{B}} | \gamma = 1] \\ &= p \cdot \Pr[b' = b \wedge \overline{\mathsf{In}_{\mathcal{B}}} | \gamma = 1] + q \cdot \Pr[\mathsf{In}_{\mathcal{B}} | \gamma = 1] \\ &= p \cdot \Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}] + q \cdot \Pr[\mathsf{In}_2].\end{aligned}$$

On the other hand, when $\gamma = 0$, $\mathcal{B}'_{\mathtt{in}}$ simulates Game 3 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b *until the point* $\mathcal{A}$ *submits an inner-dangerous query.* In particular, the uniformly chosen random value α^*_0 (independent of $c^*_{\mathtt{in}}$) is used as (r^*, K^*_1), which is exactly how they are distributed in Game 3. The rest is unchanged from the case of $\gamma = 1$, and thus, with a similar argument to the above, we have $\Pr[\gamma' = 1 | \gamma = 0] = p \cdot \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] + q \cdot \Pr[\mathsf{In}_3]$.

We can calculate $\mathcal{B}'_{\mathtt{in}}$'s $\mathtt{DCCA}$ advantage as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{DCCA}}_{\Gamma_{\mathtt{in}}, \mathcal{B}'_{\mathtt{in}}}(k) &= 2 \cdot \left|\Pr[\gamma' = \gamma] - \frac{1}{2}\right| = \left|\Pr[\gamma' = 1 | \gamma = 1] - \Pr[\gamma' = 1 | \gamma = 0]\right| \\ &= \left|p \cdot (\Pr[\mathsf{S}_2 \wedge \overline{\mathsf{In}_2}] - \Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}]) + q \cdot (\Pr[\mathsf{In}_2] - \Pr[\mathsf{In}_3])\right|.\end{aligned}$$

□ (**Claim 10**)

Claim 11 $|\Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] + \frac{1}{2}\Pr[\mathsf{In}_3] - \frac{1}{2}| = 0$.

Proof of Claim 11. Note that in Game 3, the real session-key K^*_1 is chosen uniformly at random, independently of $\mathcal{A}$'s challenge ciphertext. Furthermore, the behavior of the decapsulation oracle in Game 3 is independent of the challenge bit. Since the distribution of K^*_1 and that of K^*_0 are identical, $\mathcal{A}$'s view in Game 3 is distributed identically regardless of the challenge bit b. This guarantees that $\Pr[\mathsf{S}_3] = 1/2$, and that the event $\overline{\mathsf{In}_3}$ is independent of S_3. Hence, we have $\Pr[\mathsf{S}_3 \wedge \overline{\mathsf{In}_3}] = \frac{1}{2}\Pr[\overline{\mathsf{In}_3}]$, which implies the claim. □ (**Claim 11**)

Claim 12 *There exists a PPTA* $\mathcal{B}_{\mathtt{out}}$ *such that* $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi_{\mathtt{out}}, \mathcal{B}_{\mathtt{out}}}(k) = (1/Q) \cdot \sum_{j \in [Q]} \Pr[\mathsf{Out}^{(j)}_3]$.

Proof of Claim 12. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}_{\mathtt{out}}$ that attacks the randomness-inextractability of the detectable PKE scheme $\Pi_{\mathtt{out}}$ with the claimed advantage. The description of $\mathcal{B}_{\mathtt{out}} = (\mathcal{B}_{\mathtt{out}1}, \mathcal{B}_{\mathtt{out}2})$ is as follows:

$\mathcal{B}_{\mathtt{out1}}(1^k)$**:** $\mathcal{B}_{\mathtt{out1}}$ first runs $(pk_{\mathtt{in}}, sk_{\mathtt{in}}) \leftarrow \mathsf{KKG}_{\mathtt{in}}(1^k)$ and $(c^*_{\mathtt{in}}, \alpha^*) \leftarrow \mathsf{Encap}_{\mathtt{in}}(pk_{\mathtt{in}})$. Then $\mathcal{B}_{\mathtt{out1}}$ sets $M \leftarrow c^*_{\mathtt{in}}$ and $\mathsf{st}_{\mathcal{B}} \leftarrow (\mathcal{B}_{\mathtt{out1}}\text{'s entire view})$, and terminates with output $(M, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathtt{out2}}^{\mathsf{Dec}_{\mathsf{out}}(sk_{\mathsf{out}},\cdot)}(\mathsf{st}_{\mathcal{B}}, pk_{\mathtt{out}}, c^*)$**:** $\mathcal{B}_{\mathtt{out2}}$ sets $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathtt{out}})$, picks $K^* \in \{0,1\}^k$ uniformly at random, and runs $\mathcal{A}(PK, c^*, K^*)$.

$\mathcal{B}_{\mathtt{out2}}$ answers $\mathcal{A}$'s decapsulation queries as Game 3 does, which is possible because $\mathcal{B}_{\mathtt{out2}}$ possesses $sk_{\mathtt{in}}$ and has access to the decryption oracle $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, \cdot)$.

When $\mathcal{A}$ terminates, $\mathcal{B}_{\mathtt{out2}}$ picks $u \in [Q]$ uniformly at random, and proceeds as follows: Let $c^{(u)}$ be the u-th decapsulation query submitted by $\mathcal{A}$, and let $c^{(u)}_{\mathtt{in}} = \mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(u)})$ be the inner ciphertext corresponding to $c^{(u)}$ (note that this must have been already computed when having answered to the u-th decapsulation query). If $c^{(u)}$ is outer-dangerous, namely, it holds that $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c^{(u)}) = 1$, $c^{(u)}_{\mathtt{in}} \notin \{c^*_{\mathtt{in}}, \bot\}$, $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, c^{(u)}_{\mathtt{in}}) = (r^{(u)} \| K^{(u)}) \neq \bot$, and $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c^{(u)}_{\mathtt{in}}; r^{(u)}) = c^{(u)}$, then $\mathcal{B}_{\mathtt{out2}}$ sets $M' \leftarrow c^{(u)}_{\mathtt{in}}$ and $R' \leftarrow r^{(u)}$, and terminates with output (M', R'). Otherwise, $\mathcal{B}_{\mathtt{out2}}$ gives up and aborts.

The above completes the description of $\mathcal{B}_{\mathtt{out}}$. For $j \in [Q]$, let $\mathsf{Out}^{(j)}_{\mathcal{B}}$ be the event that $\mathcal{A}$'s j-th decapsulation query $c^{(j)}$ is outer-dangerous in the experiment simulated by $\mathcal{B}_{\mathtt{out}}$. Note that due to our design of $\mathcal{B}_{\mathtt{out}}$, $\mathcal{B}_{\mathtt{out}}$ succeeds in making the randomness-inextractability experiment output 1 if and only if $\mathsf{Out}^{(u)}_{\mathcal{B}}$ occurs, i.e. we have $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi_{\mathtt{out}}, \mathcal{B}_{\mathtt{out}}}(k) = \Pr[\mathsf{Out}^{(u)}_{\mathcal{B}}]$.

It is not hard to see that $\mathcal{B}_{\mathtt{out}}$ simulates Game 3 perfectly for $\mathcal{A}$. In particular, c^* is an encryption of $c^*_{\mathtt{in}}$ generated by using a uniformly chosen randomness, which is exactly how $\mathcal{A}$'s challenge ciphertext in Game 3 is generated. Therefore, for every $j \in [Q]$, the probability that $\mathsf{Out}^{(j)}_{\mathcal{B}}$ occurs in the experiment simulated by $\mathcal{B}_{\mathtt{out}}$ is identical to the probability that $\mathsf{Out}^{(j)}_3$ occurs in Game 3, namely we have $\Pr[\mathsf{Out}^{(j)}_{\mathcal{B}}] = \Pr[\mathsf{Out}^{(j)}_3]$ for every $j \in [Q]$. Furthermore, $u \in [Q]$ is chosen uniformly at random, independently of $\mathcal{A}$'s behavior. Therefore, for every $j \in [Q]$, we have $\Pr[u = j] = 1/Q$.

Put everything together, we can calculate $\mathcal{B}_{\mathtt{out}}$'s $\mathtt{R\text{-}Inext}$ advantage as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi_{\mathtt{out}}, \mathcal{B}_{\mathtt{out}}}(k) = \Pr[\mathsf{Out}^{(u)}_{\mathcal{B}}] = \sum_{j \in [Q]} \Pr[\mathsf{Out}^{(u)}_{\mathcal{B}} | u = j] \cdot \Pr[u = j] = \frac{1}{Q} \sum_{j \in [Q]} \Pr[\mathsf{Out}^{(j)}_{\mathcal{B}}] \\ = \frac{1}{Q} \sum_{j \in [Q]} \Pr[\mathsf{Out}^{(j)}_3].\end{aligned}$$

□ (**Claim 12**)

Claim 13 *There exists a PPTA* $\mathcal{B}'_{\mathtt{out}}$ *such that* $\mathsf{Adv}^{\mathtt{DCCA}}_{\Pi, \mathcal{B}'_{\mathtt{out}}}(k) = |\Pr[\mathsf{In}_3] - \Pr[\mathsf{In}_4]|$.

Proof of Claim 13. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}'_{\mathtt{out}}$ that attacks the $\mathtt{DCCA}$ security of the detectable PKE scheme $\Pi_{\mathtt{out}}$ with the claimed advantage. The description of $\mathcal{B}'_{\mathtt{out}} = (\mathcal{B}'_{\mathtt{out1}}, \mathcal{B}'_{\mathtt{out2}})$ is as follows:

$\mathcal{B}'^{\mathsf{Dec}_{\mathsf{out}}(sk_{\mathsf{out}},\cdot)}_{\mathtt{out1}}(pk_{\mathtt{out}})$**:** $\mathcal{B}'_{\mathtt{out1}}$ first executes $(pk_{\mathtt{in}}, sk_{\mathtt{in}}) \leftarrow \mathsf{KKG}_{\mathtt{in}}(1^k)$ and $(c^*_{\mathtt{in}}, \alpha^*) \leftarrow \mathsf{Encap}_{\mathtt{in}}(pk_{\mathtt{in}})$. Then $\mathcal{B}'_{\mathtt{out1}}$ sets $M_0 \leftarrow 0^n$, $M_1 \leftarrow c^*_{\mathtt{in}}$, and $\mathsf{st}_{\mathcal{B}} \leftarrow (\mathcal{B}'_{\mathtt{out1}}\text{'s entire view})$, and terminates with output $(M_0, M_1, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}'^{\mathsf{Dec}_{\mathsf{out}}(sk_{\mathsf{out}},\cdot)}_{\mathtt{out2}}(\mathsf{st}_{\mathcal{B}}, c^*)$**:** $\mathcal{B}'_{\mathtt{out2}}$ first picks $K^* \in \{0,1\}^k$ uniformly at random, sets $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathtt{out}})$, and runs $\mathcal{A}(PK, c^*, K^*)$.

$\mathcal{B}'_{\mathtt{out2}}$ answers $\mathcal{A}$'s decapsulation queries c as the decapsulation oracle in Game 3 does, which is possible because $\mathcal{B}'_{\mathtt{out2}}$ has access to the decryption oracle, and possesses $sk_{\mathtt{in}}$. $\mathcal{B}'_{\mathtt{out2}}$ also remembers whether $\mathcal{A}$'s query is inner-dangerous, which $\mathcal{B}_{\mathtt{out2}}$ can detect by using $\mathsf{F}_{\mathtt{in}}$ and $c^*_{\mathtt{in}}$. Note that if $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$, then $\mathcal{B}'_{\mathtt{out2}}$ can immediately return $\perp$ to $\mathcal{A}$, which is the correct behavior of the decapsulation oracle in Game 3 (and Game 4), and thus in this case $\mathcal{B}'_{\mathtt{out2}}$ need not use its decryption oracle. (Note also that such a query is not inner-dangerous.)

When $\mathcal{A}$ terminates, $\mathcal{B}'_{\mathtt{out2}}$ checks whether $\mathcal{A}$ has submitted an inner-dangerous decapsulation query. If an inner-dangerous query has been made, $\mathcal{B}'_{\mathtt{out2}}$ sets $\gamma' \leftarrow 1$, otherwise sets $\gamma' \leftarrow 0$, and terminates with output γ'.

The above completes the description of $\mathcal{B}'_{\mathtt{out}}$. Note that $\mathcal{B}'_{\mathtt{out2}}$ never submits a prohibited query c satisfying $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c) = 1$ to its decryption oracle.

Let $\gamma \in \{0, 1\}$ be $\mathcal{B}'_{\mathtt{out}}$'s challenge bit, and let $\mathsf{In}_{\mathcal{B}}$ be the event that $\mathcal{A}$ submits an inner-dangerous query in the experiment simulated by $\mathcal{B}'_{\mathtt{out}}$. Note that $\mathcal{B}'_{\mathtt{out2}}$ outputs 1 only when $\mathsf{In}_{\mathcal{B}}$ occurs. Therefore, $\mathcal{B}'_{\mathtt{out}}$'s advantage can be calculated as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{DCCA}}_{\Pi_{\mathrm{out}},\mathcal{B}'_{\mathrm{out}}}(k) &= 2 \cdot \left|\Pr[\gamma' = \gamma] - \frac{1}{2}\right| = \left|\Pr[\gamma' = 1|\gamma = 1] - \Pr[\gamma' = 1|\gamma = 0]\right| \\ &= \left|\Pr[\mathsf{In}_{\mathcal{B}}|\gamma = 1] - \Pr[\mathsf{In}_{\mathcal{B}}|\gamma = 0]\right|.\end{aligned}$$

Furthermore, it is not hard to see that if $\gamma = 1$ (resp. $\gamma = 0$), then $\mathcal{B}'_{\mathtt{out}}$ perfectly simulates Game 3 (resp. Game 4) for $\mathcal{A}$. Specifically, c^* is an encryption of $c^*_{\mathtt{in}}$ if $\gamma = 1$, and is an encryption of 0^n if $\gamma = 0$, which is exactly how $\mathcal{A}$'s challenge ciphertext in Game 3 and that in Game 4 are computed, respectively. Furthermore, $\mathcal{B}'_{\mathtt{out}}$ answers $\mathcal{A}$'s decapsulation queries c perfectly as the decapsulation oracle in Game 3 (and Game 4) does, using $\mathcal{B}'_{\mathtt{out2}}$'s decryption oracle, $\mathsf{F}_{\mathtt{out}}$ (which is publicly computable), and $sk_{\mathtt{in}}$, c^*, and $c^*_{\mathtt{in}}$ (that are all available for $\mathcal{B}'_{\mathtt{out}}$). These imply that we have $\Pr[\mathsf{In}_{\mathcal{B}}|\gamma = 1] = \Pr[\mathsf{In}_3]$ and $\Pr[\mathsf{In}_{\mathcal{B}}|\gamma = 0] = \Pr[\mathsf{In}_4]$.

In summary, we have $\mathsf{Adv}^{\mathtt{DCCA}}_{\Pi_{\mathrm{out}},\mathcal{B}'_{\mathrm{out}}}(k) = |\Pr[\mathsf{In}_3] - \Pr[\mathsf{In}_4]|$, as required. □ (**Claim 13**)

Claim 14 *There exists a PPTA $\mathcal{B}''_{\mathtt{in}}$ such that $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathrm{in}},\mathcal{B}''_{\mathrm{in}}}(k) \geq (1/Q) \cdot \sum_{j \in [Q]} \Pr[\mathsf{In}^{(j)}_4]$.*

Proof of Claim 14. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}''_{\mathtt{in}}$ that attacks the unpredictability of the detectable KEM $\Gamma_{\mathtt{in}}$ with the claimed advantage. The description of $\mathcal{B}''_{\mathtt{in}}$ is as follows:

$\mathcal{B}''^{\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}},\cdot)}_{\mathtt{in}}(pk_{\mathtt{in}})$**:** $\mathcal{B}''_{\mathtt{in}}$ first executes $(pk_{\mathtt{out}}, sk_{\mathtt{out}}) \leftarrow \mathsf{PKG}_{\mathtt{out}}(1^k)$ and $c^* \leftarrow \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, 0^n)$. Next, $\mathcal{B}''_{\mathtt{in}}$ picks $K^* \in \{0, 1\}^k$ uniformly at random, sets $PK \leftarrow (pk_{\mathtt{in}}, pk_{\mathtt{out}})$, and then runs $\mathcal{A}(PK, c^*, K^*)$.

$\mathcal{B}''_{\mathtt{in}}$ answers decapsulation queries c from $\mathcal{A}$ as the decapsulation oracle in Game 4 does, where $\mathcal{B}''_{\mathtt{in}}$ uses its own decapsulation oracle as a substitute for $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, \cdot)$.

When $\mathcal{A}$ terminates, $\mathcal{B}''_{\mathtt{in}}$ picks $u \in [Q]$ uniformly, and proceeds as follows: Let $c^{(u)}$ be the u-th query submitted by $\mathcal{A}$. If $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(j)}) = c^{(u)}_{\mathtt{in}} \neq \perp$ holds, then $\mathcal{B}''_{\mathtt{in}}$ terminates with output $c^{(u)}_{\mathtt{in}}$. Otherwise, $\mathcal{B}''_{\mathtt{in}}$ gives up and aborts.

The above completes the description of $\mathcal{B}''_{\mathtt{in}}$. It is easy to see that $\mathcal{B}''_{\mathtt{in}}$ perfectly simulates Game 4 for $\mathcal{A}$.

Let $c^*_{\mathtt{in}}$ be the challenge ciphertext generated by $\mathcal{B}''_{\mathtt{in}}$'s unpredictability experiment. Since the challenge ciphertext $c^*_{\mathtt{in}}$ is information-theoretically hidden from $\mathcal{B}''_{\mathtt{in}}$ in the unpredictability experiment, we can assume without loss of generality that $c^*_{\mathtt{in}}$ is generated at the beginning of the experiment. Now, consider the following events in the experiment simulated by $\mathcal{B}''_{\mathtt{in}}$:

$\mathsf{In}^{(j)}_{\mathcal{B}}$ **where** $j \in [Q]$**:** $\mathcal{A}$'s j-th query $c^{(j)}$ is *inner-dangerous with respect to* $c^*_{\mathtt{in}}$. Namely, it holds that $\mathsf{F}_{\mathtt{out}}(pk_{\mathtt{out}}, c^*, c^{(j)}) = 0$, $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(j)}) = c^{(j)}_{\mathtt{in}} \neq \bot$, and $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c^{(j)}_{\mathtt{in}}) = 1$.

$\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}}$ **where** $j \in [Q]$**:** $\mathcal{A}$'s j-th query $c^{(j)}$ satisfies $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, c^{(j)}) = c^{(j)}_{\mathtt{in}} \neq \bot$ and $\mathsf{F}_{\mathtt{in}}(pk_{\mathtt{in}}, c^*_{\mathtt{in}}, c^{(j)}_{\mathtt{in}})$ $= 1$.

Note the difference between $\mathsf{In}^{(j)}_{\mathcal{B}}$ and $\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}}$. The former event implies the latter, and hence for every $j \in [Q]$, we have $\Pr[\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}}] \geq \Pr[\mathsf{In}^{(j)}_{\mathcal{B}}]$. Furthermore, by definition of the unpredictability experiment and our design of $\mathcal{B}''_{\mathtt{in}}$, we have $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{in}},\mathcal{B}''_{\mathtt{in}}}(k) = \Pr[\widehat{\mathsf{In}}^{(u)}_{\mathcal{B}}]$. Moreover, note that the distribution of the "inner" ciphertext $c^*_{\mathtt{in}}$ in Game 4 and that of $\mathcal{B}''_{\mathtt{in}}$'s challenge ciphertext in the unpredictability experiment are identical. Since $\mathcal{B}''_{\mathtt{in}}$ perfectly simulates Game 4 for $\mathcal{A}$, for every $j \in [Q]$, we have $\Pr[\mathsf{In}^{(j)}_{\mathcal{B}}] = \Pr[\mathsf{In}^{(j)}_4]$. Finally, note that the index $u \in [Q]$ is chosen uniformly, independently of $\mathcal{A}$'s behavior. Hence, for every $j \in [Q]$, we have $\Pr[u = j] = 1/Q$.

Armed with these, $\mathcal{B}''_{\mathtt{in}}$'s unpredictability advantage can be calculated as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{in}},\mathcal{B}''_{\mathtt{in}}}(k) = \Pr[\widehat{\mathsf{In}}^{(u)}_{\mathcal{B}}] = \sum_{j\in[Q]} \Pr[\widehat{\mathsf{In}}^{(u)}_{\mathcal{B}} | u = j] \cdot \Pr[u = j] &= \frac{1}{Q} \sum_{j\in[Q]} \Pr[\widehat{\mathsf{In}}^{(j)}_{\mathcal{B}}] \\ &\geq \frac{1}{Q} \sum_{j\in[Q]} \Pr[\mathsf{In}^{(j)}_{\mathcal{B}}] = \frac{1}{Q} \sum_{j\in[Q]} \Pr[\mathsf{In}^{(j)}_4].\end{aligned}$$

□ (**Claim 14**)

Claims 8 to 14 guarantee that there exist PPTAs $\mathcal{B}_{\mathtt{in}}$, $\mathcal{B}'_{\mathtt{in1}}$, $\mathcal{B}'_{\mathtt{in2}}$, $\mathcal{B}_{\mathtt{out}}$, $\mathcal{B}'_{\mathtt{out}}$, and $\mathcal{B}''_{\mathtt{in}}$, satisfying Equation (8), where the second and the third terms in the right hand side of Equation (8) are derived from Claim 10 in which we set $(p, q) = (1, 1/2)$ and $(p, q) = (0, 1)$, respectively. Recall that the choice of $\mathcal{A}$ was arbitrarily, and thus for any PPTA $\mathcal{A}$, we can show a negligible upperbound on $\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma_{\mathtt{DL}},\mathcal{A}}(k)$. Therefore, $\Gamma_{\mathtt{DL}}$ is $\mathtt{CCA}$ secure. □ (**Theorem 2**)

D.4 Proof of Theorem 3: Why DCCA Security, Unpredictability, and Randomness-Inextractability Are Not Enough

Recall that due to the result by Hohenberger et al. [16], if there exists a detectable PKE scheme satisfying DCCA security and unpredictability, then there exists a CCA secure KEM (with arbitrarily long session-keys) and a CCA secure PKE scheme (which can encrypt arbitrarily long plaintexts).

We note that any CCA secure KEM can be seen as a detectable KEM satisfying DCCA security and unpredictability with respect to the "equality" predicate $\mathsf{F}(pk, c^*, c') := (c^* \overset{?}{=} c')$ [16]. Therefore, from a CCA secure KEM, we can construct a detectable KEM $\Gamma'_{\mathtt{in}} = (\mathsf{KKG}'_{\mathtt{in}}, \mathsf{Encap}'_{\mathtt{in}}, \mathsf{Decap}'_{\mathtt{in}}, \mathsf{F}'_{\mathtt{in}})$ such that:

- $\Gamma'_{\mathtt{in}}$ is DCCA secure and unpredictable.

- The session-key space is $\{0,1\}^{(3/2)k}$.[6]
- The ciphertext size is $n = n(k) > 0$ for some polynomial n.

Using this detectable KEM $\mathit{\Gamma}'_{\mathtt{in}}$ as a building block, consider another detectable KEM $\mathit{\Gamma}_{\mathtt{in}} = (\mathsf{KKG}_{\mathtt{in}}, \mathsf{Encap}_{\mathtt{in}}, \mathsf{Decap}_{\mathtt{in}}, \mathsf{F}_{\mathtt{in}})$ as described in Fig. 10 (left). The session-key space of $\mathit{\Gamma}_{\mathtt{in}}$ is $\{0,1\}^{3k}$, and the ciphertext size of it is $2n$. It is straightforward to see that if $\mathit{\Gamma}'_{\mathtt{in}}$ is $\mathtt{DCCA}$ secure and unpredictable, then so is $\mathit{\Gamma}_{\mathtt{in}}$. Note that $\mathit{\Gamma}_{\mathtt{in}}$ is designed to have an obvious malleability against the "swapping" attack: Suppose $C = (c_1, c_2)$ is generated by $\mathsf{Encap}_{\mathtt{in}}$ together with a session-key $K = (r_1 \| r_2 \| K'_1 \| K'_2) \in \{0,1\}^{3k}$, then the decapsulation result of the "swapped" ciphertext $\widehat{C} = (c_2, c_1)$ is $\widehat{K} = (r_2 \| r_1 \| K'_2 \| K'_1)$. [7]

Similarly, any $\mathtt{CCA}$ secure PKE scheme can be seen as a detectable PKE scheme satisfying $\mathtt{DCCA}$ security and randomness-inextractability with respect to the "equality" predicate [8]. Therefore, from a $\mathtt{CCA}$ secure PKE scheme, we can construct a detectable PKE scheme $\mathit{\Pi}'_{\mathtt{out}} = (\mathsf{PKG}'_{\mathtt{out}}, \mathsf{Enc}'_{\mathtt{out}}, \mathsf{Dec}'_{\mathtt{out}}, \mathsf{F}'_{\mathtt{out}})$ such that:

- $\mathit{\Pi}'_{\mathtt{out}}$ is $\mathtt{DCCA}$ secure and randomness-inextractable.
- The plaintext space is $\{0,1\}^n$.
- The randomness space of $\mathsf{Enc}'_{\mathtt{out}}$ is $\{0,1\}^k$.[9]

Using this detectable PKE scheme $\mathit{\Pi}'_{\mathtt{out}}$ as a building block, consider another detectable PKE scheme $\mathit{\Pi}_{\mathtt{out}} = (\mathsf{PKG}_{\mathtt{out}}, \mathsf{Enc}_{\mathtt{out}}, \mathsf{Dec}_{\mathtt{out}}, \mathsf{F}_{\mathtt{out}})$ as described in Fig. 10 (right). The plaintext space of $\mathit{\Pi}_{\mathtt{out}}$ is $\{0,1\}^{2n}$, and the randomness space of $\mathsf{Enc}_{\mathtt{out}}$ is $\{0,1\}^{2k}$. It is straightforward to see that if $\mathit{\Pi}'_{\mathtt{out}}$ is $\mathtt{DCCA}$ secure and randomness-inextractable, then so is $\mathit{\Pi}_{\mathtt{out}}$. Furthermore, as is similar to $\mathit{\Gamma}_{\mathtt{in}}$, $\mathit{\Pi}_{\mathtt{out}}$ is also malleable against the "swapping" attack. In fact, the "swapping" attack preserves the consistency of randomness: For a plaintext $m = (m_1, m_2) \in (\{0,1\}^n)^2$ and a randomness $r = (r_1, r_2) \in (\{0,1\}^k)^2$, let $\widehat{m} = (m_2, m_1)$ and $\widehat{r} = (r_2, r_1)$. Then, for every public key pk output by $\mathsf{PKG}_{\mathtt{out}}$, if $C = (c_1, c_2) = \mathsf{Enc}_{\mathtt{out}}(pk, m; r)$, then it holds that $\widehat{C} = (c_2, c_1) = \mathsf{Enc}_{\mathtt{out}}(pk, \widehat{m}; \widehat{r})$. [10]

Now, consider the double-layered KEM $\mathit{\Gamma}_{\mathtt{DL}} = (\mathsf{KKG}_{\mathtt{DL}}, \mathsf{Encap}_{\mathtt{DL}}, \mathsf{Decap}_{\mathtt{DL}})$ that is constructed by using $\mathit{\Gamma}_{\mathtt{in}}$ and $\mathit{\Pi}_{\mathtt{out}}$ as the inner KEM and the outer PKE scheme, respectively. It should be easy to see that this KEM $\mathit{\Gamma}_{\mathtt{DL}}$ inherits malleability against the "swapping" attack from the building blocks $\mathit{\Gamma}_{\mathtt{in}}$ and $\mathit{\Pi}_{\mathtt{out}}$.

Specifically, let $(PK, SK) = ((pk_{\mathtt{in}}, pk_{\mathtt{out}}), (sk_{\mathtt{in}}, sk_{\mathtt{out}}))$ be a key pair of $\mathit{\Gamma}_{\mathtt{DL}}$, and suppose $C = (c_1, c_2)$ is a ciphertext and $K = (K'_1 \| K'_2) \in \{0,1\}^k$ is the session-key corresponding to C. Then, we can show that the decapsulation result of the "swapped" ciphertext $\widehat{C} = (c_2, c_1)$ is the "swapped" session-key $\widehat{K} = (K'_2 \| K'_1)$, i.e., $\mathsf{Decap}_{\mathtt{DL}}(SK, \widehat{C}) = (K'_2 \| K'_1) \in \{0,1\}^k$. To see this, note that if $C = (c_1, c_2)$ is generated by $\mathsf{Encap}_{\mathtt{DL}}(PK)$, then each c_i is of the form $c_i = \mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, c_{\mathtt{in}i}; r_i)$ where each $c_{\mathtt{in}i}$ is generated by $\mathsf{Encap}_{\mathtt{in}}(pk_{\mathtt{in}})$ together with a session-key $\alpha'_i = (r_i \| K'_i) \in \{0,1\}^{(3/2)k}$. Thus, the decapsulation $\mathsf{Decap}_{\mathtt{DL}}(SK, \widehat{C} = (c_2, c_1))$ proceeds as follows:

[6] The session-key space of a detectable KEM satisfying $\mathtt{DCCA}$ security and unpredictability can be freely adjusted by using a PRG with appropriate output length, which exists if a $\mathtt{CCA}$ secure KEM exists.

[7] It would be worth noting that the detectable KEM $\mathit{\Gamma}_{\mathtt{in}}$ considered here is not $\mathtt{wNM\text{-}DCCA}$ secure.

[8] Lemma 7 shown in Section 5.1 implies that any 1-bit $\mathtt{CCA}$ secure PKE scheme is by itself a detectable PKE scheme with randomness-inextractability with respect to the equality predicate. It should be easily inferred (and easily proved) that this is true for a $\mathtt{CCA}$ secure PKE with larger plaintext space.

[9] The randomness space of a detectable PKE scheme satisfying $\mathtt{DCCA}$ security and randomness-inextractability can be freely adjusted by using a PRG with appropriate output length, which exists if a $\mathtt{CCA}$ secure PKE scheme exists.

[10] Similarly to $\mathit{\Gamma}_{\mathtt{in}}$, the detectable PKE scheme $\mathit{\Pi}_{\mathtt{out}}$ considered here is not $\mathtt{wNM\text{-}DCCA}$ secure. (It is not $\mathtt{wRNM\text{-}DCCA}$ secure, either.)

```
$\mathsf{KKG}_{\mathtt{in}}(1^k)$ :
    Return $(pk, sk) \leftarrow \mathsf{KKG}'_{\mathtt{in}}(1^k)$.
$\mathsf{Encap}_{\mathtt{in}}(pk)$ :
    $\forall i \in [2] : (c_i, \alpha'_i) \leftarrow \mathsf{Encap}'_{\mathtt{in}}(pk)$
    $\forall i \in [2]$ : Parse $\alpha'_i$ as $(r_i, K'_i) \in \{0,1\}^k \times \{0,1\}^{(1/2)k}$.
    $C \leftarrow (c_1, c_2)$
    $K \leftarrow (r_1 \| r_2 \| K'_1 \| K'_2)$
    Return $(C, K)$.
$\mathsf{Decap}_{\mathtt{in}}(sk, C)$ :
    Parse $C$ as $(c_1, c_2)$.
    $\forall i \in [2] : \alpha'_i \leftarrow \mathsf{Decap}'_{\mathtt{in}}(sk, c_i)$
    If $\alpha'_1 = \bot$ or $\alpha'_2 = \bot$ then return $\bot$.
    $\forall i \in [2]$ : Parse $\alpha'_i$ as $(r_i, K'_i) \in \{0,1\}^k \times \{0,1\}^{(1/2)k}$.
    Return $K \leftarrow (r_1 \| r_2 \| K'_1 \| K'_2)$.
$\mathsf{F}_{\mathtt{in}}(pk, C^*, C')$ :
    $(c_1^*, c_2^*) \leftarrow C^*$;   $(c'_1, c'_2) \leftarrow C'$.
    If $\exists i, j \in [2] : \mathsf{F}'_{\mathtt{in}}(pk, c_i^*, c'_j) = 1$
                                        then return 1 else return 0.
```

```
$\mathsf{PKG}_{\mathtt{out}}(1^k)$ :
    Return $(pk, sk) \leftarrow \mathsf{PKG}'_{\mathtt{out}}(1^k)$.
$\mathsf{Enc}_{\mathtt{out}}(pk, m; r)$ :
    Parse $m$ as $(m_1, m_2) \in (\{0,1\}^n)^2$.
    Parse $r$ as $(r_1, r_2) \in (\{0,1\}^k)^2$.
    $\forall i \in [2] : c_i \leftarrow \mathsf{Enc}'_{\mathtt{out}}(pk, m_i; r_i)$
    Return $C \leftarrow (c_1, c_2)$.
$\mathsf{Dec}_{\mathtt{out}}(sk, C)$ :
    Parse $C$ as $(c_1, c_2)$.
    $\forall i \in [2] : m_i \leftarrow \mathsf{Dec}'_{\mathtt{out}}(sk, c_i)$
    If $m_1 = \bot$ or $m_2 = \bot$ then return $\bot$.
    Return $m \leftarrow (m_1 \| m_2)$.
$\mathsf{F}_{\mathtt{out}}(pk, C^*, C')$ :
    $(c_1^*, c_2^*) \leftarrow C^*$;   $(c'_1, c'_2) \leftarrow C'$.
    If $\exists i, j \in [2] : \mathsf{F}'_{\mathtt{out}}(pk, c_i^*, c'_j) = 1$
                      then return 1 else return 0.
```

Fig. 10. The building blocks for showing the counterexample. The inner detectable KEM $\Gamma_{\mathtt{in}}$ (left) and the outer detectable PKE scheme $\Pi_{\mathtt{out}}$ (right).

1. It first runs $\mathsf{Dec}_{\mathtt{out}}(sk_{\mathtt{out}}, \widehat{C})$, which results in $\widehat{C}_{\mathtt{in}} = (c_{\mathtt{in}2}, c_{\mathtt{in}1})$.
2. Next, it runs $\mathsf{Decap}_{\mathtt{in}}(sk_{\mathtt{in}}, \widehat{C}_{\mathtt{in}})$, which results in $\widehat{\alpha} = (r_2 \| r_1 \| K'_2 \| K'_1) \in \{0,1\}^{3k}$.
3. Then, $\widehat{\alpha}$ is parsed into $\widehat{r} = (r_2 \| r_1) \in \{0,1\}^{2k}$ and $\widehat{K} = (K'_2 \| K'_1) \in \{0,1\}^k$.
4. Finally, $\mathsf{Decap}_{\mathtt{DL}}$ checks if $\mathsf{Enc}_{\mathtt{out}}(pk_{\mathtt{out}}, \widehat{C}_{\mathtt{in}}; \widehat{r}) = \widehat{C}$, which is always true as seen above, and thus returns $\widehat{K} = (K'_2 \| K'_1)$.

Therefore, a $\mathtt{CCA}$ adversary, given a public key $PK = (pk_{\mathtt{in}}, pk_{\mathtt{out}})$ and the challenge ciphertext $C^* = (c_1^*, c_2^*)$, can submit the "swapped" ciphertext $\widehat{C} = (c_2^*, c_1^*)$ to the decapsulation oracle, and obtain the "swapped" session-key $\widehat{K} = (K_2'^* \| K_1'^*)$. From this, the adversary can reconstruct the session-key $K^* = (K_1'^* \| K_2'^*)$, and hence always break $\mathtt{CCA}$ security (actually, even in the sense of one-wayness under 1-bounded $\mathtt{CCA}$). □ (**Theorem 3**)

D.5 Proof of Lemma 7: Randomness-Inextractability of $\Pi^n_{\mathtt{BE}}$

Fix a polynomial $n = n(k) > 0$. We will show that for any PPTA adversary $\mathcal{A}$ that attacks the randomness-inextractability of the detectable PKE scheme $\Pi^n_{\mathtt{BE}}$, there exists a PPTA adversary $\mathcal{B}$ that attacks the $\mathtt{CCA}$ security of the underlying 1-bit PKE scheme Π_1 in such a way that:

$$\mathsf{Adv}^{\mathtt{CCA}}_{\Pi_1, \mathcal{B}}(k) \geq \frac{1}{n} \cdot \mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi^n_{\mathtt{BE}}, \mathcal{A}}(k), \tag{10}$$

which by our assumption that Π_1 is $\mathtt{CCA}$ secure, will prove the lemma.

To this end, fix arbitrarily a PPTA $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that attacks the randomness-inextractability of the detectable PKE scheme $\Pi^n_{\mathtt{BE}}$. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}$ that attacks the $\mathtt{CCA}$ security of Π_1 with the advantage shown in Equation (10). The description of $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ is as follows:

$\mathcal{B}_1^{\mathsf{Dec}_1(sk,\cdot)}(pk)$: $\mathcal{B}_1$ first runs $(m = (m_1 \| \ldots \| m_n), \mathsf{st}) \leftarrow \mathcal{A}_1(1^k)$. Then $\mathcal{B}_1$ picks $u \in [n]$ uniformly at random, sets $M_1 \leftarrow m_u$, $M_0 \leftarrow 1 - m_u$, and $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_1$'s entire view), and terminates with output $(M_0, M_1, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_2^{\mathsf{Dec}_1(sk,\cdot)}(\mathsf{st}_{\mathcal{B}}, c^*)$: For all $i \in [n] \backslash \{u\}$, $\mathcal{B}_2$ runs $c_i^* \leftarrow \mathsf{Enc}_1(pk, m_i)$. Then, $\mathcal{B}_2$ sets $c_u^* \leftarrow c^*$ and $C^* \leftarrow (c_1^*, \dots, c_n^*)$, and then runs $\mathcal{A}_2(\mathsf{st}, pk, C^*)$.

Decryption queries from $\mathcal{A}_2$ are answered by using $\mathcal{B}_2$'s decryption oracle, except that if $\mathcal{B}_2$ needs to perform "$\mathsf{Dec}_1(sk, c^*)$", it does not use the decryption oracle but use m_u as the decryption result of c^*.

When $\mathcal{A}_2$ terminates with output $m' = (m'_1 \| \dots \| m'_n) \in \{0,1\}^n$ and $r' = (r'_1, \dots, r'_n) \in (\{0,1\}^\ell)^n$, $\mathcal{B}_2$ proceeds as follows: $\mathcal{B}_2$ computes $C' = (c'_1, \dots, c'_n) \leftarrow \mathsf{Enc}_{\mathtt{BE}}^n(pk, m'; r')$. (Note that $c'_i = \mathsf{Enc}_1(pk, m'_i; r'_i)$ holds for every $i \in [n]$.) If there exists $j \in [n]$ such that $c_u^* = c'_j$ and $m_u = m'_j$ simultaneously hold, then $\mathcal{B}_2$ sets $b' \leftarrow 1$, and otherwise, $\mathcal{B}_2$ sets $b' \leftarrow 0$. Finally, $\mathcal{B}_2$ terminates with output b'.

The above completes the description of $\mathcal{B}$. Note that $\mathcal{B}_2$ never submits the prohibited query c^* to its decryption oracle.

Let $b \in \{0,1\}$ be $\mathcal{B}$'s challenge bit, and let $\mathsf{Q}_{\mathcal{B}}$ be the event that $\mathsf{F}_{\mathtt{BE}}^n(pk, C^*, C') = 1$ occurs, i.e. there exist $i, j \in [n]$ such that $c_i^* = c'_j$, in the experiment simulated by $\mathcal{B}$. $\mathcal{B}$'s $\mathtt{CCA}$ advantage is calculated as follows:

$$\mathsf{Adv}_{\Pi_1,\mathcal{B}}^{\mathtt{CCA}}(k) = 2 \cdot \left| \Pr[b' = b] - \frac{1}{2} \right| = \left| \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0] \right|.$$

Consider the case $b = 1$. It is easy to see that in this case, $\mathcal{B}$ perfectly simulates the randomness-inextractability experiment for $\mathcal{A}$. In particular, since $c^* = c_u^*$ is an encryption of m_u, the challenge ciphertext C^* is generated in the same manner as that in $\mathcal{A}$'s randomness-inextractability experiment. Furthermore, $\mathcal{A}$'s decryption oracle is also perfectly simulated. In particular, even if $\mathcal{A}$'s decryption query C contains c^*, $\mathcal{B}_2$ uses m_u as the decryption result of c^*, and for $\mathcal{A}$'s queries that do not contain c^*, $\mathcal{B}_2$ can decrypt them using its own decryption oracle. Therefore, the probability that $\mathcal{A}$ outputs (m', r') such that $\mathsf{Q}_{\mathcal{B}}$ occurs is exactly the same as the probability that $\mathcal{A}$ outputs (m', r') that makes the randomness-inextractability experiment output 1, i.e. $\Pr[\mathsf{Q}_{\mathcal{B}} | \gamma = 1] = \mathsf{Adv}_{\Pi_{\mathtt{BE}}^n,\mathcal{A}}^{\mathtt{R\text{-}Inext}}(k)$. Furthermore, when $\mathsf{Q}_{\mathcal{B}}$ occurs, there exists a position $i \in [n]$ such that $c_i^* = c'_j$ holds for some $j \in [n]$. Under such positions i, j, it also holds that $m_i = m'_j$, due to the correctness of Π. Due to our design of $\mathcal{B}$, if this position i was u that $\mathcal{B}$ initially guesses, then $\mathcal{B}$ outputs $b' = 1$. Since the information of $u \in [n]$ is information-theoretically hidden from $\mathcal{A}$'s view throughout the experiment simulated by $\mathcal{B}$, $\mathcal{A}$'s behavior is independent of the choice of u. Therefore, conditioned on the event $\mathsf{Q}_{\mathcal{B}}$ and $b = 1$, the probability that $\mathcal{B}$'s guess on the position that causes the event $\mathsf{Q}_{\mathcal{B}}$ was right (and hence $\mathcal{B}$ outputs 1) is at least $1/n$. That is, we have $\Pr[b' = 1 | b = 1] \geq (1/n) \cdot \Pr[\mathsf{Q}_{\mathcal{B}} | \gamma = 1] = (1/n) \cdot \mathsf{Adv}_{\Pi_{\mathtt{BE}}^n,\mathcal{A}}^{\mathtt{R\text{-}Inext}}(k)$.

When $b = 0$, on the other hand, $\mathcal{B}_2$ never outputs 1, and hence $\Pr[b' = 1 | b = 0] = 0$. To see this, note that in this case c^* is an encryption of $1 - m_u$. Note also that every c'_j is of the form $c'_j = \mathsf{Enc}_1(pk, m'_j; r'_j)$. Therefore, even if there is a position $j \in [n]$ such that $c_u^* = c'_j$ holds, $m_u = m'_j$ cannot hold under the same j due to the correctness of Π_1.

In summary, we have $\mathsf{Adv}_{\Pi_1,\mathcal{B}}^{\mathtt{CCA}}(k) \geq (1/n) \cdot \mathsf{Adv}_{\Pi_{\mathtt{BE}}^n,\mathcal{A}}^{\mathtt{R\text{-}Inext}}(k)$, as required. □ (**Lemma 7**)

D.6 Proof of Lemma 8: wRNM-DCCA Security of $\Pi_{\mathtt{EtBE}}$

In this proof, for simplicity and clarity, we rely on the security results for the bitwise-encrypt construction $\Pi_{\mathtt{BE}}^n$, which in turn follows from the $\mathtt{CCA}$ security of the building block 1-bit scheme Π_1. Furthermore, for notational convenience, for a ciphertext $C = (c_1, \dots, c_n)$ of $\Pi_{\mathtt{BE}}^n$ and a ciphertext c of Π_1, we write "$c \in C$" to mean that there exists $i \in [n]$ such that $c = c_i$.

Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ be any PPTA adversary that attacks the $\mathtt{wRNM\text{-}DCCA}$ security of the detectable PKE scheme $\Pi_{\mathtt{EtBE}}$. We will show that there exist PPTAs $\mathcal{B}_{\mathtt{p}}$ and $\mathcal{B}_{\mathtt{n}}$ such that

$$\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{EtBE}},\mathcal{A}}(k) \leq 2 \cdot \mathsf{Adv}^{\mathtt{DCCA}}_{\Pi_{\mathtt{BE}},\mathcal{B}_{\mathtt{p}}}(k) + \mathsf{Adv}^{\mathcal{Q}\text{-}\mathtt{NM}}_{\mathcal{C},\mathcal{B}_{\mathtt{n}}}(k). \tag{11}$$

For simplicity, we assume that $\mathcal{A}_2$'s decryption queries C always satisfy $\mathtt{DUPCHK}(C) = 0$ and $\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C) = 0$, and the ciphertext C' that is finally output by $\mathcal{A}_2$ satisfies $\mathtt{DUPCHK}(C') = 0$ (but we allow C' to satisfy $\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C') = 1$).[11]

Now, consider the following sequence of games: (Here, the values with asterisk (*) represent those related to $\mathcal{A}$'s challenge ciphertext $C^* = (c^*_1, \dots, c^*_n)$.)

Game 1: This is the experiment $\mathsf{Expt}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{EtBE}},\mathcal{A}}(k)$ itself.

Game 2: Same as Game 1, except that if C^* satisfies $\mathtt{DUPCHK}(C^*) = 1$, then $\mathcal{A}_2$ and $\mathcal{A}_3$ are never executed. Instead, the game picks a bit $b' \in \{0, 1\}$ uniformly at random, and this bit b' is used for judging whether $\mathcal{A}$ has succeeded in the game (i.e. $b' = b$ occurs).

Game 3: Same as Game 2, except that if $\mathtt{DUPCHK}(C^*) = 0$, then when computing the decryption result m' (which is input to $\mathcal{A}_3$) of the ciphertext $C' = (c'_1, \dots c'_n)$ that is finally output by $\mathcal{A}_2$, every c'_i such that $c'_i \in C^*$ is never decrypted, and instead the corresponding bit of the codeword $s^* = (s^*_1 \| \dots \| s^*_n)$ is used as the decryption result of c'_i, where s^* is the codeword computed by $s^* \leftarrow \mathsf{E}(1^k, m_b)$ during the generation of C^*.

More formally, in Game 3, m' is computed by the following procedure: (The difference from Game 2 is only in the step 1 below.)

1. $\forall i \in [n] : s'_i \leftarrow \begin{cases} s^*_j & \text{if } c'_i = c^*_j \text{ for some } j \in [n] \\ \mathsf{Dec}_1(sk, c'_i) & \text{otherwise} \end{cases}$
2. If $\exists i \in [n] : s'_i = \perp$ then return $m' \leftarrow \perp$.
3. $s' \leftarrow (s'_1 \| \dots \| s'_n)$
4. $m' \leftarrow \mathsf{D}(1^k, s')$
5. If $m' \in \{m_0, m_1\}$ then $m' \leftarrow \mathsf{same}$.
6. Return m'.

Note that in the step 1, if $c'_i \in C^*$, then the position j such that $c'_i = c^*_j$ holds is uniquely determined, because we are assuming $\mathtt{DUPCHK}(C') = 0$.

Game 4: Same as Game 3, except that the information of s^* is erased from C^*. More precisely, the step "$C^* \leftarrow \mathsf{Enc}^n_{\mathtt{BE}}(pk, s^*)$" in Game 3 is replaced with the step "$C^* \leftarrow \mathsf{Enc}^n_{\mathtt{BE}}(pk, 0^n)$."

The above completes the description of the games.

For $i \in [4]$, let S_i be the event that $b' = b$ occurs in Game i. $\mathcal{A}$'s $\mathtt{wRNM\text{-}DCCA}$ advantage can be estimated as follows:

$$\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{EtBE}},\mathcal{A}}(k) = 2 \cdot \left|\Pr[\mathsf{S}_1] - \frac{1}{2}\right| \leq 2 \cdot \sum_{i \in [3]} \left|\Pr[\mathsf{S}_i] - \Pr[\mathsf{S}_{i+1}]\right| + \left|\Pr[\mathsf{S}_4] - \frac{1}{2}\right|. \tag{12}$$

It remains to upperbound each term that appears in the right hand side of the above inequality.

[11] This assumption is without loss of generality, because even if $\mathcal{A}$ does not respect this assumption, we can always consider a "wrapper" algorithm $\mathcal{A}'$ for $\mathcal{A}$ that runs in exactly the same way as $\mathcal{A}$, except that if $\mathcal{A}$'s decryption query does not satisfy the conditions, $\mathcal{A}'$ instead makes a dummy decryption query satisfying them (and returns $\perp$ to $\mathcal{A}$ as an "answer" to the query). $\mathcal{A}'$ can handle the final output C' of $\mathcal{A}_2$ analogously. Such "wrapper" algorithm $\mathcal{A}'$ has exactly the same advantage as $\mathcal{A}$.

Claim 15 $\Pr[\mathsf{S}_1] = \Pr[\mathsf{S}_2]$.

Proof of Claim 15. For $i \in [2]$, let C_i be the event that $\mathtt{DUPCHK}(C^*) = 1$ occurs in Game i. Note that Game 1 and Game 2 can differ only if C_1 or C_2 occurs in the corresponding games, and hence we have $\Pr[\mathsf{S}_1 \wedge \overline{\mathsf{C}_1}] = \Pr[\mathsf{S}_2 \wedge \overline{\mathsf{C}_2}]$ and $\Pr[\mathsf{C}_1] = \Pr[\mathsf{C}_2]$. Furthermore, it is clear that $\Pr[\mathsf{S}_1 | \mathsf{C}_1] = \Pr[\mathsf{S}_2 | \mathsf{C}_2] = 1/2$, because if $\mathtt{DUPCHK}(C^*) = 1$ occurs, then $\mathcal{A}$'s view is independent of the challenge bit b in Game 1, while a random bit b' is used to judge whether $\mathcal{A}$ succeeds in Game 2. Therefore, we have $\Pr[\mathsf{S}_1] = \Pr[\mathsf{S}_2]$, as required. □ (**Claim 15**)

Claim 16 $\Pr[\mathsf{S}_2] = \Pr[\mathsf{S}_3]$.

Proof of Claim 16. The difference between Game 2 and Game 3 is in how the decryption result m' of the ciphertext C' that is finally output by $\mathcal{A}_2$, is computed. More specifically, in Game 2, m' is computed in exactly the same way as in Game 1 (which is the original $\mathtt{wRNM\text{-}DCCA}$ experiment). However, in Game 3, for every $i \in [n]$, if there is $j \in [n]$ such that $c'_i = c^*_j$, then the bit s^*_j is directly used as the decryption result of c'_i. However, since each c^*_i is an encryption of s^*_i in both Game 2 and Game 3, s'_i is exactly s^*_j in both games. Therefore, from $\mathcal{A}$'s viewpoint, these games are identical, which implies the claim. □ (**Claim 16**)

Claim 17 *There exists a PPTA $\mathcal{B}_{\mathtt{p}}$ such that* $\mathsf{Adv}^{\mathtt{DCCA}}_{\Pi^n_{\mathtt{BE}},\mathcal{B}_{\mathtt{p}}}(k) = |\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$.

Proof of Claim 17. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}_{\mathtt{p}}$ that attacks the $\mathtt{DCCA}$ security of the bitwise-encrypt construction $\Pi^n_{\mathtt{BE}}$ with the claimed advantage. The description of $\mathcal{B}_{\mathtt{p}} = (\mathcal{B}_{\mathtt{p1}}, \mathcal{B}_{\mathtt{p2}})$ is as follows:

$\mathcal{B}_{\mathtt{p1}}^{\mathsf{Dec}^n_{\mathtt{BE}}(sk,\cdot)}(pk)$**:** $\mathcal{B}_{\mathtt{p1}}$ runs $\mathcal{A}_1(pk)$, where $\mathcal{B}_{\mathtt{p1}}$ answers $\mathcal{A}_1$'s decryption queries in a straightforward manner using $\mathcal{B}_{\mathtt{p1}}$'s own decryption oracle.

When $\mathcal{A}_1$ terminates with output (m_0, m_1, st), $\mathcal{B}_{\mathtt{p1}}$ picks a fair coin $b \in \{0,1\}$, and runs $s^* = (s^*_1 \| \dots \| s^*_n) \leftarrow \mathsf{E}(1^k, m_b)$. Then $\mathcal{B}_{\mathtt{p1}}$ sets $M_1 \leftarrow s^*$, $M_0 \leftarrow 0^n$, and $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathtt{p1}}$'s entire view), and terminates with output $(M_0, M_1, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathtt{p2}}^{\mathsf{Dec}^n_{\mathtt{BE}}(sk,\cdot)}(\mathsf{st}_{\mathcal{B}}, C^* = (c^*_1, \dots, c^*_n))$**:** If $\mathtt{DUPCHK}(C^*) = 1$, then $\mathcal{B}_{\mathtt{p2}}$ picks a fair coin $\gamma' \in \{0,1\}$, and terminates with output γ'. Otherwise, $\mathcal{B}_{\mathtt{p2}}$ runs $\mathcal{A}_2(\mathsf{st}, C^*)$.

$\mathcal{A}_2$'s decryption queries are answered as the decryption oracle in Game 3 does by using $\mathcal{B}_{\mathtt{p2}}$'s own decryption oracle, which is possible because $\mathcal{A}_2$'s queries C are assumed to satisfy $\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C) = 0$, which guarantees that $\mathcal{B}_{\mathtt{p2}}$ never falls into the situation in which it has to submit a prohibited query satisfying $\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C) = 1$ to the oracle.

When $\mathcal{A}_2$ terminates with output $(\mathsf{st}', C' = (c'_1, \dots, c'_n))$, $\mathcal{B}_{\mathtt{p2}}$ performs the following procedure:

1. Let $\mathtt{QUOTING} := \{i | c'_i \in C^*\}$, and let a be the smallest index in the set $[n] \backslash \mathtt{QUOTING}$. (If $\mathtt{QUOTING} = [n]$, then this step is skipped, and go to the step 3.) $\mathcal{B}_{\mathtt{p2}}$ needs to decrypt c'_i for $i \in [n] \backslash \mathtt{QUOTING}$, with the help of its own decryption oracle, but without violating the rule of the $\mathtt{DCCA}$ experiment (i.e. $\mathcal{B}_{\mathtt{p2}}$ is not allowed to submit a query C such that $\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C') = 1$). To accomplish this, let $C'' = (c''_1, \dots, c''_n)$ be a ciphertext of $\Pi^n_{\mathtt{BE}}$ such that for every $i \in [n]$, $c''_i = c'_i$ if $c'_i \notin C^*$ and $c''_i = c'_a$ if $c'_i \in C^*$. Note that it is guaranteed that $\mathsf{Dec}^n_{\mathtt{BE}}(sk, C'') = \bot$ holds if and only if $\mathsf{Dec}^n_{\mathtt{BE}}(sk, C') = \bot$. [12] Note also that $c''_i \notin C^*$

[12] This is because all quoted ciphertexts $c'_i \in C^*$ satisfy $\mathsf{Dec}_1(sk, c'_i) \neq \bot$, and hence $\mathsf{Dec}^n_{\mathtt{BE}}(sk, C'') = \bot$ if and only if there exists $i \in [n]$ such that $c'_i \notin C^*$ and $\mathsf{Dec}_1(sk, c'_i) = \bot$, which is exactly the same as the condition of $\mathsf{Dec}^n_{\mathtt{BE}}(sk, C') = \bot$.

holds for every $i \in [n]$, and hence it is guaranteed that $\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C'') = 0$. $\mathcal{B}_{\mathrm{p}2}$ submits C'' as a decryption query to its own decryption oracle, and receives the result $s'' = (s''_1 \| \dots \| s''_n)$ or $s'' = \bot$.

2. If $s'' = \bot$ or $s''_i = \bot$ for some $i \in [n]$, then $\mathcal{B}_{\mathrm{p}2}$ sets $m' \leftarrow \bot$, and directly goes to the step 6.
3. Otherwise (i.e. $s'' \neq \bot$ and every s''_i is non-$\bot$), for every $i \in [n]$, $\mathcal{B}_{\mathrm{p}2}$ sets

$$s'_i \leftarrow \begin{cases} s^*_j & \text{if } c'_i = c^*_j \text{ for some } j \in [n] \\ s''_i & \text{otherwise} \end{cases},$$

and then sets $s' \leftarrow (s'_1 \| \dots \| s'_n)$.

4. $\mathcal{B}_{\mathrm{p}2}$ computes $m' \leftarrow \mathsf{D}(1^k, s')$.
5. If $m' \in \{m_0, m_1\}$, then $\mathcal{B}_{\mathrm{p}2}$ sets $m' \leftarrow \mathsf{same}$.
6. Finally, $\mathcal{B}_{\mathrm{p}2}$ runs $b' \leftarrow \mathcal{A}_3(\mathsf{st}', m')$, and terminates with output $\gamma' \leftarrow (b' \stackrel{?}{=} b)$.

The above completes the description of $\mathcal{B}_{\mathrm{p}}$. Note that as explained above, $\mathcal{B}_{\mathrm{p}2}$ never submits a prohibited query C such that $\mathsf{F}^n_{\mathtt{BE}}(pk, C^*, C) = 1$ to the decryption oracle.

Let $\gamma \in \{0, 1\}$ be $\mathcal{B}_{\mathrm{p}}$'s challenge bit. $\mathcal{B}_{\mathrm{p}}$'s $\mathtt{DCCA}$ advantage is estimated as follows:

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

It is not hard to see that if $\gamma = 1$, then $\mathcal{B}_{\mathrm{p}}$ simulates Game 3 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b. Specifically, in this case, the challenge ciphertext C^* encrypts a codeword s^* of m_b, and is exactly how it is generated in Game 3. Furthermore, $\mathcal{A}_2$'s decryption queries are perfectly answered as in Game 3. It is also not hard to see that $\mathcal{B}_{\mathrm{p}2}$ computes the decryption result m' of the ciphertext C' (which is finally output by $\mathcal{A}_2$) by appropriately using the decryption oracle (without violating the rule of the $\mathtt{DCCA}$ experiment). Under the situation, the probability that $b' = b$ occurs in the experiment simulated by $\mathcal{B}_{\mathrm{p}}$ is exactly the same as the probability that $b' = b$ occurs in Game 3. That is, we have $\Pr[b' = b|\gamma = 1] = \Pr[\mathsf{S}_3]$.

When $\gamma = 0$, on the other hand, $\mathcal{B}_{\mathrm{p}}$ simulates Game 4 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b. In particular, in this case, C^* is an encryption of 0^n, which is exactly how it is generated in Game 4. The rest is unchanged from the case of $\gamma = 1$, and hence, with a similar argument to the above, we have $\Pr[b' = b|\gamma = 0] = \Pr[\mathsf{S}_4]$.

In summary, we have $\mathsf{Adv}^{\mathtt{DCCA}}_{\Pi^n_{\mathtt{BE}},\mathcal{B}_{\mathrm{p}}}(k) = |\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$, as required. □ (**Claim 17**)

Claim 18 *There exists a PPTA $\mathcal{B}_{\mathrm{n}}$ such that* $\mathsf{Adv}^{\mathcal{Q}\text{-}\mathtt{NM}}_{\mathcal{C},\mathcal{B}_{\mathrm{n}}}(k) = 2 \cdot |\Pr[\mathsf{S}_4] - 1/2|$.

Proof of Claim 18. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}_{\mathrm{n}}$ that attacks the $\mathcal{Q}$-non-malleability of the code $\mathcal{C}$ with the claimed advantage. The description of $\mathcal{B}_{\mathrm{n}} = (\mathcal{B}_{\mathrm{n}1}, \mathcal{B}_{\mathrm{n}2})$ is as follows:

$\mathcal{B}_{\mathrm{n}1}(1^k)$**:** $\mathcal{B}_{\mathrm{n}1}$ first runs $(pk, sk) \leftarrow \mathsf{PKG}_1(1^k)$, $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}_{\mathtt{EtBE}}(sk,\cdot)}(pk)$, and $C^* = (c^*_1, \dots, c^*_n) \leftarrow \mathsf{Enc}^n_{\mathtt{BE}}(pk, 0^n)$. If $\mathtt{DUPCHK}(C^*) = 1$, then $\mathcal{B}_{\mathrm{n}1}$ prepares the state information $\mathsf{st}_{\mathcal{B}}$ that tells $\mathcal{B}_{\mathrm{n}2}$ that $\mathcal{B}_{\mathrm{n}1}$ has given up, and terminates with output $(\bot, \bot, \bot, \mathsf{st}_{\mathcal{B}})$.

Otherwise (i.e. $\mathtt{DUPCHK}(C^*) = 0$), $\mathcal{B}_{\mathtt{n1}}$ runs $(C' = (c'_1, \dots, c'_n), \mathsf{st}') \leftarrow \mathcal{A}_2^{\mathsf{Dec}_{\mathtt{EtBE}}(sk,\cdot)}(\mathsf{st}, C^*)$. Then, for every $i \in [n]$ such that $c'_i \notin C^*$, $\mathcal{B}_{\mathtt{in1}}$ computes $s'_i = \mathsf{Dec}_1(sk, c'_i)$. If some of s'_i is $\bot$, then it is guaranteed that $\mathsf{Dec}_{\mathtt{EtBE}}(sk, C') = \bot$, and thus $\mathcal{B}_{\mathtt{n1}}$ has already computed $m' = \bot$, in which case $\mathcal{B}_{\mathtt{n1}}$ terminates with output $(\bot, \bot, \bot, \mathsf{st}_{\mathcal{B}})$.

Otherwise (i.e. all of s'_i's are non-$\bot$), $\mathcal{B}_{\mathtt{n1}}$ proceeds to the preparation of a function $f \in \mathcal{Q}_n$ used for the $\mathcal{Q}$-$\mathtt{NM}$ experiment. Specifically, $\mathcal{B}_{\mathtt{n1}}$ defines n functions $f_1, \dots, f_n \in \{\mathtt{one}, \mathtt{zero}\} \cup \{\mathtt{quote}^j\}_{j \in [n]}$ as follows:[13] For every $i \in [n]$, let

$$f_i := \begin{cases} \mathtt{quote}^j & \text{if } c'_i = c^*_j \text{ for some } j \in [n] \\ \mathtt{one} & \text{if } c'_i \notin C^* \text{ and } s'_i = 1 \\ \mathtt{zero} & \text{if } c'_i \notin C^* \text{ and } s'_i = 0 \end{cases},$$

and let $f : \{0,1\}^n \to \{0,1\}^n$ be the function defined by $f(x) := (f_1(x) \| \dots \| f_n(x))$. Note that since we are assuming $\mathtt{DUPCHK}(C') = 0$, the position $j \in [n]$ such that $c'_i = c^*_j$ must be unique if such j exists. Therefore, $f \in \mathcal{Q}_n$.

Finally, $\mathcal{B}_{\mathtt{n1}}$ sets $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathtt{n1}}$'s entire view), and terminates with output $(f, m_0, m_1, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathtt{n2}}(\mathsf{st}_{\mathcal{B}}, m'')$: $\mathcal{B}_{\mathtt{n2}}$ first checks if $\mathcal{B}_{\mathtt{n1}}$ has given up by checking $\mathsf{st}_{\mathcal{B}}$, in which case $\mathcal{B}_{\mathtt{n2}}$ picks a random bit $b' \in \{0,1\}$ and terminates with output b'. Otherwise, $\mathcal{B}_{\mathtt{n2}}$ also checks if the decryption result m' of C' is already known to be $\bot$ (because $s' = \bot$). If this is *not* the case, then $\mathcal{B}_{\mathtt{n2}}$ sets $m' \leftarrow m''$. Finally, $\mathcal{B}_{\mathtt{n2}}$ runs $b' \leftarrow \mathcal{A}_3(\mathsf{st}', m')$, and terminates with output b'.

The above completes the description of $\mathcal{B}_{\mathtt{n}}$.

Let $b \in \{0,1\}$ be $\mathcal{B}_{\mathtt{n}}$'s challenge bit. We argue that $\mathcal{B}_{\mathtt{n}}$ simulates Game 4 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b. Specifically, if $\mathcal{B}_{\mathtt{n1}}$ does not give up, then the challenge ciphertext C^* is generated as in Game 4, and the probability that $\mathcal{B}_{\mathtt{n1}}$ gives up is exactly the same as the probability that Game 4 terminates at the step of generating the challenge ciphertext C^*. Furthermore, unless $s' = \bot$ (in which case $m' = \bot$ is already known to $\mathcal{B}_{\mathtt{n1}}$), the value m'' that $\mathcal{B}_{\mathtt{n2}}$ receives as input is computed identically to m' computed in Game 4. Specifically, in this case, the $\mathcal{Q}$-$\mathtt{NM}$ experiment computes $s^* = (s^*_1, \dots, s^*_n) \leftarrow \mathsf{E}(1^k, m_b)$, $s' \leftarrow f(s^*)$, $m'' \leftarrow \mathsf{D}(1^k, s')$, and if $m'' \in \{m_0, m_1\}$, m'' is replaced with same. Here, by our design of $f \in \mathcal{Q}_n$, it is guaranteed that $s' = f(s^*)$ computed above equals to the one that is computed in Game 4, and hence m'' given to $\mathcal{B}_{\mathtt{n2}}$ is exactly the one that should be input to $\mathcal{A}_3$ in Game 4. Therefore, under this situation, the probability that $b' = b$ occurs in the experiment simulated by $\mathcal{B}_{\mathtt{n}}$ is exactly the same as the probability that S_4 occurs in Game 4. Thus, $\mathcal{B}_{\mathtt{n}}$'s advantage in the $\mathcal{Q}$-$\mathtt{NM}$ experiment can be calculated as follows:

$$\mathsf{Adv}^{\mathcal{Q}\text{-}\mathtt{NM}}_{\mathcal{C}, \mathcal{B}_{\mathtt{n}}}(k) = 2 \cdot \left| \Pr[b' = b] - \frac{1}{2} \right| = 2 \cdot \left| \Pr[\mathsf{S}_4] - \frac{1}{2} \right|.$$

□ (**Claim 18**)

Claims 15 to 18 and Equation (12) guarantee that there exist PPTAs $\mathcal{B}_{\mathtt{p}}$ and $\mathcal{B}_{\mathtt{n}}$ satisfying Equation (11). By the assumptions that Π_1 is $\mathtt{CCA}$ secure and that $\mathcal{C}$ is $\mathcal{Q}$-non-malleable, combined with Lemma 6, we can conclude that $\mathsf{Adv}^{\mathtt{CCA}}_{\Pi_{\mathtt{EtBE}}, \mathcal{A}}(k)$ is negligible. Recall that the choice of $\mathcal{A}$ was arbitrarily, and thus for any PPTA $\mathcal{A}$, we can show a negligible upperbound on $\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{EtBE}}, \mathcal{A}}(k)$. Therefore, $\Pi_{\mathtt{EtBE}}$ is $\mathtt{wRNM\text{-}DCCA}$ secure. □ (**Lemma 8**)

[13] Recall that every function f in $\mathcal{Q}_n$ can be described by n functions $f_1, \dots, f_n \in \{\mathtt{one}, \mathtt{zero}\} \cup \{\mathtt{quote}^j\}_{j \in [n]}$, so that $f(x) = (f_1(x) \| \dots \| f_n(x))$ holds for all $x \in \{0,1\}^n$. Furthermore, f needs to satisfy the "no duplicated quoting" condition (Equation (1)), namely, there exist no indices $i, i', j \in [n]$ such that $f_i = f_{i'} = \mathtt{quote}^j$ and $i \neq i'$.

D.7 Proof of Theorem 5: A CCA Attack on the "One-Key-Pair" Construction $\widetilde{\Gamma}'$

For notational convenience, for a string $x = (x_1 \| \dots \| x_k) \in \{0,1\}^k$ and positions $i, j \in [k]$ with $i \le j$, we define $x^{[i:j]} := (x_i \| \dots \| x_j) \in \{0,1\}^{j-i+1}$, and we extend the definition for the cases $(i,j) = (1,0)$ and $(i,j) = (k+1,k)$ so that $x^{[1:0]}$ and $x^{[(k+1):k]}$ are defined to be the empty string. Furthermore, for a string x which is longer than k-bits, we denote by $\mathsf{LSB}_k(x)$ the least significant k-bits of x.

Call a string $x \in \{0,1\}^{k-1}$ *sensitive* if

$$\mathsf{LSB}_k(\mathsf{G}(x^{[1:(i^\star-1)]} \| 1 \| x^{[(i^\star+1):k]})) \neq \mathsf{LSB}_k(\mathsf{G}(x^{[1:(i^\star-1)]} \| 0 \| x^{[(i^\star+1):k]}))$$

holds for some $i^\star \in [k]$, and call the position $i^\star \in [k]$ the *sensitive position* of x. (If there are multiple such positions, the sensitive position is defined as the smallest one among them.)

We first introduce the following procedure "Embed" which takes as input (1) a sensitive string $K^\star_{\mathtt{in}} \in \{0,1\}^{k-1}$, (2) the sensitive position $i^\star$ of $K^\star_{\mathtt{in}}$, (3) a public key pk, and (4) any ciphertext c' (of the underlying 1-bit PKE scheme Π_1), and outputs a ciphertext/session-key pair (C, K) into which c' is "embedded" as one of the "inner" ciphertexts, as follows:

$\mathsf{Embed}(K^\star_{\mathtt{in}}, i^\star, pk, c')$**:** Set $K_{\mathtt{in}} = (K^{(1)}_{\mathtt{in}} \| \dots \| K^{(k)}_{\mathtt{in}}) \leftarrow (K^{\star[1:(i^\star-1)]}_{\mathtt{in}} \| 1 \| K^{\star[(i^\star+1):k]}_{\mathtt{in}})$ and $c^{(i^\star)}_{\mathtt{in}} \leftarrow c'$. Next, compute $c^{(i)}_{\mathtt{in}} \leftarrow \mathsf{Enc}_1(pk, K^{(i)}_{\mathtt{in}})$ for every $i \in [k] \backslash \{i^\star\}$. Then, execute the rest of the procedure of $\widetilde{\mathsf{Encap}}'(pk)$ (i.e. from the step of computing $\alpha \leftarrow \mathsf{G}(K_{\mathtt{in}})$) using $K_{\mathtt{in}}$ and $c^{(1)}_{\mathtt{in}}, \dots, c^{(k)}_{\mathtt{in}}$ that have been already computed, and output a ciphertext $C = (c_1, \dots, c_n, \widehat{c})$ and the corresponding session-key K.

The following shows that Embed can be used to decrypt an honestly generated ciphertext c', given access to the decapsulation oracle of $\widetilde{\Gamma}'$:

Claim 19 *Let $K^\star_{\mathtt{in}} \in \{0,1\}^{k-1}$ be a sensitive string and $i^\star$ be the corresponding sensitive position. Furthermore, let (pk, sk) be a key pair output by $\mathsf{PKG}_1(1^k)$ and c' be a ciphertext output by $\mathsf{Enc}_1(pk, \cdot)$. If (C, K) is output by $\mathsf{Embed}(K^\star_{\mathtt{in}}, i^\star, pk, c')$, then it holds that $\mathsf{Dec}_1(sk, c') = (\widetilde{\mathsf{Decap}}'(sk, C) \stackrel{?}{=} K)$.*

Proof of Claim 19. Let $K^\star_{\mathtt{in}}$, $i^\star$, pk, and c' be as stated in the lemma, and suppose (C, K) is output from $\mathsf{Embed}(K^\star_{\mathtt{in}}, i^\star, pk, c')$. We will show that if c' is an encryption of 1, then $\widetilde{\mathsf{Decap}}'(sk, C) = K$ holds, while if c' is an encryption of 0, then this does not hold, which proves the claim.

This can be easily seen by checking the behavior of $\widetilde{\mathsf{Decap}}'(sk, C)$ step by step. Specifically, by the definition of how C is generated in Embed, the value $K_{\mathtt{in}} \in \{0,1\}^k$ recovered in the procedure of $\widetilde{\mathsf{Decap}}'(sk, C)$ satisfies $K_{\mathtt{in}} = (K^{\star[1:(i^\star-1)]}_{\mathtt{in}} \| \mathsf{Dec}_1(sk, c') \| K^{\star[(i^\star+1):k]}_{\mathtt{in}})$. (Since c' is generated by $\mathsf{Enc}_1(pk, \cdot)$, we have $\mathsf{Dec}_1(sk, c') \in \{0,1\}$.) Since Embed generates the "outer" encryptions $c_1, \dots c_n$ and $\widehat{c}$ as if $K_{\mathtt{in}} = (K^{\star[1:(i^\star-1)]}_{\mathtt{in}} \| 1 \| K^{\star[(i^\star+1):k]}_{\mathtt{in}})$ is used, if c' is an encryption of 1, then the validity check by re-encryption performed in the last step is also passed and the output of $\widetilde{\mathsf{Decap}}'(sk, C)$ is equal to $K = \mathsf{LSB}_k(\mathsf{G}(K_{\mathtt{in}}))$. On the other hand, if c' is an encryption of 0, then the $i^\star$-th bit of $K_{\mathtt{in}}$ computed in $\widetilde{\mathsf{Decap}}'(sk, C)$ is 0. However, by the property of the sensitive string $K^\star_{\mathtt{in}}$, it is guaranteed that $\mathsf{LSB}_k(\mathsf{G}(K^{\star[1:(i^\star-1)]}_{\mathtt{in}} \| 0 \| K^{\star[(i^\star+1):k]}_{\mathtt{in}})) \neq \mathsf{LSB}_k(\mathsf{G}(K^{\star[1:(i^\star-1)]}_{\mathtt{in}} \| 1 \| K^{\star[(i^\star+1):k]}_{\mathtt{in}})) = K$, and thus the output of $\widetilde{\mathsf{Decap}}'(sk, C)$ is either some value different from K or $\perp$ (due to the failure of the validity check). □ (**Claim 19**)

Next, we show that if G is a PRG, then one can efficiently find a sensitive string together with the corresponding sensitive position with overwhelming probability. Specifically, consider the following procedure Find:

$\mathsf{Find}(1^k)$**:** Firstly, pick two strings $x = (x_1\|\dots\|x_k) \in \{0,1\}^k$ and $x' = (x'_1\|\dots\|x'_k) \in \{0,1\}^k$ uniformly at random, and abort if $\mathsf{LSB}_k(\mathsf{G}(x)) = \mathsf{LSB}_k(\mathsf{G}(x'))$. Otherwise, from $i = 1$ to k, check if $\mathsf{LSB}_k(\mathsf{G}(x^{[1:(i-1)]}\|x'^{[i:k]})) \neq \mathsf{LSB}_k(\mathsf{G}(x^{[1:i]}\|x'^{[(i+1):k]}))$ holds. If it holds for some $i \in [k]$, then let $i^\star$ of the smallest position found, set $x^\star \leftarrow (x^{[1:(i^\star-1)]}\|x'^{[(i^\star+1):k]})$, and return $(x^\star, i^\star)$ as a pair of a sensitive string and the corresponding sensitive position. Otherwise (i.e. no position $i \in [k]$ satisfying the above check was found), give up and abort.

Claim 20 *If* G *is a PRG, then* $\mathsf{Find}(1^k)$ *outputs a sensitive string* $x^\star \in \{0,1\}^{k-1}$ *and the corresponding sensitive position* $i^\star \in [k]$ *with overwhelming probability.*

Proof of Claim 20. We first show that if we pick two strings $x, x' \in \{0,1\}^k$ uniformly at random, then the probability that $\mathsf{LSB}_k(\mathsf{G}(x)) = \mathsf{LSB}_k(\mathsf{G}(x'))$ occurs, which we denote by p, is upperbounded by $\mathsf{Adv}^{\mathtt{PRG}}_{\mathsf{G},\mathcal{B}}(k) + 2^{-k}$ for some PPTA $\mathcal{B}$, and thus is negligible. Specifically, consider the distinguisher $\mathcal{B}$ which is given 1^k and $y \in \{0,1\}^{\ell'}$ as input, picks a string $x' \in \{0,1\}^k$ uniformly at random, and returns the bit $(\mathsf{LSB}_k(\mathsf{G}(x')) \overset{?}{=} \mathsf{LSB}_k(y))$. Since $\mathcal{B}$ outputs 1 only when $\mathsf{LSB}_k(\mathsf{G}(x')) = \mathsf{LSB}_k(y)$ holds, $\mathcal{B}$'s advantage in distinguishing the output of G is as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{PRG}}_{\mathsf{G},\mathcal{B}}(k) &= \Big| \Pr_{x,x' \leftarrow \{0,1\}^k}[\mathsf{LSB}_k(\mathsf{G}(x')) = \mathsf{LSB}_k(\mathsf{G}(x))] \\ &\qquad - \Pr_{y \leftarrow \{0,1\}^{\ell'}, x' \leftarrow \{0,1\}^k}[\mathsf{LSB}_k(\mathsf{G}(x')) = \mathsf{LSB}_k(y)] \Big| \\ &= p - 2^{-k},\end{aligned}$$

which implies $p \leq \mathsf{Adv}^{\mathtt{PRG}}_{\mathsf{G},\mathcal{B}}(k) + 2^{-k}$, as required.

Next, given two strings x, x' satisfying $\mathsf{LSB}_k(\mathsf{G}(x)) \neq \mathsf{LSB}_k(\mathsf{G}(x'))$, it is easy to see that there must exist at least one position $i^\star$ such that $\mathsf{LSB}_k(\mathsf{G}(x^{[1:(i^\star-1)]}\|x'^{[i^\star:k]})) \neq \mathsf{LSB}_k(\mathsf{G}(x^{[1:i^\star]}\|x'^{[(i^\star+1):k]}))$, because otherwise $\mathsf{LSB}_k(\mathsf{G}(x)) = \mathsf{LSB}_k(\mathsf{G}(x'))$ holds, which is a contradiction. Furthermore, under this $i^\star$, we have $x_{i^\star} \neq x'_{i^\star}$. Therefore, no matter whether $(x_{i^\star}, x'_{i^\star}) = (0,1)$ or $(1,0)$, $x^\star = (x^{[1:(i^\star-1)]}\|x'^{[(i^\star+1):k]})$ is a sensitive string and $i^\star$ is the corresponding sensitive position.

Therefore, we can conclude that with overwhelming probability, $\mathsf{Find}(1^k)$ outputs a sensitive string and its corresponding sensitive position. □ (**Claim 20**)

Now, armed with Claims 19 and 20, consider the following $\mathtt{CCA}$ adversary $\mathcal{A}$ against the KEM $\widetilde{\Gamma}'$: $\mathcal{A}$ is initially given a public key pk and the challenge ciphertext $C^* = (c^*_1, \dots, c^*_n, \widehat{c}^*)$, where $\widehat{c}^* = \mathsf{SEnc}(K^*_{\mathtt{out}}, (c^{*(1)}_{\mathtt{in}}\|\dots\|c^{*(k)}_{\mathtt{in}}))$. Note that each of c^*_i and $c^{*(i)}_{\mathtt{in}}$ is guaranteed to be in the range of $\mathsf{Enc}_1(pk, \cdot)$ by the definition of $\widetilde{\mathsf{Encap}}'(pk)$. $\mathcal{A}$ first runs $\mathsf{Find}(1^k)$ to find a sensitive string $K^\star_{\mathtt{in}} \in \{0,1\}^{k-1}$ together with its sensitive position $i^\star$, which succeeds with overwhelming probability due to Claim 20. Next, $\mathcal{A}$ decrypts each of the "outer" ciphertexts $c^*_1, \dots, c^*_n$ by running $(C_i, K_i) \leftarrow \mathsf{Embed}(K^\star_{\mathtt{in}}, i^\star, pk, c^*_i)$ and then checking whether $\widetilde{\mathsf{Decap}}'(sk, C_i) = K_i$ holds, where $\widetilde{\mathsf{Decap}}'(sk, C_i)$ is performed by using $\mathcal{A}$'s decapsulation oracle. By Claim 19, $\mathcal{A}$ obtains

$s^* = (s_1^* \| \dots \| s_n^*)$. Then $\mathcal{A}$ runs $K^*_{\mathtt{out}} \leftarrow \mathsf{D}(1^k, s^*)$, and then $\mathsf{SDec}(K^*_{\mathtt{out}}, \widehat{c}^*)$, to obtain the "inner" ciphertexts $c^{*(1)}_{\mathtt{in}}, \dots, c^{*(k)}_{\mathtt{in}}$. Now, using the same procedure of using Embed and the decapsulation oracle $\widetilde{\mathsf{Decap}}'(sk, \cdot)$, $\mathcal{A}$ decrypts each of the inner ciphertexts $c^{*(1)}_{\mathtt{in}}, \dots, c^{*(k)}_{\mathtt{in}}$ to recover $K^*_{\mathtt{in}} = (K^{*(1)}_{\mathtt{in}} \| \dots \| K^{*(k)}_{\mathtt{in}})$, from which $\mathcal{A}$ can recover the session-key K^* corresponding to C^* by computing $K^* = \mathsf{LSB}_k(\mathsf{G}(K^*_{\mathtt{in}}))$. The probability that $\mathcal{A}$ succeeds in recovering the session key K^* corresponding to C^* (and thus breaks even the one-wayness of Γ' under $\mathtt{CCA}$) is exactly the probability that $\mathsf{Find}(1^k)$ succeeds, which is overwhelming. □ (**Theorem 5**)

E Other Miscellaneous Proofs and Facts

E.1 Proof of Lemma 2: $\mathcal{Q} \subseteq \mathcal{P}$

Fix $n \in \mathbb{N}$. We will show that for any $f \in \mathcal{Q}_n$, it holds that $f \in \mathcal{P}_n$. To this end, fix arbitrarily a function $f \in \mathcal{Q}_n$. By definition of $\mathcal{Q}_n$, there exist n functions $f_1, \dots, f_n : \{0,1\}^n \to \{0,1\}$ such that $f(x) = (f_1(x) \| \dots \| f_n(x))$ holds for all $x \in \{0,1\}^n$, and $f_1, \dots, f_n$ satisfy the "no duplicated quoting" condition (Equation (1)), namely, there exist no indices $i, i', j \in [n]$ such that $f_i = f_{i'} = \mathtt{quote}^j$ and $i \neq i'$.

In the following, we show that there exists a corresponding function $\widehat{f} \in \mathcal{P}_n$ that satisfies $\widehat{f}(x) = f(x)$ for every $x \in \{0,1\}^n$. Recall that all functions in $\mathcal{P}_n$ have corresponding bitwise tampering functions $\widehat{f}_1, \dots, \widehat{f}_n \in \mathcal{F}_{\mathtt{BIT}}$ and a permutation $\pi : [n] \to [n]$. Thus, below we show a procedure for specifying such $\widehat{f}_1, \dots, \widehat{f}_n$ and π, thereby construct $\widehat{f} \in \mathcal{P}_n$:

1. Let $\mathtt{QUOTING} := \{i | f_i \in \{\mathtt{quote}^j\}_{j \in [n]}\}$ and $\mathtt{QUOTED} := \{j | \exists i \in [n] : f_i = \mathtt{quote}^j\}$. (Intuitively, $\mathtt{QUOTING}$ is the subset of $[n]$ indicating the positions in the output of f that quote one of the positions in the input. On the other hand, $\mathtt{QUOTED}$ is the subset of $[n]$ indicating the positions in the input to f that are quoted in one of the output bits of f. Note that $|\mathtt{QUOTING}| = |\mathtt{QUOTED}|$ holds, because there are no positions $i, i', j \in [n]$ such that $f_i = f_{i'} = \mathtt{quote}^j$ and $i \neq i'$.)
2. Initialize a function π, whose domain is $[n]$, such that $\pi(i) := \perp$ for all $i \in [n]$. (In the following steps, π will be made a permutation over $[n]$.)
3. For every $i \in \mathtt{QUOTING}$, define $\pi^{-1}(i) := j$ where j is such that $f_i = \mathtt{quote}^j$. (Note that due to the "no duplicated quoting" condition (Equation (1)), such j is unique for every $i \in \mathtt{QUOTING}$. At this point, only the mapping from $\mathtt{QUOTED}$ to $\mathtt{QUOTING}$ is defined in π.)
4. Let $\phi : [n] \backslash \mathtt{QUOTING} \to [n] \backslash \mathtt{QUOTED}$ be any injection, and define $\pi^{-1}(i) = \phi(i)$ for every $i \in [n] \backslash \mathtt{QUOTING}$. (At this point, π becomes a permutation over $[n]$.)
5. For every $i \in [n]$, define $\widehat{f}_i \in \mathcal{F}_{\mathtt{BIT}}$ as follows:
$$\widehat{f}_i := \begin{cases} \mathtt{forward} & \text{if } f_{\pi(i)} \in \{\mathtt{quote}^j\}_{j \in [n]} \\ \mathtt{set} & \text{if } f_{\pi(i)} = \mathtt{one} \\ \mathtt{reset} & \text{if } f_{\pi(i)} = \mathtt{zero} \end{cases} . \tag{13}$$
6. Let $\widehat{f} : \{0,1\}^n \to \{0,1\}^n$ be the function that is defined using $\widehat{f}_1, \dots, \widehat{f}_n$ and π as follows:
$$\widehat{f}(x) := \Big(\widehat{f}_{\pi^{-1}(1)}(x_{\pi^{-1}(1)}) \,\|\, \dots \,\|\, \widehat{f}_{\pi^{-1}(n)}(x_{\pi^{-1}(n)}) \Big), \tag{14}$$
where $x = (x_1 \| \dots \| x_n) \in \{0,1\}^n$.

The above completes the procedure for defining $\widehat{f}$. By definition, we have $\widehat{f} \in \mathcal{P}$.

We now show that f and $\widehat{f}$ are equivalent as a function. i.e. $f(x) = \widehat{f}(x)$ for every $x \in \{0,1\}^n$: Let $x = (x_1 \| \dots \| x_n) \in \{0,1\}^n$. Then, for each $i \in [n]$:

- If $f_i = \mathtt{one}$, then Equation (13) guarantees that $\widehat{f}_{\pi^{-1}(i)} = \mathtt{set}$, and thus the i-th bit of $\widehat{f}(x)$ is $\widehat{f}_{\pi^{-1}(i)}(x_{\pi^{-1}(i)}) = \mathtt{set}(x_{\pi^{-1}(i)}) = 1 = \mathtt{one}(x) = f_i(x)$.
- If $f_i = \mathtt{zero}$, then Equation (13) guarantees that $\widehat{f}_{\pi^{-1}(i)} = \mathtt{reset}$, and thus the i-th bit of $\widehat{f}(x)$ is $\widehat{f}_{\pi^{-1}(i)}(x_{\pi^{-1}(i)}) = \mathtt{reset}(x_{\pi^{-1}(i)}) = 0 = \mathtt{zero}(x) = f_i(x)$.
- If $f_i = \mathtt{quote}^j$ for some $j \in [n]$, then Equation (13) guarantees $\widehat{f}_{\pi^{-1}(i)} = \mathtt{forward}$ and the step 3 of the above procedure guarantees $\pi^{-1}(i) = j$, which in turn imply that the i-th bit of $\widehat{f}(x)$ is $\widehat{f}_{\pi^{-1}(i)}(x_{\pi^{-1}(i)}) = \mathtt{forward}(x_j) = x_j = \mathtt{quote}^j(x) = f_i(x)$.

Put everything together, for all $x \in \{0,1\}^n$, we have $f(x) = \widehat{f}(x)$. Since $\widehat{f} \in \mathcal{P}_n$, we have $f \in \mathcal{P}_n$.

We have seen that for every $n \in \mathbb{N}$, any function $f \in \mathcal{Q}_n$ also belongs to $\mathcal{P}_n$, and hence it follows that $\mathcal{Q}_n \subseteq \mathcal{P}_n$ for every $n \in \mathbb{N}$, which in turn implies $\mathcal{Q} \subseteq \mathcal{P}$. □ (**Lemma 2**)

E.2 Proof of Lemma 4: w(R)NM-DCCA Security and Randomness-Inextractability of $\Pi_{\mathtt{HYB}}$

Regarding the "non-malleability" notions, it is indeed straightforward to see that if Π is wNM-DCCA secure and E is CCA secure, then the detectable PKE scheme $\Pi_{\mathtt{HYB}}$ is wNM-DCCA secure, with essentially the same proof as the proof of the CCA security for ordinary hybrid encryption by Cramer and Shoup [8]. Thus, here we provide a proof for the (perhaps) less straightforward statement: If Π is wRNM-DCCA secure and E is CCA secure, then the detectable PKE scheme $\Pi_{\mathtt{HYB}}$ is wRNM-DCCA secure. (It turns out that it is somewhat complicated due to the "replayable" nature of wRNM-DCCA security.)

After it, we provide a proof for the randomness-inextractability of $\Pi_{\mathtt{HYB}}$.

Proof for* wRNM-DCCA *Security. We will show that for any PPTA adversary $\mathcal{A}$ that attacks the wRNM-DCCA security of $\Pi_{\mathtt{HYB}}$, there exist PPTAs $\mathcal{B}_{\mathsf{p}}$ and $\mathcal{B}_{\mathsf{e}}$ such that

$$\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{HYB}},\mathcal{A}}(k) \leq 2 \cdot 2^{-k} + 2 \cdot \mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi,\mathcal{B}_{\mathsf{p}}}(k) + \mathsf{Adv}^{\mathtt{CCA}}_{E,\mathcal{B}_{\mathsf{e}}}(k). \tag{15}$$

We show this via a sequence of games.

Fix any PPTA adversary $\mathcal{A}$ that attacks the wRNM-DCCA security of $\Pi_{\mathtt{HYB}}$. Consider the following sequence of games: (Here, the values with asterisk (*) represent those related to $\mathcal{A}$'s challenge ciphertext.)

Game 1: This is the experiment $\mathsf{Expt}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{HYB}},\mathcal{A}}(k)$ itself. For defining the subsequent games, we make Game 1 pick a string $\widehat{K} \in \{0,1\}^k$ uniformly at random at the beginning of the game. (This value does not appear in $\mathcal{A}$'s view in Game 1, and thus does not affect $\mathcal{A}$'s behavior at all.)

Game 2: Same as Game 1, except that the final output $C' = (c', \widetilde{c}')$ of $\mathcal{A}_2$ is processed (and the result m' is input to $\mathcal{A}_3$) as follows:

1. $K' \leftarrow \mathsf{Dec}(sk, c')$
2. If $K' = \bot$ then return $m' \leftarrow \bot$ (and the following steps are not performed).

3. $m' \leftarrow \begin{cases} \mathsf{same} & \text{if } K' = \widehat{K} \wedge \widehat{c}' = \widehat{c}^* \\ \mathsf{SDec}(K^*, \widehat{c}') & \text{if } K' = \widehat{K} \wedge \widehat{c}' \neq \widehat{c}^* \\ \mathsf{SDec}(K', \widehat{c}') & \text{if } K' \neq \widehat{K} \end{cases}$
4. If $m' \in \{m_0, m_1\}$ then $m' \leftarrow \mathsf{same}$.
5. Return m'.

Game 3: Same as Game 2, except that c^* is generated by "$c^* \leftarrow \mathsf{Enc}(pk, \widehat{K})$," instead of "$c^* \leftarrow \mathsf{Enc}(pk, K^*)$."

The above completes the description of the games. For $i \in [3]$, let S_i be the event that in Game i, $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs).

By definition, $\mathcal{A}$'s `wRNM-DCCA` advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi_{\mathtt{HYB}},\mathcal{A}}(k) = 2 \cdot \left|\Pr[\mathsf{S}_1] - \frac{1}{2}\right| \leq 2 \cdot \sum_{i \in [2]} \left|\Pr[\mathsf{S}_i] - \Pr[\mathsf{S}_{i+1}]\right| + 2 \cdot \left|\Pr[\mathsf{S}_3] - \frac{1}{2}\right|. \tag{16}$$

It remains to show the upperbound of each term that appears in the right hand side of the above inequality.

Claim 21 $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]| \leq 2^{-k}$.

Proof of Claim 21. Note that the difference between Game 1 and Game 2 is only in how the ciphertext $C' = (c', \widehat{c}')$ that is finally output by $\mathcal{A}_2$ is processed. More specifically, these games differ only if $\mathsf{Dec}(sk, c') = K' = \widehat{K}$ holds. Therefore, the difference $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]|$ can be upperbounded by the probability that $\mathcal{A}_2$ finally outputs such a ciphertext in Game 1. However, $\widehat{K}$ is chosen uniformly at random from $\{0,1\}^k$, and is information-theoretically hidden from $\mathcal{A}$'s view in Game 1. Thus, the probability that $\mathcal{A}_2$'s final output $C' = (c', \widehat{c}')$ satisfies $\mathsf{Dec}(sk, c') = \widehat{K}$ is at most 2^{-k}. Therefore, we have $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]| \leq 2^{-k}$. □ (**Claim 21**)

Claim 22 *There exists a PPTA $\mathcal{B}_\mathsf{p}$ such that* $\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi,\mathcal{B}_\mathsf{p}}(k) = |\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]|$.

Proof of Claim 22. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}_\mathsf{p}$ that attacks the `wRNM-DCCA` security of the underlying detectable PKE scheme Π with the claimed advantage. The description of $\mathcal{B}_\mathsf{p} = (\mathcal{B}_{\mathsf{p}1}, \mathcal{B}_{\mathsf{p}2}, \mathcal{B}_{\mathsf{p}3})$ is as follows:

$\mathcal{B}_{\mathsf{p}1}^{\mathsf{Dec}(sk,\cdot)}(pk)$**:** $\mathcal{B}_{\mathsf{p}1}$ runs $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(pk)$, where $\mathcal{B}_{\mathsf{p}2}$ answers $\mathcal{A}_1$'s decryption queries in a straightforward manner using its own decryption oracle $\mathsf{Dec}(sk, \cdot)$. Then, $\mathcal{B}_{\mathsf{p}1}$ picks $K^*, \widehat{K} \in \{0,1\}^k$ uniformly at random, sets $M_1 \leftarrow K^*$, $M_0 \leftarrow \widehat{K}$, and $\mathsf{st}_\mathcal{B} \leftarrow$ ($\mathcal{B}_{\mathsf{p}1}$'s entire view), and terminates with output $(M_0, M_1, \mathsf{st}_\mathcal{B})$.

$\mathcal{B}_{\mathsf{p}2}^{\mathsf{Dec}(sk,\cdot)}(\mathsf{st}_\mathcal{B}, c^*)$**:** $\mathcal{B}_{\mathsf{p}2}$ picks a fair coin $b \in \{0,1\}$, and then runs $\widehat{c}^* \leftarrow \mathsf{SEnc}(K^*, m_b)$. Then, $\mathcal{B}_{\mathsf{p}2}$ sets $C^* \leftarrow (c^*, \widehat{c}^*)$, and runs $\mathcal{A}_2(\mathsf{st}, C^*)$.

$\mathcal{B}_{\mathsf{p}2}$ answers the decryption queries from $\mathcal{A}_2$ by using its own decryption oracle $\mathsf{Dec}(sk, \cdot)$. Here, note that $\mathcal{A}_2$'s queries $C = (c, \widehat{c})$ must satisfy $\mathsf{F}_{\mathtt{HYB}}(pk, C^*, C) = \mathsf{F}(pk, c^*, c) = 0$, and thus $\mathcal{B}_{\mathsf{p}2}$ never falls into the situation in which it has to submit prohibited decryption queries c satisfying $\mathsf{F}(pk, c^*, c) = 1$.

When $\mathcal{A}_2$ terminates with output $(C' = (c', \widehat{c}'), \mathsf{st}')$, $\mathcal{B}_2$ sets $\mathsf{st}'_\mathcal{B} \leftarrow$ ($\mathcal{B}_{\mathsf{p}2}$'s entire view). Then, $\mathcal{A}_2$ outputs $(c', \mathsf{st}'_\mathcal{B})$ if $c' \neq c^*$ or $(\perp, \mathsf{st}'_\mathcal{B})$ otherwise, and terminates.

$\mathcal{B}_{\mathsf{p}3}(\mathsf{st}'_{\mathcal{B}}, M')$**:** (where $M' = \mathsf{same}$ or $M' = \mathsf{Dec}(sk, c')$) If $c' = c^*$, then $\mathcal{B}_{\mathsf{p}3}$ sets $M' \leftarrow \mathsf{same}$. Then, depending on the value of M', $\mathcal{B}_{\mathsf{p}3}$ proceeds as follows:

- If $M' = \bot$, then $\mathcal{B}_{\mathsf{p}3}$ sets $m' \leftarrow \bot$.
- If $M' = \mathsf{same}$ and $\widehat{c}^* = \widehat{c}'$, then $\mathcal{B}_{\mathsf{p}3}$ sets $m' \leftarrow \mathsf{same}$.
- If $M' = \mathsf{same}$ and $\widehat{c}^* \neq \widehat{c}'$, then $\mathcal{B}_{\mathsf{p}3}$ computes $m' \leftarrow \mathsf{SDec}(K^*, \widehat{c}')$.
- Otherwise (i.e. $M' \notin \{\bot, \mathsf{same}\}$), $\mathcal{B}_{\mathsf{p}3}$ interprets M' as the decryption result $K' = \mathsf{Dec}(sk, c') \in \{0,1\}^k$, and computes $m' \leftarrow \mathsf{SDec}(K', \widehat{c}')$.

Then, if $m' \in \{m_0, m_1\}$, then $\mathcal{B}_{\mathsf{p}3}$ sets $m' \leftarrow \mathsf{same}$. Finally, $\mathcal{B}_{\mathsf{p}3}$ runs $b' \leftarrow \mathcal{A}_3(\mathsf{st}', m')$, and terminates with output $\gamma' \leftarrow (b' \overset{?}{=} b)$.

The above completes the description of $\mathcal{B}_\mathsf{p}$. Note that $\mathcal{B}_{\mathsf{p}2}$ never submits a prohibited decryption query c such that $\mathsf{F}(pk, c^*, c) = 1$. Furthermore, $\mathcal{B}_{\mathsf{p}2}$'s final output c' is guaranteed to be distinct from c^*.

Let $\gamma \in \{0,1\}$ be $\mathcal{B}_\mathsf{p}$'s challenge bit. $\mathcal{B}_\mathsf{p}$'s $\mathtt{wRNM\text{-}DCCA}$ advantage is estimated as follows:

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

It is not hard to see that if $\gamma = 1$, then $\mathcal{B}_\mathsf{p}$ simulates Game 2 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b. Specifically, in this case c^* is an encryption of K^*, which is exactly how it is generated in Game 2. $\mathcal{A}_2$'s decryption queries are perfectly answered using $\mathcal{B}_{\mathsf{p}2}$'s decryption oracle. We argue that $\mathcal{A}_2$'s final output $C' = (c', \widehat{c}')$ is processed in exactly the same way as that in Game 2. To see this, if $M' \neq \mathsf{same}$ (which means $M' \notin \{K^*, \widehat{K}\}$), then $\mathcal{B}_{\mathsf{p}3}$ computes m' in the same way as in Game 2.; If $M' = \widehat{K}$, then m' is also computed as in Game 2.; If $M' = K^*$,[14] then we have two subcases: (1) If $\widehat{c}^* = \widehat{c}'$, then it is guaranteed that $\mathsf{SDec}(K^*, \widehat{c}') = m_b \in \{m_0, m_1\}$, and thus setting $m' \leftarrow \mathsf{same}$ directly is the correct behavior of Game 2. (2) If $\widehat{c}^* \neq \widehat{c}'$, then the step "$m' \leftarrow \mathsf{SDec}(K^*, \widehat{c}')$" correctly computes $m' = \mathsf{SDec}(K', \widehat{c}')$. Therefore, $\mathcal{A}_2$'s final output C' is processed in exactly the same way as that in Game 2. This means that 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. we have $\Pr[b' = b|\gamma = 1] = \Pr[\mathsf{S}_2]$.

When $\gamma = 0$, on the other hand, $\mathcal{B}_\mathsf{p}$ simulates Game 3 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is b. In particular, in this case, c^* is an encryption of $\widehat{K}$, and this is the only difference from the case $\gamma = 1$. Therefore, with essentially the same argument as the above, we have $\Pr[b' = b|\gamma = 0] = \Pr[\mathsf{S}_3]$.

In summary, we have $\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi,\mathcal{B}_\mathsf{p}}(k) = |\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]|$, as required. □ (**Claim 22**)

Claim 23 *There exists a PPTA $\mathcal{B}_\mathsf{e}$ such that* $\mathsf{Adv}^{\mathtt{CCA}}_{E,\mathcal{B}_\mathsf{e}}(k) = 2 \cdot |\Pr[\mathsf{S}_3] - 1/2|$.

Proof of Claim 23. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}_\mathsf{e}$ that attacks the $\mathtt{CCA}$ security of the SKE scheme E with the claimed advantage. The description of $\mathcal{B}_\mathsf{e} = (\mathcal{B}_{\mathsf{e}1}, \mathcal{B}_{\mathsf{e}2})$ is as follows:

$\mathcal{B}_{\mathsf{e}1}(1^k)$**:** $\mathcal{B}_{\mathsf{e}1}$ runs $(pk, sk) \leftarrow \mathsf{PKG}(1^k)$ and $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}_{\mathtt{HYB}}(sk,\cdot)}(pk)$, sets $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathsf{e}1}$'s entire view), and terminates with output $(m_0, m_1, \mathsf{st}_{\mathcal{B}})$.

[14] This case includes the case of $c' = c^*$, because c^* is an encryption of K^*.

$\mathcal{B}_{\mathsf{e2}}^{\mathsf{SDec}(K^*,\cdot)}(\mathsf{st}_{\mathcal{B}}, \widehat{c}^*)$**:** (where K^* denotes the key in $\mathcal{B}_{\mathsf{e}}$'s $\mathtt{CCA}$ experiment) $\mathcal{B}_{\mathsf{e2}}$ picks $\widehat{K} \in \{0,1\}^k$ uniformly at random, and computes $c^* \leftarrow \mathsf{Enc}(pk, \widehat{K})$. Then, $\mathcal{B}_{\mathsf{e2}}$ sets $C^* \leftarrow (c^*, \widehat{c}^*)$, and runs $\mathcal{A}_2(\mathsf{st}, C^*)$.

$\mathcal{B}_{\mathsf{e2}}$ answers $\mathcal{A}$'s decryption queries as the decryption oracle in Game 3 does, which is possible because $\mathcal{B}_{\mathsf{e2}}$ possesses sk.

When $\mathcal{A}_2$ terminates with output $(C' = (c', \widehat{c}'), \mathsf{st}')$, $\mathcal{B}_{\mathsf{e2}}$ computes $K' \leftarrow \mathsf{Dec}(sk, c')$, and proceeds as follows:

- If $K' = \bot$, then $\mathcal{B}_{\mathsf{e2}}$ sets $m' \leftarrow \bot$.
- If $K' = \widehat{K}$ and $\widehat{c}' = \widehat{c}$, then $\mathcal{B}_{\mathsf{e2}}$ sets $m' \leftarrow \mathsf{same}$.
- If $K' = \widehat{K}$ and $\widehat{c}' \neq \widehat{c}$, then $\mathcal{B}_{\mathsf{e2}}$ submits $\widehat{c}'$ to its own decryption oracle $\mathsf{SDec}(K^*, \cdot)$, and receives the result m'.
- Otherwise (i.e. $K' \notin \{\bot, \widehat{K}\}$), $\mathcal{B}_{\mathsf{e2}}$ computes $m' \leftarrow \mathsf{SDec}(K', \widehat{c}')$.

Then, if $m' \in \{m_0, m_1\}$, then $\mathcal{B}_{\mathsf{e2}}$ sets $m' \leftarrow \mathsf{same}$. Finally, $\mathcal{B}_{\mathsf{e2}}$ runs $b' \leftarrow \mathcal{A}_3(\mathsf{st}', m')$, and terminates with output b'.

The above completes the description of $\mathcal{B}_{\mathsf{e}}$. Note that $\mathcal{B}_{\mathsf{e}}$ submits at most one decryption query (and thus it is in fact an adversary against the 1-bounded $\mathtt{CCA}$ security [7] of E), and never submits a prohibited decryption query $\widehat{c}^*$.

Let $b \in \{0,1\}$ be $\mathcal{B}_{\mathsf{e}}$'s challenge bit. It is easy to see that $\mathcal{B}_{\mathsf{e}}$ simulates Game 3 perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is that of $\mathcal{B}_{\mathsf{e}}$'s. In particular, the final output $C' = (c', \widehat{c}')$ of $\mathcal{A}_2$ is processed in exactly the same way as that in Game 3. Therefore, the probability that $b' = b$ occurs in the experiment simulated by $\mathcal{B}_{\mathsf{e}}$ is exactly the same as the probability that $\mathcal{A}$ succeeds in guessing the challenge bit in Game 3, i.e. we have $\Pr[b' = b] = \Pr[\mathsf{S}_3]$. Thus, we have

$$\mathsf{Adv}^{\mathtt{CCA}}_{E,\mathcal{B}_{\mathsf{e}}}(k) = 2 \cdot \left|\Pr[b' = b] - \frac{1}{2}\right| = 2 \cdot \left|\Pr[\mathsf{S}_3] - \frac{1}{2}\right|.$$

□ (**Claim 23**)

Claims 21 to 23 and Equation (16) guarantee that there exist PPTAs $\mathcal{B}_{\mathsf{p}}$ and $\mathcal{B}_{\mathsf{e}}$ satisfying Equation (15). Hence, we have shown that if Π is $\mathtt{wRNM\text{-}DCCA}$ secure and E is $\mathtt{CCA}$ secure, then $\Pi_{\mathtt{HYB}}$ is $\mathtt{wRNM\text{-}DCCA}$ secure.

Proof for Randomness-Inextractability. Let $\mathcal{A}$ be any PPTA adversary that attacks the randomness-inextractability of $\Pi_{\mathtt{HYB}}$. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}$ that attacks the randomness-inextractability of the underlying detectable PKE scheme Π. The description of $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ is as follows:

$\mathcal{B}_1(1^k)$**:** $\mathcal{B}_1$ picks $K^* \in \{0,1\}^k$ uniformly at random, sets $\mathsf{st}_{\mathcal{B}} \leftarrow (\mathcal{B}_1$'s entire view), and terminates with output $(K^*, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_2^{\mathsf{Dec}(sk,\cdot)}(\mathsf{st}_{\mathcal{B}}, pk, c^*)$**:** $\mathcal{B}_2$ runs $(m, \mathsf{st}) \leftarrow \mathcal{A}_1(1^k)$ and $\widehat{c}^* \leftarrow \mathsf{SEnc}(K^*, m)$, sets $C^* \leftarrow (c^*, \widehat{c}^*)$, and then runs $\mathcal{A}_2(\mathsf{st}, pk, C^*)$.

$\mathcal{B}_2$ answers $\mathcal{A}_2$'s decryption queries by using $\mathcal{B}_2$'s own decryption oracle in a straightforward manner.

When $\mathcal{A}_2$ terminates with output $(m', R' = (r', K'))$ where $(r', K') \in \{0,1\}^\ell \times \{0,1\}^k$, $\mathcal{B}_2$ outputs (K', r'), and terminates.

$\mathsf{KKG}_{\mathtt{CKN}}(1^k)$:	$\mathsf{Encap}_{\mathtt{CKN}}(pk; R)$:	$\mathsf{Decap}_{\mathtt{CKN}}(sk, C)$:	$\mathsf{F}_{\mathtt{CKN}}(pk, C^*, C')$:
$(pk, sk) \leftarrow \mathsf{PKG}(1^k)$ Return (pk, sk).	Parse R as $(r, s) \in \{0,1\}^\ell \times \{0,1\}^k$. $c \leftarrow \mathsf{Enc}(pk, s; r)$ $\alpha \leftarrow \mathsf{G}(s)$ Parse α as $(K_{\mathtt{m}}, K)$ $\in \{0,1\}^k \times \{0,1\}^n$. $\tau \leftarrow \mathsf{Mac}(K_{\mathtt{m}}, c)$ $C \leftarrow (c, \tau)$ Return (C, K).	$(c, \tau) \leftarrow C$ $s \leftarrow \mathsf{Dec}(sk, c)$ If $s = \bot$ then return $\bot$. $\alpha \leftarrow \mathsf{G}(s)$ Parse α as $(K_{\mathtt{m}}, K)$ $\in \{0,1\}^k \times \{0,1\}^n$. If $\mathsf{MVer}(K_{\mathtt{m}}, c, \tau) = \top$ then return K else return $\bot$.	$(c^*, \tau^*) \leftarrow C^*$ $(c', \tau') \leftarrow C'$ Return $\mathsf{F}(pk, c^*, c')$.

Fig. 11. The CKN construction $\mathit{\Gamma}_{\mathtt{CKN}}$: A method for converting a `wRNM-DCCA` secure detectable PKE scheme $\mathit{\Pi}$ into a `wNM-DCCA` secure detectable KEM, using a MAC $\mathcal{M}$ and a PRG G as additional building blocks.

The above completes the description of $\mathcal{B}$. It is easy to see that $\mathcal{B}$ perfectly simulates the randomness-inextractability experiment for $\mathcal{A}$. Furthermore, if $\mathcal{A}$ succeeds in violating the randomness-inextractability of $\mathit{\Pi}_{\mathtt{HYB}}$, then it holds that $\mathsf{F}(pk, c^*, \mathsf{Enc}(pk, K'; r')) = 1$, which is exactly the condition of making $\mathcal{B}$'s randomness-inextractability experiment output 1. This means that we have

$$\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Pi},\mathcal{B}}(k) = \mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Pi}_{\mathtt{HYB}},\mathcal{A}}(k).$$

Since $\mathit{\Pi}$ is assumed to be randomness-inextractable, the above equality shows that $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Pi}_{\mathtt{HYB}},\mathcal{A}}(k)$ is negligible. Since the choice of $\mathcal{A}$ was arbitrary, for any PPTA $\mathcal{A}$ we can show a negligible upperbound on $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Pi}_{\mathtt{HYB}},\mathcal{A}}(k)$. This means that $\mathit{\Pi}_{\mathtt{HYB}}$ satisfies randomness-inextractability. □ (**Lemma 4**)

E.3 From `wRNM-DCCA` Secure Detectable PKE to `wNM-DCCA` Secure Detectable KEM

In this section, we explain that the method by Canetti, Krawczyk, and Nielsen [4] for converting a replayable `CCA` secure PKE scheme into a `CCA` secure detectable KEM using a MAC, can be used for converting a `wRNM-DCCA` secure detectable PKE scheme into a `wNM-DCCA` secure KEM as well. Furthermore, the transformation also preserves the unpredictability and randomness-inextractability.

More precisely, we show how to construct a `wNM-DCCA` secure detectable KEM (with unpredictability) from a `wRNM-DCCA` secure detectable PKE scheme (with unpredictability), a strongly one-time secure MAC, and a PRG.

Formally, the construction is as follows: Let $n = n(k)$ be a positive polynomial. Let $\mathit{\Pi} = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{F})$ be a detectable PKE scheme such that the randomness space of Enc is $\{0,1\}^\ell$ for some polynomial $\ell = \ell(k)$. Furthermore, let $\mathcal{M} = (\mathsf{Mac}, \mathsf{MVer})$ be a MAC, and $\mathsf{G} : \{0,1\}^k \to \{0,1\}^{k+n}$ be a PRG.[15] Then, we construct a detectable KEM $\mathit{\Gamma}_{\mathtt{CKN}} = (\mathsf{KKG}_{\mathtt{CKN}}, \mathsf{Encap}_{\mathtt{CKN}}, \mathsf{Decap}_{\mathtt{CKN}}, \mathsf{F}_{\mathtt{CKN}})$, which we call the CKN construction, as described in Fig. 11. In the figure, we make the randomness R used by $\mathsf{Encap}_{\mathtt{CKN}}$ explicit so that it is convenient to consider randomness-inextractability. The session-key space of $\mathit{\Gamma}_{\mathtt{CKN}}$ is $\{0,1\}^n$, and the randomness space of $\mathsf{Encap}_{\mathtt{CKN}}$ is $\{0,1\}^{\ell+k}$.

The security properties of $\mathit{\Gamma}_{\mathtt{CKN}}$ are guaranteed by the following lemmas.

Lemma 10. *Assume that the detectable PKE scheme $\mathit{\Pi}$ is `wRNM-DCCA` secure, the MAC $\mathcal{M}$ is `SOT` secure, and G is a PRG. Then, the detectable KEM $\mathit{\Gamma}_{\mathtt{CKN}}$ in Fig. 11 is `wNM-DCCA` secure.*

Lemma 11. *Assume that the detectable PKE scheme $\mathit{\Pi}$ satisfies unpredictability. Then, the detectable KEM $\mathit{\Gamma}_{\mathtt{CKN}}$ in Fig. 11 also satisfies unpredictability.*

[15] The formal definition of a MAC can be found in Appendix A.2.

Lemma 12. *Assume that the detectable PKE scheme $\mathit{\Pi}$ satisfies randomness-inextractability. Then, the detectable KEM $\mathit{\Gamma}_{\mathtt{CKN}}$ in Fig. 11 also satisfies randomness-inextractability.*

In the following, we first show the proof of Lemma 10, next the proof of Lemma 11, and finally the proof of Lemma 12.

Proof of Lemma 10. We will show that for any PPTA adversary $\mathcal{A}$ that attacks the `wNM-DCCA` security of the detectable KEM $\mathit{\Gamma}_{\mathtt{CKN}}$, there exist PPTAs $\mathcal{B}_{\mathtt{p}}$, $\mathcal{B}_{\mathtt{g}}$, and $\mathcal{B}_{\mathtt{m}}$, such that

$$\mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\mathit{\Gamma}_{\mathtt{CKN}},\mathcal{A}}(k) \le 2 \cdot \Big(2^{-k} + \mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\mathit{\Pi},\mathcal{B}_{\mathtt{p}}}(k) + \mathsf{Adv}^{\mathtt{PRG}}_{\mathsf{G},\mathcal{B}_{\mathtt{g}}}(k) + \mathsf{Adv}^{\mathtt{SOT}}_{\mathcal{M},\mathcal{B}_{\mathtt{m}}}(k)\Big). \tag{17}$$

To this end, fix arbitrarily a PPTA $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that attacks the `wNM-DCCA` security of $\mathit{\Gamma}_{\mathtt{CKN}}$. Consider the following sequence of games: (Here, the values with asterisk (*) represent those related to $\mathcal{A}$'s challenge ciphertext.)

Game 1: This is the `wNM-DCCA` experiment $\mathsf{Expt}^{\mathtt{wNM\text{-}DCCA}}_{\mathit{\Gamma}_{\mathtt{CKN}},\mathcal{A}}(k)$ itself. For defining the subsequent games, we make the experiment pick a random value $s' \in \{0,1\}^k$ at the beginning of the game. (Since s' does not appear in $\mathcal{A}$'s view in Game 1, it does not affect $\mathcal{A}$'s behavior at all.)

Game 2: Same as Game 1, except that the decapsulation result K' (which is input to $\mathcal{A}_2$) of the ciphertext $C' = (c', \tau')$ (which is finally output by $\mathcal{A}_1$) is computed by:

$$K' \leftarrow \begin{cases} \mathsf{Decap}_{\mathtt{CKN}}(sk, C') & \text{if } \mathsf{Dec}(sk, c') \ne s' \\ K_1^* & \text{if } \mathsf{Dec}(sk, c') = s' \text{ and } \mathsf{MVer}(K_{\mathtt{m}}^*, c', \tau') = \top \\ \bot & \text{if } \mathsf{Dec}(sk, c') = s' \text{ and } \mathsf{MVer}(K_{\mathtt{m}}^*, c', \tau') = \bot \end{cases}.$$

Game 3: Same as Game 2, except that the information of s^* is erased from c^*, and instead s' is encrypted. More precisely, in Game 3, c^* is computed by "$c^* \leftarrow \mathsf{Enc}(pk, s')$," instead of "$c^* \leftarrow \mathsf{Enc}(pk, s^*)$."

Game 4: Same as Game 3, except that $K_{\mathtt{m}}^* \in \{0,1\}^k$ and $K_1^* \in \{0,1\}^n$ are picked uniformly at random, independently of s^*.

Game 5: Same as Game 4, except that the decapsulation result K' (which is input to $\mathcal{A}_2$) of the ciphertext $C' = (c', \tau')$ (which is finally output by $\mathcal{A}_1$) is computed as follows:

$$K' \leftarrow \begin{cases} \mathsf{Decap}_{\mathtt{CKN}}(sk, C') & \text{if } \mathsf{Dec}(sk, c') \ne s' \\ \bot & \text{if } \mathsf{Dec}(sk, c') = s' \end{cases}$$

For $i \in [5]$, let S_i be the event that in Game i, $\mathcal{A}$ succeeds in guessing b (i.e. $b' = b$ occurs).

By definitions of the events and applying the triangle inequality, $\mathcal{A}$'s `wNM-DCCA` advantage can be estimated as follows:

$$\mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\mathit{\Gamma}_{\mathtt{CKN}},\mathcal{A}}(k) = 2 \cdot \Big|\Pr[\mathsf{S}_1] - \frac{1}{2}\Big| \le 2 \cdot \sum_{i \in [4]} \Big|\Pr[\mathsf{S}_i] - \Pr[\mathsf{S}_{i+1}]\Big| + 2 \cdot \Big|\Pr[\mathsf{S}_5] - \frac{1}{2}\Big|. \tag{18}$$

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

Claim 24 $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]| \le 2^{-k}$.

Proof of Claim 24. The difference between Game 1 and Game 2 is only in how the ciphertext $C' = (c', \tau')$ (which is finally output by $\mathcal{A}_1$) satisfying $\mathsf{Dec}(sk, c') = s'$ is decapsulated. Therefore, the difference between $\Pr[\mathsf{S}_1]$ and $\Pr[\mathsf{S}_2]$ is upperbounded by the probability that $\mathcal{A}_1$ outputs such a ciphertext in Game 1. However, s' is chosen uniformly at random from $\{0,1\}^k$, and is information-theoretically hidden from $\mathcal{A}_1$ in Game 1. Hence, the probability that $\mathcal{A}_1$ outputs such a ciphertext is upperbounded by 2^{-k}. □ (**Claim 24**)

Claim 25 *There exists a PPTA $\mathcal{B}_{\mathtt{p}}$ such that* $\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi,\mathcal{B}_{\mathtt{p}}}(k) = |\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]|$.

Proof of Claim 25. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}_{\mathtt{p}}$ that attacks the wRNM-DCCA security of the detectable PKE scheme Π with the claimed advantage. The description of $\mathcal{B}_{\mathtt{p}} = (\mathcal{B}_{\mathtt{p}1}, \mathcal{B}_{\mathtt{p}2}, \mathcal{B}_{\mathtt{p}3})$ is as follows:

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

$\mathcal{B}_{\mathtt{p}2}^{\mathsf{Dec}(sk,\cdot)}(\mathsf{st}_{\mathcal{B}}, c^*)$**:** $\mathcal{B}_{\mathtt{p}2}$ computes $\alpha^* \leftarrow \mathsf{G}(s^*)$, parses α^* as $(K^*_{\mathtt{m}}, K^*_1) \in \{0,1\}^k \times \{0,1\}^n$, and then computes $\tau^* \leftarrow \mathsf{Mac}(K^*_{\mathtt{m}}, c^*)$. Next, $\mathcal{B}_{\mathtt{p}2}$ picks $K^*_0 \in \{0,1\}^n$ and $b \in \{0,1\}$ uniformly at random. Then, $\mathcal{B}_{\mathtt{p}2}$ sets $C^* \leftarrow (c^*, \tau^*)$, and runs $\mathcal{A}_1(pk, C^*, K^*_b)$.
$\mathcal{B}_{\mathtt{p}2}$ answers $\mathcal{A}_1$'s decapsulation queries $C = (c, \tau)$ by executing $\mathsf{Decap}_{\mathtt{CKN}}(SK, C)$, where $\mathcal{B}_{\mathtt{p}2}$ uses its own decryption oracle as a substitute for $\mathsf{Dec}(sk, \cdot)$. (Note that here, by the rule of the wNM-DCCA experiment, $\mathcal{A}_1$'s decapsulation queries $C = (c, \tau)$ satisfy $\mathsf{F}_{\mathtt{CKN}}(pk, C^*, C) = 0$, which is equivalent to $\mathsf{F}(pk, c^*, c) = 0$, and hence $\mathcal{B}_{\mathtt{p}2}$ need not submit prohibited queries c satisfying $\mathsf{F}(pk, c^*, c) = 1$ to $\mathcal{B}_{\mathtt{p}2}$'s own decryption oracle.)
When $\mathcal{A}_1$ terminates with output $(C' = (c', \tau'), \mathsf{st})$, $\mathcal{B}_{\mathtt{p}2}$ sets $c'' \leftarrow c'$ if $c' \neq c^*$, or sets $c'' \leftarrow \bot$ otherwise. $\mathcal{B}_{\mathtt{p}2}$ also prepares $\mathsf{st}'_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathtt{p}2}$'s entire view). Finally, $\mathcal{B}_{\mathtt{p}2}$ terminates with output $(c'', \mathsf{st}'_{\mathcal{B}})$.

$\mathcal{B}_{\mathtt{p}3}(\mathsf{st}'_{\mathcal{B}}, m'')$**:** $\mathcal{B}_{\mathtt{p}3}$ checks if $c' = c^*$ or $m'' = \mathsf{same}$. If this is the case, then $\mathcal{B}_{\mathtt{p}3}$ computes

$$K' \leftarrow \begin{cases} K^*_1 & \text{if } \mathsf{MVer}(K^*_{\mathtt{m}}, c', \tau') = \top \\ \bot & \text{otherwise} \end{cases}.$$

Otherwise (i.e. $c' \neq c^*$ and $m'' \neq \mathsf{same}$), $\mathcal{B}_{\mathtt{p}3}$ interprets m'' as $s' = \mathsf{Dec}(sk, c')$, and computes the decapsulation result K' of C' by performing the remaining procedure of $\mathsf{Decap}_{\mathtt{CKN}}(sk, C')$.
Finally, $\mathcal{B}_{\mathtt{p}3}$ runs $b' \leftarrow \mathcal{A}_2(\mathsf{st}, K')$, and terminates with output $\gamma' \leftarrow (b' \overset{?}{=} b)$.

The above completes the description of $\mathcal{B}_{\mathtt{p}}$. Note that $\mathcal{B}_{\mathtt{p}2}$ never submits a prohibited decryption query c such that $\mathsf{F}(pk, c^*, c) = 1$ to its decryption oracle, and the ciphertext c' finally output by $\mathcal{B}_{\mathtt{p}2}$ is guaranteed to be distinct from $\mathcal{B}_{\mathtt{p}}$'s challenge ciphertext c^*.

Let $\gamma \in \{0,1\}$ be $\mathcal{B}_{\mathtt{p}}$'s challenge bit. Note that $\mathcal{B}_{\mathtt{p}3}$ outputs $\gamma' = 1$ only when $b' = b$ occurs. Therefore, $\mathcal{B}_{\mathtt{p}}$'s wRNM-DCCA advantage can be calculated as follows:

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

We argue that if $\gamma = 1$, then $\mathcal{B}_{\mathtt{p}}$ simulates Game 2 perfectly for $\mathcal{A}$. Specifically, if $\gamma = 1$, then c^* is an encryption of s^*, and thus $\mathcal{A}$'s challenge ciphertext $C^* = (c^*, \tau^*)$ is computed in exactly the same way as that in Game 2. Furthermore, $\mathcal{B}_{\mathsf{p}}$'s response to $\mathcal{A}_1$'s decryption queries is perfect due to the use of $\mathcal{B}_{\mathsf{p}}$'s decryption oracle. Finally, the decapsulation result K' of the ciphertext $C' = (c', \tau')$ that is finally output by $\mathcal{A}_1$ is also computed in exactly the same way as it is done in Game 2. To see this, note that if $c' = c^*$ or $m'' = \mathsf{same}$, then it is guaranteed that $\mathsf{Dec}(sk, c') \in \{s^*, s'\}$. Correspondingly, if $\mathsf{Dec}(sk, c') \in \{s^*, s'\}$, then Game 2 computes K' by

$$K' \leftarrow \begin{cases} K^*_1 & \text{if } \mathsf{MVer}(K^*_{\mathtt{m}}, c', \tau') = \top \\ \bot & \text{otherwise} \end{cases}.$$

(In particular, if $C' = (c', \tau')$ satisfies $\mathsf{Dec}(sk, c') = s^*$, then $\mathsf{Decap}_{\mathtt{CKN}}(sk, C')$ computes $\alpha^* = (K^*_{\mathtt{m}} \| K^*_1) \leftarrow \mathsf{G}(s^*)$, and hence outputs K^*_1 or $\bot$ depending on the result of $\mathsf{MVer}(K^*_{\mathtt{m}}, c', \tau')$.) Furthermore, if $c' \neq c^*$ and $m'' \neq \mathsf{same}$, then it holds that $K' = \mathsf{Decap}_{\mathtt{CKN}}(sk, C')$, and thus K' is computed identically by $\mathcal{B}_{\mathsf{p}3}$ as in Game 2. Therefore, $\mathcal{B}_{\mathsf{p}3}$ inputs a correct decapsulation result K' into $\mathcal{A}_2$. Under this situation, the probability that $b' = b$ occurs in the experiment simulated by $\mathcal{B}_{\mathsf{p}}$ is exactly the same as the probability that $\mathcal{A}$ succeeds in guessing b, i.e. we have $\Pr[b' = b | \gamma = 1] = \Pr[\mathsf{S}_2]$.

On the other hand, when $\gamma = 0$, $\mathcal{B}_{\mathsf{p}}$ simulates Game 3 perfectly for $\mathcal{A}$. Specifically, if $\gamma = 0$, then c^* is an encryption of s', which is exactly how c^* is computed in Game 3. The rest is unchanged from the case of $\gamma = 1$. Hence, with a similar argument to the above, we have $\Pr[b' = b | \gamma = 0] = \Pr[\mathsf{S}_3]$.

In summary, we have $\mathsf{Adv}^{\mathtt{wRNM\text{-}DCCA}}_{\Pi, \mathcal{B}_{\mathsf{p}}}(k) = |\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]|$. □ (**Claim 25**)

Claim 26 *There exists a PPTA* $\mathcal{B}_{\mathsf{g}}$ *such that* $\mathsf{Adv}^{\mathtt{PRG}}_{\mathsf{G}, \mathcal{B}_{\mathsf{g}}}(k) = |\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$.

Proof of Claim 26. Using $\mathcal{A}$ as a building block, we show how to construct a PPTA distinguisher $\mathcal{B}_{\mathsf{g}}$ for the PRG G with the claimed advantage. The description of $\mathcal{B}_{\mathsf{g}}$ is as follows:

$\mathcal{B}_{\mathsf{g}}(1^k, \alpha^*)$**:** (where $\alpha^* \in \{0,1\}^{k+n}$) $\mathcal{B}_{\mathsf{g}}$ first parses α^* as $(K^*_{\mathtt{m}}, K^*_1) \in \{0,1\}^k \times \{0,1\}^n$, and picks $s', K^*_0 \in \{0,1\}^k$ and $b \in \{0,1\}$ uniformly at random. $\mathcal{B}_{\mathsf{g}}$ next computes $(pk, sk) \leftarrow \mathsf{PKG}(1^k)$, $c^* \leftarrow \mathsf{Enc}(pk, s')$, and $\tau^* \leftarrow \mathsf{Mac}(K^*_{\mathtt{m}}, c^*)$. Then, $\mathcal{B}_{\mathsf{g}}$ sets $C^* \leftarrow (c^*, \tau^*)$, and runs $(C', \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Decap}_{\mathtt{CKN}}(sk, \cdot)}(pk, C^*, K^*_b)$. $\mathcal{B}_{\mathsf{g}}$ decapsulates C' in exactly the same way as Game 3 does, and obtains K'. Then, $\mathcal{B}_{\mathsf{g}}$ runs $b' \leftarrow \mathcal{A}_2(\mathsf{st}, K')$, and terminates with output $(b' \stackrel{?}{=} b)$.

The above completes the description of $\mathcal{B}_{\mathsf{g}}$. Note that $\mathcal{B}_{\mathsf{g}}$ outputs 1 only when $b' = b$ occurs.

It is easy to see that if α^* is an output of the PRG G (for some uniformly chosen seed), then $\mathcal{B}_{\mathsf{g}}$ simulates Game 3 perfectly for $\mathcal{A}$, where the seed used for computing α^* is regarded as s^* in Game 3. Hence, the probability that $b' = b$ occurs (and hence $\mathcal{B}_{\mathsf{g}}$ outputs 1) is exactly the same as the probability that it occurs in Game 3, i.e. we have $\Pr_{s^* \leftarrow \{0,1\}^k;\ \alpha^* \leftarrow \mathsf{G}(s^*)}[\mathcal{B}_{\mathsf{g}}(1^k, \alpha^*) = 1] = \Pr[\mathsf{S}_3]$.

On the other hand, if α^* is a random $(k+n)$-bit string, then $\mathcal{B}_{\mathsf{g}}$ simulates Game 4 perfectly for $\mathcal{A}$. With a similar argument to the above, we have $\Pr_{\alpha^* \leftarrow \{0,1\}^{k+n}}[\mathcal{B}_{\mathsf{g}}(1^k, \alpha^*)] = \Pr[\mathsf{S}_4]$.

In summary, we have

$$\begin{aligned} \mathsf{Adv}^{\mathtt{PRG}}_{\mathsf{G}, \mathcal{B}_{\mathsf{g}}}(k) &= \left| \Pr_{s^* \leftarrow \{0,1\}^k;\ \alpha^* \leftarrow \mathsf{G}(s^*)}[\mathcal{B}_{\mathsf{g}}(1^k, \alpha^*) = 1] - \Pr_{\alpha^* \leftarrow \{0,1\}^{k+n}}[\mathcal{B}_{\mathsf{g}}(1^k, \alpha^*) = 1] \right| \\ &= \left| \Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4] \right|. \end{aligned}$$

□ (**Claim 26**)

Claim 27 *There exists a PPTA* $\mathcal{B}_{\mathtt{m}}$ *such that* $\mathsf{Adv}^{\mathtt{SOT}}_{\mathcal{M},\mathcal{B}_{\mathtt{m}}}(k) \geq |\Pr[\mathsf{S}_4] - \Pr[\mathsf{S}_5]|$.

Proof of Claim 27. For $i \in \{4,5\}$, let B_i be the event that in Game i, the ciphertext $C' = (c', \tau')$ which is finally output by $\mathcal{A}_1$ satisfies the conditions $\mathsf{Dec}(sk, c') = s'$ and $\mathsf{MVer}(K^*_{\mathtt{m}}, c', \tau') = \top$. Note that Game 4 and Game 5 can differ only when B_4 or B_5 occurs in the corresponding games. (If the ciphertext C' finally output by $\mathcal{A}_1$ never satisfies the above conditions, then Game 4 and Game 5 compute the decapsulation result K' in exactly the same way, and hence these games proceed identically.) Therefore, we have

$$|\Pr[\mathsf{S}_4] - \Pr[\mathsf{S}_5]| \leq \Pr[\mathsf{B}_4] = \Pr[\mathsf{B}_5].$$

We show that using $\mathcal{A}$ as a building block, we can construct a PPTA adversary $\mathcal{B}_{\mathtt{m}}$ that attacks the $\mathtt{SOT}$ security of the MAC $\mathcal{M}$ with advantage $\mathsf{Adv}^{\mathtt{SOT}}_{\mathcal{M},\mathcal{B}_{\mathtt{m}}}(k) \geq \Pr[\mathsf{B}_5]$, which, combined with the above inequality, proves the claim. The description of $\mathcal{B}_{\mathtt{m}} = (\mathcal{B}_{\mathtt{m1}}, \mathcal{B}_{\mathtt{m2}})$ is as follows:

$\mathcal{B}_{\mathtt{m1}}(1^k)$**:** $\mathcal{B}_{\mathtt{m1}}$ picks $s' \in \{0,1\}^k$ uniformly at random, and runs $(pk, sk) \leftarrow \mathsf{PKG}(1^k)$ and $c^* \leftarrow \mathsf{Enc}(pk, s')$. Then $\mathcal{B}_{\mathtt{m1}}$ sets $M \leftarrow c^*$ and $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_{\mathtt{m1}}$'s entire view), and terminates with output $(M, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{\mathtt{m2}}(\mathsf{st}_{\mathcal{B}}, \tau^*)$**:** $\mathcal{B}_{\mathtt{m2}}$ picks $K^* \in \{0,1\}^k$ uniformly at random, sets $C^* \leftarrow (c^*, \tau^*)$, and runs $(C' = (c', \tau'), \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Decap}_{\mathtt{CKN}}(sk,\cdot)}(pk, C^*, K^*)$. Finally, $\mathcal{B}_{\mathtt{m2}}$ sets $M' \leftarrow c'$, and terminates with output (M', τ').

The above completes the description of $\mathcal{B}_{\mathtt{m}}$. Note that it is guaranteed that $(M, \tau^*) \neq (M', \tau')$, because we set $M = c^*$ and $M' = c'$, and $\mathcal{A}_1$'s output $C' = (c', \tau')$ always satisfies $C^* = (c^*, \tau^*) \neq (c', \tau') = C'$ according to the rule of the $\mathtt{wNM\text{-}DCCA}$ experiment.

Let $K \in \{0,1\}^k$ be the key chosen in $\mathcal{B}_{\mathtt{m}}$'s $\mathtt{SOT}$ experiment. Then, it is straightforward to see that $\mathcal{B}_{\mathtt{m}}$ simulates Game 5 perfectly for $\mathcal{A}_1$ by regarding K as $K^*_{\mathtt{m}}$ in Game 5. Therefore, the probability that $\mathsf{MVer}(K, M', \tau') = \top$ and $(M', \tau') \neq (M, \tau^*)$ occur (and hence $\mathcal{B}_{\mathtt{m}}$'s $\mathtt{SOT}$ experiment outputs 1) is at least $\Pr[\mathsf{B}_5]$. That is, we have $\mathsf{Adv}^{\mathtt{SOT}}_{\mathcal{M},\mathcal{B}_{\mathtt{m}}}(k) \geq \Pr[\mathsf{B}_5]$, as required. □ (**Claim 27**)

Claim 28 $\Pr[\mathsf{S}_5] = 1/2$.

Proof of Claim 28. In Game 5, both K^*_1 and K^*_0 are chosen uniformly at random, and the results of the decapsulation oracle and the decapsulation result K' of C' do not reveal any information of the challenge bit b. Therefore, $\mathcal{A}$'s view is identical regardless of b, which means that b is information-theoretically hidden from $\mathcal{A}$'s view. Therefore, the probability that $\mathcal{A}$ succeeds in guessing b is exactly 1/2. □ (**Claim 28**)

Claims 24 to 28 and Equation (18) guarantee that there exist PPTAs $\mathcal{B}_{\mathtt{p}}$, $\mathcal{B}_{\mathtt{g}}$, and $\mathcal{B}_{\mathtt{m}}$, satisfying Equation (17). This, combined with our assumptions on the building blocks, implies that $\mathsf{Adv}^{\mathtt{wNM\text{-}DCCA}}_{\Gamma_{\mathtt{CKN}},\mathcal{A}}(k)$ is negligible. Since the choice of the PPTA adversary $\mathcal{A}$ was arbitrarily, for any PPTA $\mathtt{wNM\text{-}DCCA}$ adversary $\mathcal{A}$ we can show that its advantage is negligible. Hence, $\Gamma_{\mathtt{CKN}}$ is $\mathtt{wNM\text{-}DCCA}$ secure. □ (**Lemma 10**)

Proof of Lemma 11. Let $\mathcal{A}$ be any PPTA adversary that attacks the unpredictability of the detectable KEM $\Gamma_{\mathtt{CKN}}$. By definition, $\mathcal{A}$'s unpredictability advantage is calculated as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{CKN}},\mathcal{A}}(k) &= \Pr[(pk, sk) \leftarrow \mathsf{KKG}_{\mathtt{CKN}}(1^k);\ C' = (c', \tau') \leftarrow \mathcal{A}^{\mathsf{Decap}_{\mathtt{CKN}}(sk,\cdot)}(pk); \\ &\qquad\qquad (C^* = (c^*, \tau^*), K^*) \leftarrow \mathsf{Encap}_{\mathtt{CKN}}(pk) : \mathsf{F}_{\mathtt{CKN}}(pk, C^*, C') = 1] \\ &= \Pr[(pk, sk) \leftarrow \mathsf{PKG}(1^k);\ C' = (c', \tau') \leftarrow \mathcal{A}^{\mathsf{Decap}_{\mathtt{CKN}}(sk,\cdot)}(pk); \\ &\qquad\qquad s^* \leftarrow \{0,1\}^k; c^* \leftarrow \mathsf{Enc}(pk, s^*) : \mathsf{F}(pk, c^*, c') = 1].\end{aligned}$$

With this in mind, using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}$ that attacks the unpredictability of the detectable PKE scheme Π with advantage $\mathsf{Adv}^{\mathtt{UNP}}_{\Pi,\mathcal{B}}(k) = \mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{CKN}},\mathcal{A}}(k)$, which implies the lemma. The description of $\mathcal{B}$ is as follows:

$\mathcal{B}^{\mathsf{Dec}(sk,\cdot)}(pk)$**:** $\mathcal{B}$ runs $\mathcal{A}(pk)$. $\mathcal{B}$ answers the decapsulation queries from $\mathcal{A}$ in a straightforward manner by using $\mathcal{B}$'s own decryption oracle $\mathsf{Dec}(sk, \cdot)$. When $\mathcal{A}$ terminates with output $C' = (c', \tau')$, $\mathcal{B}$ picks $s^* \in \{0,1\}^k$ uniformly at random, and terminates with output (s^*, c') (where s^* is regarded as a plaintext of Π).

The above completes the description of $\mathcal{B}$. Note that $\mathcal{B}$ perfectly simulates the unpredictability experiment (regarding the detectable KEM $\Gamma_{\mathtt{CKN}}$) for $\mathcal{A}$. In particular, $\mathcal{B}$ can perfectly respond to $\mathcal{A}$'s decapsulation queries by using $\mathcal{B}$'s own decryption oracle.

$\mathcal{B}$'s unpredictability advantage is calculated as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{UNP}}_{\Pi,\mathcal{B}}(k) &= \Pr[(pk, sk) \leftarrow \mathsf{PKG}(1^k);\ (s^*, c') \leftarrow \mathcal{B}^{\mathsf{Dec}(sk,\cdot)}(pk); \\ &\qquad\qquad c^* \leftarrow \mathsf{Enc}(pk, s^*) : \mathsf{F}(pk, c^*, c') = 1] \\ &= \Pr[(pk, sk) \leftarrow \mathsf{PKG}(1^k);\ C' = (c', \tau') \leftarrow \mathcal{A}^{\mathsf{Decap}_{\mathtt{CKN}}(sk,\cdot)}(pk); \\ &\qquad\qquad s^* \leftarrow \{0,1\}^k; c^* \leftarrow \mathsf{Enc}(pk, s^*) : \mathsf{F}(pk, c^*, c') = 1],\end{aligned}$$

and the rightmost is exactly $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{CKN}},\mathcal{A}}(k)$ as we showed above. That is, we have $\mathsf{Adv}^{\mathtt{UNP}}_{\Pi,\mathcal{B}}(k) = \mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{CKN}},\mathcal{A}}(k)$, which, by our assumption that Π satisfies unpredictability, implies that $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{CKN}},\mathcal{A}}(k)$ is negligible. Since the choice of $\mathcal{A}$ was arbitrarily, for any PPTA adversary $\mathcal{A}$ we can show a negligible upperbound on $\mathsf{Adv}^{\mathtt{UNP}}_{\Gamma_{\mathtt{CKN}},\mathcal{A}}(k)$. Hence, $\Gamma_{\mathtt{CKN}}$ satisfies unpredictability. □ (**Lemma 11**)

Proof of Lemma 12. Let $\mathcal{A}$ be any PPTA adversary that attacks the randomness-inextractability of the detectable KEM $\Gamma_{\mathtt{CKN}}$.

By definition, $\mathcal{A}$'s randomness-inextractability advantage is calculated as follows:

$$\begin{aligned}&\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Gamma_{\mathtt{CKN}},\mathcal{A}}(k) \\ &= \Pr[(pk, sk) \leftarrow \mathsf{KKG}_{\mathtt{CKN}}(1^k);\ (C^*, K^*) \leftarrow \mathsf{Encap}_{\mathtt{CKN}}(pk);\ R' \leftarrow \mathcal{A}^{\mathsf{Decap}_{\mathtt{CKN}}(sk,\cdot)}(pk, C^*, K^*); \\ &\qquad\qquad (C', K') \leftarrow \mathsf{Encap}_{\mathtt{CKN}}(pk; R') : \mathsf{F}_{\mathtt{CKN}}(pk, C^*, C') = 1] \\ &= \Pr[(pk, sk) \leftarrow \mathsf{PKG}(1^k);\ s^* \leftarrow \{0,1\}^k;\ c^* \leftarrow \mathsf{Enc}(pk, s^*);\ (K^*_{\mathtt{m}}, K^*) \leftarrow \mathsf{G}(s^*);\ \tau^* \leftarrow \mathsf{Mac}(K^*_{\mathtt{m}}, c^*); \\ &\qquad C^* \leftarrow (c^*, \tau^*);\ R' = (r', s') \leftarrow \mathcal{A}^{\mathsf{Decap}_{\mathtt{CKN}}(sk,\cdot)}(pk, C^*, K^*);\ c' \leftarrow \mathsf{Enc}(pk, s'; r') : \mathsf{F}(pk, c^*, c') = 1].\end{aligned}$$

With this in mind, using $\mathcal{A}$ as a building block, we show how to construct a PPTA adversary $\mathcal{B}$ that attacks the randomness-inextractability of the detectable PKE scheme Π with advantage $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Pi,\mathcal{B}}(k) = \mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\Gamma_{\mathtt{CKN}},\mathcal{A}}(k)$, which implies the lemma. The description of $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ is as follows:

$\mathcal{B}_1(1^k)$**:** $\mathcal{B}_1$ picks $s^* \in \{0,1\}^k$ uniformly, sets $\mathsf{st}_{\mathcal{B}} \leftarrow$ ($\mathcal{B}_1$'s entire view), and terminates with output $(s^*, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_2^{\mathsf{Dec}(sk,\cdot)}(\mathsf{st}_{\mathcal{B}}, pk, c^*)$**:** $\mathcal{B}_2$ computes $(K^*_{\mathtt{m}}, K^*) \leftarrow \mathsf{G}(s^*)$ (where $K^*_{\mathtt{m}} \in \{0,1\}^k$ and $K^* \in \{0,1\}^n$) and $\tau^* \leftarrow \mathsf{Mac}(K^*_{\mathtt{m}}, c^*)$, sets $C^* \leftarrow (c^*, \tau^*)$, and then runs $\mathcal{A}(pk, C^*, K^*)$. $\mathcal{B}_2$ answers the decapsulation queries from $\mathcal{A}$ in a straightforward manner by using $\mathcal{B}_2$'s own decryption oracle $\mathsf{Dec}(sk, \cdot)$. When $\mathcal{A}$ terminates with output $R' = (r', s')$, $\mathcal{B}_2$ terminates with output (s', r') where s' is viewed as a plaintext for $\mathit{\Pi}$.

The above completes the description of $\mathcal{B}$. Note that $\mathcal{B}$ simulates the randomness-inextractability experiment (regarding the detectable KEM $\mathit{\Gamma}_{\mathtt{CKN}}$) for $\mathcal{A}$ perfectly. In particular, $\mathcal{B}$ can perfectly respond to $\mathcal{A}$'s decapsulation queries by using $\mathcal{B}$'s own decryption oracle.

$\mathcal{B}$'s randomness-inextractability advantage is calculated as follows:

$$\begin{aligned}
&\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Pi},\mathcal{B}}(k)\\
&= \Pr[(s^*, \mathsf{st}_{\mathcal{B}}) \leftarrow \mathcal{B}_1(1^k);\ (pk, sk) \leftarrow \mathsf{PKG}(1^k);\ c^* \leftarrow \mathsf{Enc}(pk, s^*);\ (s', r') \leftarrow \mathcal{B}_2^{\mathsf{Dec}(sk,\cdot)}(\mathsf{st}_{\mathcal{B}}, pk, c^*);\\
&\qquad\qquad c' \leftarrow \mathsf{Enc}(pk, s'; r') : \mathsf{F}(pk, c^*, c') = 1]\\
&= \Pr[(pk, sk) \leftarrow \mathsf{PKG}(1^k);\ s^* \leftarrow \{0,1\}^k;\ c^* \leftarrow \mathsf{Enc}(pk, s^*);\ (K^*_{\mathtt{m}}, K^*) \leftarrow \mathsf{G}(s^*);\ \tau^* \leftarrow \mathsf{Mac}(K^*_{\mathtt{m}}, c^*);\\
&\qquad C^* \leftarrow (c^*, \tau^*);\ R' = (r', s') \leftarrow \mathcal{A}^{\mathsf{Decap}_{\mathtt{CKN}}(sk,\cdot)}(pk, C^*, K^*);\ c' \leftarrow \mathsf{Enc}(pk, s'; r') : \mathsf{F}(pk, c^*, c') = 1],
\end{aligned}$$

and the rightmost is exactly $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Gamma}_{\mathtt{CKN}},\mathcal{A}}(k)$ as we showed above. That is, we have $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Pi},\mathcal{B}}(k) = \mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Gamma}_{\mathtt{CKN}},\mathcal{A}}(k)$, which, by our assumption that $\mathit{\Pi}$ satisfies randomness-inextractability, implies that $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Gamma}_{\mathtt{CKN}},\mathcal{A}}(k)$ is negligible. Since the choice of $\mathcal{A}$ was arbitrarily, for any PPTA adversary $\mathcal{A}$ we can show a negligible upperbound on $\mathsf{Adv}^{\mathtt{R\text{-}Inext}}_{\mathit{\Gamma}_{\mathtt{CKN}},\mathcal{A}}(k)$. Hence, $\mathit{\Gamma}_{\mathtt{CKN}}$ satisfies randomness-inextractability. □ (**Lemma 12**)