Chosen Ciphertext Security via UCE

Takahiro Matsuda and Goichiro Hanaoka

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

Abstract. Bellare, Hoang, and Keelveedhi (CRYPTO'13) introduced a security notion for a family of (hash) functions called *universal computational extractor* (UCE), and showed how it can be used to realize various kinds of cryptographic primitives in the standard model whose (efficient) constructions were only known in the random oracle model. Although the results of Bellare et al. have shown that UCEs are quite powerful and useful, the notion of UCE is new, and its potential power and limitation do not seem to have been clarified well. To further widen and deepen our understanding of UCE, in this paper we study the construction of chosen ciphertext secure (CCA secure) public key encryption (PKE), one of the most important primitives in the area of cryptography to which (in)applicability of UCEs was not covered by the work of Bellare et al.

We concretely consider the setting in which other than a UCE, we only use chosen plaintext secure (CPA secure) PKE as an additional building block, and obtain several negative and positive results. As our negative results, we show difficulties of instantiating the random oracle in the Fujisaki-Okamoto (FO) construction (PKC'99) with a UCE, by exhibiting pairs of CPA secure PKE and a UCE for which the FO construction instantiated with these pairs becomes insecure (assuming that CPA secure PKE and a UCE exist at all). Then, as our main positive result, we show how to construct a CCA secure PKE scheme using only CPA secure PKE and a UCE as building blocks. Furthermore, we also show how to extend this result to a CCA secure deterministic PKE scheme for block sources (with some constraint on the running time of the sources). Our positive results employ the ideas and techniques from the Dolev-Dwork-Naor (DDN) construction (STOC'91), and for convenience we abstract and formalize the "core" structure of the DDN construction as a stand-alone primitive that we call *puncturable tag-based encryption*, which might be of independent interest.

1 Introduction

Background and Motivation. For the constructions of cryptographic primitives in which we use a hash function as a building block, if we can view the hash function as a random oracle [8], then in most cases we can obtain simple and practical constructions. Moreover, there are some cryptographic primitives whose (efficient) constructions are known only if we use a random oracle. However, random oracles do not exist in the real world, and there are several problems for security proofs in the random oracle model (e.g. [13,23,31]). Therefore, it is in general desirable to consider the constructions of cryptographic primitives without using random oracles.

H. Krawczyk (Ed.): PKC 2014, LNCS 8383, pp. 56–76, 2014.

In CRYPTO 2013, Bellare, Hoang, and Keelveedhi [4] introduced a new security notion for a family of (hash) functions called *universal computational extractor* (UCE), whose main purpose is to "instantiate" and "replace" random oracles used in a wide class of the constructions of cryptographic primitives with UCEs. The UCE security is intended to capture the security satisfied by a hash function that "behaves like a random oracle" as close as possible, and roughly guarantees that outputs of a hash function (in the family) look random, as long as the inputs to the hash function are hard-to-find even given the related information (called *leakage*) of the inputs, and as long as the inputs are independent of a function index that specifies the function from the family.[1] Bellare et al. [4] showed how UCEs can be used to realize various kinds of cryptographic primitives in the standard model whose (efficient) constructions were only known in the random oracle model (such as deterministic public key encryption [3] and message-locked encryption [7]).

Although the results of Bellare et al. have shown that a UCE is quite powerful and useful, the notion of UCE is new, and its potential power and limitation do not seem to have been clarified well. To further widen and deepen our understanding of UCE, in this paper we study the construction of chosen ciphertext secure (CCA secure) public key encryption (PKE) [33,36,19], one of the most important primitives in the area of cryptography for which we have witnessed the great success in the literature (e.g. [9,20,21,1,34]) and yet to which (in)applicability of UCE was not covered by the work of Bellare et al. (In fact, Bellare et al. showed the instantiability of the random oracle in the OAEP scheme [9], but they only showed the chosen plaintext (CPA) security.) As a first step towards clarifying the usefulness of UCEs in the context of constructing CCA secure PKE, in this paper we concretely consider the setting where, other than a UCE, we only use CPA secure PKE as an additional (and seemingly minimal) building block, and obtain several negative and positive results.

Our Contributions. In this paper, we investigate the usefulness and (in)applicability of UCEs in the context of constructing CCA secure PKE. As mentioned above, we concretely study the setting in which other than a UCE, we only use CPA secure PKE as an additional building block, and obtain several negative and positive results.

Our starting point is the Fujisaki-Okamoto (FO) construction [20] which constructs CCA secure PKE from a random oracle and a CPA secure PKE scheme (satisfying some property on cardinality of ciphertexts). As our negative results, in Section 3, we show the difficulties of instantiating the random oracle in the FO construction with a UCE if we simply put a function index of a UCE into a public key. Specifically, we first show that (assuming that CPA secure PKE and a UCE exist) there exists a pair of CPA secure PKE and a UCE for which the FO construction instantiated with this pair is *not* even CPA secure. This result is shown by designing a pair of a CPA secure PKE

[1] Actually, "UCE" is not a single security notion, but a family of security notions for a function family, from which a particular notion is specified when we specify what class of "sources" we will consider. For more details, see the explanation and the formal definition in Section 2.1. For convenience, in the introduction, when we just write "UCE" (resp. "UCE security"), we mean a function family that satisfies some version of UCE security notions (resp. one of UCE security notions), and exactly which notion is used will be specified in the formal statements given in Sections 3 and 5.

scheme having a "weak randomness" and a UCE having a function-index-dependent "weak input" so that when this pair is used as building blocks in the FO construction, the resulting PKE scheme has a public-key-dependent "weak plaintext," which is weak in the sense that a ciphertext leaks the information of whether or not this weak plaintext is encrypted. We then further investigate whether the FO construction can be secure for "public-key-independent" messages, which could be still useful for example in the setting where the FO construction is used as a key encapsulation mechanism (KEM) by encrypting a random message and using it as a session-key (for SKE). We show another negative result for this case by exhibiting yet another pair of CPA secure PKE and a UCE such that when used as building blocks, the FO construction is *not* CCA1 secure even if we restrict an adversary to choose two uniformly random (and hence public-key-independent) plaintexts as its challenge plaintexts and allow the adversary to make only one decryption query. This result is obtained by designing a pair of CPA secure PKE and a UCE which have a public-key-dependent "critical ciphertext" whose decryption result reveals the (essential part of) secret key. For more details, see Section 3.

Given the above negative results, we depart from the original FO construction [20]. By employing the ideas and techniques from the classical Dolev-Dwork-Naor (DDN) construction [19] together with a UCE, we obtain several positive results. Specifically, in Section 5, as our main positive result we show how to construct a CCA secure PKE scheme using only a CPA secure PKE scheme and a UCE. We actually construct a CCA secure key encapsulation mechanism (KEM), but by combining it with a CCA secure SKE scheme, we obtain a full-fledged CCA secure PKE scheme [17]. Furthermore, we show how this KEM can be extended to obtain a deterministic PKE (DPKE) scheme that is CCA secure for block sources (with some additional constraint on the running time of the sources), using the same building blocks as above. To the best of our knowledge, our DPKE scheme is the first scheme which achieves CCA security for block sources in the standard model without using lossy trapdoor functions (TDFs) [35] or related primitives (though we have some non-standard restriction on the running time of sources). By noting that a CCA secure DPKE scheme (for block sources with bounded running time) is as it is an injective TDF which satisfies adaptively one-wayness [26], this result immediately yields an adaptively one-way TDF as well. We also show how to weaken the assumption on the UCE security if the underlying PKE scheme is additionally a lossy encryption scheme [6]. The ideas and techniques for our proposed constructions are explained in more details in "*Overview of Techniques*" paragraph below.

Our positive results clarify not only a new and important primitive for which UCEs are useful, but also insights for the "gap" between CPA and CCA security for PKE. Specifically, our results imply that if there exists a CPA secure PKE scheme and a UCE, then there exist a CCA secure PKE scheme and a CCA secure DPKE for block sources (with some constraint on the running time). This could be contrasted with the current state-of-the-art attempts for constructing PKE schemes that satisfy security which is as close as CCA security, using only a CPA secure PKE scheme as a building block. The current best security is bounded CCA security [16] (more precisely, non-malleability under bounded-CCA [15] and its slightly stronger variant [30]). Therefore, our results serve as a concrete evidence that a UCE is quite a strong primitive, and has the power to "jump" the currently known gap between CPA and CCA security for PKE schemes.

As explained in details below, in our proposed constructions, we employ the ideas and techniques from the DDN construction [19]. For ease of notation and reducing the description complexity, we abstract the "core" structure of the DDN construction as tag-based encryption (TBE) [28,25] with some special property, and formalize it as a stand-alone primitive which we call *puncturable TBE* (PTBE). This formalization may be useful for understanding the security proof of the DDN construction, and future works that use the ideas and the techniques of the DDN construction in a similar way to ours, and may be of independent interest. For more details, see Section 4.

Due to space limitation, most of the proofs of the theorems and lemmas in this paper are omitted and will be given in the full version, and we only give proof sketches or intuitive explanations.

Overview of Techniques. Our proposed CCA secure KEM is based on the DDN construction [19], which originally constructs a CCA secure PKE scheme using a CPA secure PKE scheme, a non-interactive zero-knowledge (NIZK) proof, and a one-time signature scheme. In the original DDN construction, the NIZK proof roughly ensures that each "component"-ciphertext from the underlying CPA secure PKE scheme is in a valid form, i.e. it is in the range of the encryption algorithm and encrypts the same value. Here, if there is another mechanism that ensures the "validity" of component-ciphertexts, then we can remove the NIZK proof from this construction. This is the place where a UCE comes into play. Specifically, by relying on the power of UCE, for the DDN construction we realize the mechanism of the "randomness-recovering decryption" (also called "witness-recovering decryption") [20,21,35,10,37,32,24,29], where (a part of) randomness used to generate a ciphertext is recovered in the decryption process, and this recovered randomness is used to check the validity of the component-ciphertexts by re-encryption. This "decrypt-then-re-encrypt"-style validity check works as an alternative of the NIZK proof in the original DDN construction. Actually, such a mechanism of recovering randomness in the decryption process usually causes a circularity between a plaintext and a randomness (used to generate the ciphertext itself), but in our construction this circularity can be overcome by the security of a UCE.

Then, our proposed CCA secure KEM is obtained by applying one more enhancement to this "DDN without NIZK" construction. Specifically, we implement the mechanism of preventing the "re-use" of component-ciphertexts in the DDN construction, which is originally realized by a one-time signature (i.e. the technique of using a verification key of the one-time signature as a kind of "non-reusable tag" in each ciphertext), with a commitment scheme. This change not only leads to smaller ciphertexts, but also (by appropriately combining it with a UCE) to a scheme with "full randomness-recovering," namely, in the decryption process an entire randomness is recovered. Hence, with a similar observation in [10], we also obtain a CCA secure DPKE scheme for block sources. (However, we need to put some additional constraint on the sources, due to the requirement on UCE security notions that we use.) For more details about our constructions, see Section 5.

Related Work. The notion of CCA security for PKE was formalized by Naor and Yung [33] and Rackoff and Simon [36]. Since the introduction of the notion, CCA secure PKE schemes have been studied in a number of papers, and thus we only briefly review constructions from general cryptographic assumptions. Dolev, Dwork, and Naor [19]

showed the first construction of a CCA secure PKE scheme, from a CPA secure scheme and a NIZK proof system, based on the construction by Naor and Yung [33] that achieves weaker non-adaptive CCA (CCA1) security. Canetti, Halevi, and Katz [14] showed how to transform an identity-based encryption scheme into a CCA secure PKE scheme. Kiltz [25] showed that the transform of [14] is applicable to a weaker primitive of tag-based encryption (TBE). Peikert and Waters [35] showed how to construct a CCA secure PKE scheme from a *lossy* trapdoor function (TDF). Subsequent works showed that TDFs with weaker security/functionality properties are sufficient for obtaining CCA secure PKE schemes [37,26,39]. Myers and Shelat [32] showed that a CCA secure PKE scheme for 1-bit plaintexts can be turned into one for arbitrarily long plaintexts. Hohenberger, Lewko, and Waters [24] showed that CCA secure PKE can be constructed from a PKE scheme with a weaker security notion called detectable CCA security. Lin and Tessaro [27] showed how to amplify weak CCA security into strong (ordinary) CCA secure one. Recently, Sahai and Waters [38] showed how (among other primitives) CCA secure PKE can be constructed using indistinguishability obfuscation [2,22]. Very recently, Matsuda and Hanaoka [29] showed how to construct CCA secure PKE using obfuscation for point functions (with multi-bit output), and Dachman-Soled [18] showed a construction from PKE satisfying (the standard model) plaintext-awareness as well as some additional "simulatability" property. We note that our proposed constructions and these two constructions [29,18] have the properties that they all rely on the ideas and techniques of the DDN construction [19].

2 Preliminaries

In this section, we review the basic notation and the definitions of primitives.

Basic Notation. $\mathbb{N}$ denotes the set of all natural numbers. For $m, n \in \mathbb{N}$, we define $[n] := \{1, \ldots, n\}$, and "$\mathsf{Func}_{m \to n}$" denotes the set of all functions F of the form $F : \{0,1\}^m \to \{0,1\}^n$. "$x \leftarrow y$" denotes that x is chosen uniformly at random from y if y is a finite set, x is output from y if y is a function or an algorithm, or y is assigned to x otherwise. If x and y are strings, then "$|x|$" denotes the bit-length of x, "$x \| y$" denotes the concatenation x and y, and "$(x \stackrel{?}{=} y)$" is 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. $\mathcal{A}^{\mathcal{O}}$ denotes an algorithm $\mathcal{A}$ with oracle access to $\mathcal{O}$. A function $\epsilon(k) : \mathbb{N} \to [0,1]$ 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 the paper, we use the character "k" for the security parameter. For an algorithm M, we denote by $\mathsf{t}_{\mathsf{M}} = \mathsf{t}_{\mathsf{M}}(k)$ the maximum (worst-case) running time of M when M is run with security parameter k.

2.1 Universal Computational Extractor (UCE)

Here, we recall the definition of UCE (universal computational extractor) [4], which is a family of security notions for a (hash) function family. We first recall the syntax of a function family, and then the definitions of UCE security. We also introduce a property that we call *smoothness* which is used in our negative results in Section 3.

Syntax. Let $m, n : \mathbb{N} \to \mathbb{N}$ be functions of k. A family of functions (function family) $\mathcal{F}$ with input length m and output length n consists of the following two deterministic PTAs $(\mathsf{FKG}, \mathsf{F})$: FKG is the key generation algorithm which takes 1^k as input, and outputs a function index κ.; F is the evaluation algorithm that takes a function index κ and a string $x \in \{0,1\}^m$ as input, and outputs a string $y \in \{0,1\}^n$. For notational convenience, we write $\mathsf{F}_\kappa(\cdot)$ to mean $\mathsf{F}(\kappa, \cdot)$.

UCE Security. Before giving the formal definitions, we give some overview. As mentioned earlier, the UCE security is a family of security notions, from which a particular notion is specified when we specify a class $\mathtt{S}$ of "sources" $\mathcal{S}$. A source is a part of an adversary's algorithm that is responsible for computing the inputs to the function $\mathsf{F}_\kappa(\cdot)$ (that are chosen independently of the function index κ) together with some relevant information called *leakage* L, where the independence of the inputs from κ is captured by allowing $\mathcal{S}$ only oracle access to the function. The UCE security for the class $\mathtt{S}$ ($\mathtt{UCE}[\mathtt{S}]$ security, for short), states that for any PPTA adversary, called *distinguisher*, who receives the function index κ and the leakage L, cannot tell whether L is computed by a source $\mathcal{S} \in \mathtt{S}$ using the function $\mathsf{F}_\kappa(\cdot)$ or using a random function, better than a random guess. How strong/weak, and how useful $\mathtt{UCE}[\mathtt{S}]$ security is depends on what restrictions we put on the class $\mathtt{S}$ of sources. The wider the class $\mathtt{S}$ is, the stronger $\mathtt{UCE}[\mathtt{S}]$ security becomes. In other words, for classes $\mathtt{S}$ and $\mathtt{S}'$ of sources, if $\mathtt{S} \subseteq \mathtt{S}'$, then $\mathtt{UCE}[\mathtt{S}']$ security implies $\mathtt{UCE}[\mathtt{S}]$ security.

In the proceedings version [4], Bellare et al. considered a class of computationally unpredictable sources (which we denote by $\mathtt{S}^{\mathtt{cup}}$), which roughly requires that given a leakage L computed by a source $\mathcal{S}$ in the class under the situation $\mathcal{S}$ has oracle access to a random function, it is hard to find any query to the oracle made by $\mathcal{S}$. Bellare et al. used $\mathtt{UCE}[\mathtt{S}^{\mathtt{cup}}]$ secure function families to achieve a number of positive results. Unfortunately, however, Brzuska, Farshim, and Mittelbach [12] later showed that if indistinguishability obfuscation [2,22] is possible, then $\mathtt{UCE}[\mathtt{S}^{\mathtt{cup}}]$ security is unachievable (see also [5]). Since Garg et al. [22] recently showed a candidate construction of it, as mentioned in [5], currently it seems more likely that indistinguishability obfuscation is possible than $\mathtt{UCE}[\mathtt{S}^{\mathtt{cup}}]$ secure function families exist. To avoid the attack by Brzuska et al. [12], Bellare et al. [5] suggested several approaches for weakening the $\mathtt{UCE}[\mathtt{S}^{\mathtt{cup}}]$ security by putting several restrictions on the sources so that the indistinguishability obfuscation-based attack is not possible (and they re-achieved their results of [4] by using appropriately weakened versions of UCE security notions). In this paper, we adopt the two approaches suggested in [5] for weakening $\mathtt{UCE}[\mathtt{S}^{\mathtt{cup}}]$ security: to consider statistical unpredictability, and to put the restrictions on the running time and the number of queries of sources.

Now we proceed to the formal definitions. Let $\mathcal{F} = (\mathsf{FKG}, \mathsf{F})$ be a function family with input length $m = m(k)$ and output length $n = n(k)$. A *source* $\mathcal{S}$ (for $\mathcal{F}$) is an oracle PPTA that takes 1^k as input, expects to have access to an oracle $\mathcal{O} \in \mathsf{Func}_{m \to n}$, and outputs some value $L \in \{0,1\}^*$ (called *leakage*). For a pair of a source $\mathcal{S}$ and an adversary $\mathcal{A}$ (called "distinguisher"), consider the $\mathtt{UCE}$ experiment $\mathsf{Expt}^{\mathtt{UCE}}_{\mathcal{F},(\mathcal{S},\mathcal{A})}(k)$ that is defined as in Fig. 1 (leftmost).

$\mathsf{Expt}^{\mathtt{UCE}}_{\mathcal{F},(\mathcal{S},\mathcal{A})}(k)$:
$\kappa \leftarrow \mathsf{FKG}(1^k)$
$\mathcal{O}_1(\cdot) \leftarrow \mathsf{F}_\kappa(\cdot)$
$\mathcal{O}_0(\cdot) \leftarrow \mathsf{Func}_{m\to n}$
$b \leftarrow \{0,1\}$
$L \leftarrow \mathcal{S}^{\mathcal{O}_b}(1^k)$
$b' \leftarrow \mathcal{A}(1^k, \kappa, L)$
Return $(b' \stackrel{?}{=} b)$.

$\mathsf{Expt}^{\mathtt{UNP}}_{\mathcal{S},\mathcal{P}}(k)$:
$\mathcal{O}(\cdot) \leftarrow \mathsf{Func}_{m\to n}$
$L \leftarrow \mathcal{S}^{\mathcal{O}}(1^k)$
Let Q be $\mathcal{S}$'s queries submitted to $\mathcal{O}$.
$x' \leftarrow \mathcal{P}(1^k, L)$
Return 1 iff $x' \in Q$.

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

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

Fig. 1. The experiments for defining security. The UCE experiment for a function family $\mathcal{F}$ (leftmost), the UNP experiment for a source $\mathcal{S}$ (second-left), the CPA security experiment for a PKE scheme Π (second-right), and the CCA security experiment for a KEM Γ (rightmost).

Definition 1. *We say that a function family $\mathcal{F}$ is* UCE[S]*-secure if for all sources $\mathcal{S} \in \mathtt{S}$ and for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{UCE}}_{\mathcal{F},(\mathcal{S},\mathcal{A})}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{UCE}}_{\mathcal{F},(\mathcal{S},\mathcal{A})}(k) = 1] - 1/2|$ is negligible.*

We next define the classes of the sources that we treat in this paper. For a source $\mathcal{S}$ and a PPTA $\mathcal{P}$ (called "predictor"), consider the *unpredictability* experiment $\mathsf{Expt}^{\mathtt{UNP}}_{\mathcal{S},\mathcal{P}}(k)$ defined as in Fig. 1 (second-left).[2]

Definition 2. *For polynomials $t, q > 0$, we say that a source $\mathcal{S}$ is* (t,q)-computationally (resp. statistically) unpredictable*, denoted by $\mathcal{S} \in \mathtt{S}^{\mathtt{cup}}_{t,q}$ (resp. $\mathcal{S} \in \mathtt{S}^{\mathtt{sup}}_{t,q}$), if (1) $\mathcal{S}$'s running time is at most t and $\mathcal{S}$ makes at most q queries, and (2) for all PPTAs (resp. all computationally unbounded algorithms) $\mathcal{P}$, $\mathsf{Adv}^{\mathtt{UNP}}_{\mathcal{S},\mathcal{P}}(k) := \Pr[\mathsf{Expt}^{\mathtt{UNP}}_{\mathcal{S},\mathcal{P}}(k) = 1]$ is negligible. Furthermore, we just say that a source $\mathcal{S}$ is* computationally (resp. statistically) unpredictable*, denoted by $\mathcal{S} \in \mathtt{S}^{\mathtt{cup}}$ (resp. $\mathcal{S} \in \mathtt{S}^{\mathtt{sup}}$), if $\mathcal{S}$ is (t,q)-computationally (resp. statistically) unpredictable for some positive polynomials t, q.*

We remark that our definition of (t,q)-computationally/statistically unpredictable source is simpler than the "parallel sources" introduced in [5], which also considers some restrictions on the running time, the number of queries (and the output length), and additionally on how the source is run "parallelly." We choose not to use the definition of the parallel sources in [5] as it is, because in this paper we do not need to consider the "parallel run" of the sources, in which case we believe our definitions are more straightforward and simpler. We note that any (t,q)-computationally/statistically unpredictable sources that we defined above can always be cast as computationally/statistically unpredictable parallel sources of [5] with appropriate parameters.[3]

We also remark that we could also consider the restriction on the output length of the sources (i.e. the length of leakage). In this paper we choose not to do so for simplicity. However, we note that in each of our results for which we use a UCE security notion as an assumption, the output length of the sources used in the security proofs will be clear.

[2] Bellare et al. [4] introduced two kinds of definitions for unpredictability, (ordinary) "*unpredictability*" and "*simple unpredictability*," and showed their equivalence. The unpredictability in our paper is the simple unpredictability in [4], which is simpler and easier to work with.

[3] More precisely, our definition of the class $\mathtt{S}^{\mathtt{cup}}_{t,q}$ (resp. $\mathtt{S}^{\mathtt{sup}}_{t,q}$) is strictly contained by the class $\mathcal{S}^{\mathrm{cup}} \cap \mathcal{S}^{\mathrm{prl}}_{t,0,q}$ (resp. $\mathcal{S}^{\mathrm{sup}} \cap \mathcal{S}^{\mathrm{prl}}_{t,0,q}$) in [5].

Smoothness. To show our negative results in Section 3, it is useful to introduce the following property of a function family.

Definition 3. *Let $\mathcal{F} = (\mathsf{FKG}, \mathsf{F})$ be a function family with input length $m = m(k)$ and output length $n = n(k)$. We define the* smoothness *of $\mathcal{F}$, denoted by $\mathsf{Smth}_{\mathcal{F}}(k)$, as* $\mathsf{Smth}_{\mathcal{F}}(k) := \mathbf{E}_{\kappa \leftarrow \mathsf{FKG}(1^k)} \left[\max_{y \in \{0,1\}^n} \Pr_{x \leftarrow \{0,1\}^m}[\mathsf{F}_\kappa(x) = y] \right]$.

The following lemma states a simple fact that a function family satisfying a very weak form of UCE security has negligible smoothness.

Lemma 1. *Let $\mathcal{F}$ be a function family with input length $m = m(k)$ and output length $n = n(k)$ satisfying $m, n \in \omega(\log k)$. If $\mathcal{F}$ is $\mathtt{UCE}[\mathsf{S}^{\mathtt{sup}}_{O(m+n+k),1}]$ secure, then $\mathsf{Smth}_{\mathcal{F}}(k)$ is negligible.*

2.2 Basic Primitives

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

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

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

For $\mathtt{ATK} \in \{\mathtt{CPA}, \mathtt{CCA1}\}$, we say that a PKE scheme Π is ATK secure if for all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{ATK}}_{\Pi,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{ATK}}_{\Pi,\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the experiment $\mathsf{Expt}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k)$ is defined as in Fig. 1 (second-right), and the experiment $\mathsf{Expt}^{\mathtt{CCA1}}_{\Pi,\mathcal{A}}(k)$ is defined as in $\mathsf{Expt}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k)$, except that $\mathcal{A}_1$ has access to the decryption oracle $\mathsf{Dec}(sk, \cdot)$. In both of the experiments, it is required that $|m_0| = |m_1|$.

Here, we recall one of the requirements for the building block PKE scheme for the original FO construction [20]. We say that a PKE scheme $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ has the *large ciphertext cardinality* property if for all pk output by $\mathsf{PKG}(1^k)$, it holds that $\min_m |\{\mathsf{Enc}(pk, m; r) | r \in \{0,1\}^*\}| \in k^{\omega(1)}$. (Not all PKE schemes have this property, but any CPA secure PKE scheme can be turned into one satisfying it [20].)

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

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

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

We say that a KEM Γ is CCA secure if for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{CCA}}_{\Gamma,\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the experiment $\mathsf{Expt}^{\mathtt{CCA}}_{\Gamma,\mathcal{A}}(k)$ is defined as in Fig. 1 (rightmost). In the experiment, the oracle $\mathcal{O}$ is the decapsulation oracle $\mathsf{Decap}(sk, \cdot)$, and $\mathcal{A}$ is not allowed to query c^*.

Commitment Scheme. (We only define a non-interactive commitment scheme that has a setup procedure, which is sufficient for our purpose.) A commitment scheme $\mathcal{C}$ consists of the following two PPTAs $(\mathsf{CKG}, \mathsf{Com})$: CKG takes 1^k as input, and outputs a commitment key ck.; Com takes ck and a message m, and outputs a commitment c.

For security of a commitment scheme, we require the standard *hiding* and *binding* properties. We in fact need weaker properties for both: hiding for messages chosen independently of a commitment key, and binding in which one of the messages needs to be chosen before a commitment key is given, which we call *target-binding*. (The difference between (ordinary) binding and target-binding is similar to the difference between collision resistance and target collision resistance of a hash function.) Due to space limitation, we omit the formal definitions. See the full version for them.

We also require the size of a commitment to be k when generated using a commitment key ck output by $\mathsf{CKG}(1^k)$. This is not a strong requirement if we only consider computational security notions. In particular, a commitment scheme satisfying the above functionality/security requirements can be constructed from any $\mathtt{CPA}$ secure PKE.

3 Uninstantiability of the Fujisaki-Okamoto Construction

In this section, we show our negative results: uninstantiability of the random oracle in the Fujisaki-Okamoto (FO) construction [20] with a UCE secure function family.

This section is organized as follows: In Section 3.1, we review the FO construction [20] in which the random oracle is replaced with a function family. In Section 3.2, we show a pair of a $\mathtt{CPA}$ secure PKE scheme (with large ciphertext cardinality) and a $\mathtt{UCE}[\mathtt{S}]$ secure function family (for some class $\mathtt{S}$ of sources) which, when used as building blocks, makes the FO construction $\mathtt{CPA}$ *insecure*. This attack is demonstrated by using a public-key-dependent plaintext. Then in Section 3.3, we show a pair of a $\mathtt{CPA}$ secure PKE scheme (with large ciphertext cardinality) a $\mathtt{UCE}[\mathtt{S}']$ secure function family (for another class $\mathtt{S}'$ of sources) which, when used as building blocks, makes the FO construction $\mathtt{CCA1}$ *insecure*. This attack is possible even if an adversary has to use public-key-independent plaintexts as its challenge plaintexts, and is allowed to make only one decryption query.

Important Remarks. We would like to emphasize that our results are *not* showing that the FO construction is in general insecure in the standard model. Rather, we show that there are particular pairs of a $\mathtt{CPA}$ secure PKE scheme and a function family satisfying some UCE security notions that make the FO construction insecure. Furthermore, our result is only about the FO construction [20] in which we instantiate the random oracle by putting a function index of the used function family into a public key. It would be interesting and worth clarifying the (im)possibility of instantiating the random oracle in [20] in a way different from ours, and the random oracles in the "hybrid-encryption"-style FO construction [21], with UCE secure function families.

$\mathsf{PKG}_{\mathtt{FO}}(1^k)$:	$\mathsf{Enc}_{\mathtt{FO}}(PK_{\mathtt{FO}}, m; r)$:	$\mathsf{Dec}_{\mathtt{FO}}(SK_{\mathtt{FO}}, C_{\mathtt{FO}})$:
$(pk, sk) \leftarrow \mathsf{PKG}(1^k)$	$(pk, \kappa) \leftarrow PK_{\mathtt{FO}}$	$(sk, pk, \kappa) \leftarrow SK_{\mathtt{FO}}$
$\kappa \leftarrow \mathsf{FKG}(1^k)$	$\alpha \leftarrow (r \| m)$	$\alpha \leftarrow \mathsf{Dec}(sk, C_{\mathtt{FO}})$
$PK_{\mathtt{FO}} \leftarrow (pk, \kappa)$	$R \leftarrow \mathsf{F}_\kappa(\alpha)$	If $\alpha = \perp$ then return $\perp$.
$SK_{\mathtt{FO}} \leftarrow (sk, pk, \kappa)$	$C_{\mathtt{FO}} \leftarrow \mathsf{Enc}(pk, \alpha; R)$	$R \leftarrow \mathsf{F}_\kappa(\alpha)$
Return $(PK_{\mathtt{FO}}, SK_{\mathtt{FO}})$.	Return $C_{\mathtt{FO}}$.	Parse α as $(r, m) \in \{0,1\}^{k+k}$.
		If $\mathsf{Enc}(pk, \alpha; R) = C_{\mathtt{FO}}$
		then return m else return $\perp$.

Fig. 2. The FO construction $\Pi_{\mathtt{FO}}[\Pi, \mathcal{F}]$ based on a PKE scheme Π and a function family $\mathcal{F}$

3.1 The Fujisaki-Okamoto Construction Using a Function Family

Firstly, for ease of notation, we introduce the following conditions for a pair of a PKE scheme and a function family that can be used as building blocks of the FO construction.

Definition 4. *Let $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme and $\mathcal{F}$ be a function family. We say that the pair $(\Pi, \mathcal{F})$ is* FO-compatible *if (1) the plaintext space of Π is $\{0,1\}^{2k}$, (2) the randomness space of Enc is $\{0,1\}^k$, (3) Π has the large ciphertext cardinality property[4], and (4) the input length and output length of $\mathcal{F}$ are $2k$ and k, respectively.*

Now, using a FO-compatible pair $(\Pi, \mathcal{F})$ as building blocks, we define the PKE scheme $\Pi_{\mathtt{FO}}[\Pi, \mathcal{F}] = (\mathsf{PKG}_{\mathtt{FO}}, \mathsf{Enc}_{\mathtt{FO}}, \mathsf{Dec}_{\mathtt{FO}})$ (with plaintext space $\{0,1\}^k$), which we call the *FO construction*, as in Fig. 2.

As mentioned earlier, this PKE scheme can be seen as the original FO construction [20] in which the random oracle is instantiated with the function family $\mathcal{F}$ by putting a function index for $\mathcal{F}$ into a public key. There would be several other ways for instantiating the random oracle with a function family. However, since the original FO construction [20] uses just one random oracle, we believe that the construction in Fig. 2 is the most natural and straightforward instantiation of the random oracle for the original FO construction [20].

3.2 Counterexample for Public-Key-Dependent Plaintexts

This subsection is devoted to proving the following result.

Theorem 1. *Assume that there exists a FO-compatible pair of a* CPA *secure PKE scheme and a* UCE[S] *secure function family with $\mathtt{S}^{\mathtt{sup}}_{O(k),1} \subseteq \mathtt{S} \subseteq \mathtt{S}^{\mathtt{cup}}$. Then, there exists a FO-compatible pair of a* CPA *secure PKE scheme $\widetilde{\Pi}$ and a* UCE[S] *secure function family $\widetilde{\mathcal{F}}$ such that the FO construction $\Pi_{\mathtt{FO}}[\widetilde{\Pi}, \widetilde{\mathcal{F}}]$ is* not CPA *secure.*

Proof of Theorem 1. Let $(\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}), \mathcal{F} = (\mathsf{FKG}, \mathsf{F}))$ be a FO-compatible pair of a CPA secure PKE scheme Π and a UCE[S] secure function family guaranteed to exist by the assumption of the theorem. Then, we construct another PKE scheme $\widetilde{\Pi} = (\widetilde{\mathsf{PKG}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Dec}})$ based on Π, and another function family $\widetilde{\mathcal{F}} = (\widetilde{\mathsf{FKG}}, \widetilde{\mathsf{F}})$ based on $\mathcal{F}$, as in Fig. 3 (left-top and left-bottom, respectively). It is straightforward to see

[4] This is the property required for the building PKE scheme in the original FO construction [20]. We recall the definition of this property in Section 2.2.

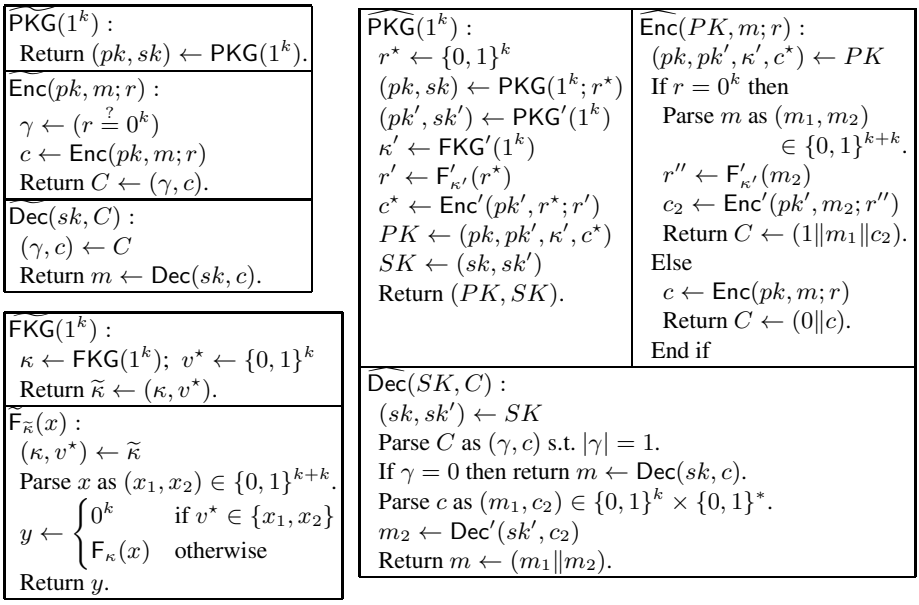

Fig. 3. The building blocks for the FO construction used for showing the uninstantiability: The PKE scheme $\widetilde{\Pi}$ (left-top), the PKE scheme $\widehat{\Pi}$ (right), and the function family $\widetilde{\mathcal{F}}$ (left-bottom)

that the pair $(\widetilde{\Pi}, \widetilde{\mathcal{F}})$ is FO-compatible if so is the pair $(\Pi, \mathcal{F})$. In particular, $\widetilde{\Pi}$ satisfies correctness, and preserves the large ciphertext cardinality property of Π.

Note that $\widetilde{\Pi}$ is designed to have a "weak randomness" $r = 0^k$, and $\widetilde{\mathcal{F}}$ is designed to have a "weak input" $v^\star$ which appears in the function index. We can exploit these "weaknesses" from each building block for attacking the $\mathtt{CPA}$ security of $\Pi_{\mathtt{FO}}[\widetilde{\Pi}, \widetilde{\mathcal{F}}]$.

The following lemmas, together with Lemma 1, imply Theorem 1.

Lemma 2. *If the PKE scheme Π is $\mathtt{CPA}$ secure, then so is the PKE scheme $\widetilde{\Pi}$ constructed as in Fig. 3 (left-top).*

Lemma 3. *For any $\mathtt{S}$ such that $\mathtt{S} \subseteq \mathtt{S}^{\mathtt{cup}}$, if the function family $\mathcal{F}$ is $\mathtt{UCE}[\mathtt{S}]$ secure, then so is the function family $\widetilde{\mathcal{F}}$ constructed as in Fig. 3 (left-bottom).*

Lemma 4. *If $\mathsf{Smth}_{\widetilde{\mathcal{F}}}(k)$ is negligible, then the FO construction $\Pi_{\mathtt{FO}}[\widetilde{\Pi}, \widetilde{\mathcal{F}}]$ is* not $\mathtt{CPA}$ *secure.*

Lemma 2 is trivial to see, because in the $\mathtt{CPA}$ experiment, the probability that the "weak randomness" $r = 0^k$ is chosen is exponentially small. A high level intuition for the proof of Lemma 3 is that the "weak input" $v^\star$ is only in a function index $\widetilde{\kappa}$, chosen uniformly at random and hidden information-theoretically from a source in the unpredictability experiment, and thus it does not do any harm to the $\mathtt{UCE}[\mathtt{S}]$ security of the underlying function family $\mathcal{F}$.

Finally, we provide a sketch for the proof of Lemma 4. Recall that a public key $PK_{\mathtt{FO}}$ of the FO construction $\Pi_{\mathtt{FO}}[\widetilde{\Pi}, \widetilde{\mathcal{F}}]$ is of the form $PK_{\mathtt{FO}} = (pk, \widetilde{\kappa} = (\kappa, v^\star))$,

where $v^\star$ is the "weak input" of $\widetilde{\mathcal{F}}$. Now, let us observe what happens when we encrypt the "weak input" $v^\star$ by $\mathsf{Enc}_{\mathtt{FO}}(PK_{\mathtt{FO}}, \cdot)$. By the design of $\widetilde{\Pi}$, $\widetilde{\mathcal{F}}$, and $\Pi_{\mathtt{FO}}[\widetilde{\Pi}, \widetilde{\mathcal{F}}]$, for any randomness $r \in \{0,1\}^k$ used in $\mathsf{Enc}_{\mathtt{FO}}(PK_{\mathtt{FO}}, \cdot)$, we have

$$\mathsf{Enc}_{\mathtt{FO}}(PK_{\mathtt{FO}}, v^\star; r) = \widetilde{\mathsf{Enc}}(pk, (r\|v^\star); \mathsf{F}_{\widetilde{\kappa}}(r\|v^\star)) = \widetilde{\mathsf{Enc}}(pk, (r\|v^\star); 0^k) = (1\|c'),$$

where $c' = \mathsf{Enc}(pk, (r\|v^\star); 0^k)$, and hence the first bit of $\mathsf{Enc}_{\mathtt{FO}}(PK_{\mathtt{FO}}, v^\star)$ is always 1. On the other hand, if we encrypt a random plaintext m, then by the smoothness of $\widetilde{\mathcal{F}}$ (which is guaranteed to be negligible by the $\mathtt{UCE}[\mathtt{S}]$ security of $\widetilde{\mathcal{F}}$, which is in turn based on the $\mathtt{UCE}[\mathtt{S}^{\mathtt{sup}}_{O(k),1}]$ security of $\mathcal{F}$ and Lemmas 1 and 3), the probability that the first bit of $\mathsf{Enc}_{\mathtt{FO}}(PK_{\mathtt{FO}}, m)$ becomes 1 is negligible. This difference can be used to break the $\mathtt{CPA}$ security of $\Pi_{\mathtt{FO}}[\widetilde{\Pi}, \widetilde{\mathcal{F}}]$. □

3.3 Counterexample for Public-Key-Independent Plaintexts

Here, we consider whether the FO construction can provide security for public-key-independent plaintexts (such as uniform random values). If this is possible, then the FO construction may be still used as a secure KEM by encrypting a random message and using it as a session-key. Unfortunately, however, we show that this is not the case. Specifically, this subsection is devoted to proving the following theorem.

Theorem 2. *Assume that there exists a FO-compatible pair of a* $\mathtt{CPA}$ *secure PKE scheme* $(\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ *and a* $\mathtt{UCE}[\mathtt{S}]$ *secure function family with* $\mathtt{S}^{\mathtt{cup}}_{O(\mathsf{t}_{\mathsf{PKG}}+\mathsf{t}_{\mathsf{Enc}}),1} \subseteq \mathtt{S} \subseteq \mathtt{S}^{\mathtt{cup}}$. *Then, there exists a FO-compatible pair of a* $\mathtt{CPA}$ *secure PKE scheme* $\widehat{\Pi}$ *and a* $\mathtt{UCE}[\mathtt{S}]$ *secure function family* $\widehat{\mathcal{F}}$ *such that the FO construction* $\Pi_{\mathtt{FO}}[\widehat{\Pi}, \widehat{\mathcal{F}}]$ *is* not $\mathtt{CCA1}$ *secure. Furthermore, the* $\mathtt{CCA1}$ *attack for* $\Pi_{\mathtt{FO}}[\widehat{\Pi}, \widehat{\mathcal{F}}]$ *succeeds even if an adversary uses two uniformly random plaintexts as its challenge plaintexts and makes only one decryption query.*

Proof of Theorem 2. Let $(\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}), \mathcal{F} = (\mathsf{FKG}, \mathsf{F}))$ be a FO-compatible pair as before. Without loss of generality, we assume that the randomness space of PKG in Π is $\{0,1\}^k$. (This can be freely adjusted by using an appropriate pseudorandom generator.) To simplify the notation, let us write $\Pi' = (\mathsf{PKG}', \mathsf{Enc}', \mathsf{Dec}')$ to mean Π in which the plaintext space is restricted to $\{0,1\}^k$ (say, by defining $\mathsf{Enc}'(pk, m; r) := \mathsf{Enc}(pk, (m\|0^k); r)$). Similarly, let us write $\mathcal{F}' = (\mathsf{FKG}', \mathsf{F}')$ to mean $\mathcal{F}$ in which the input length is restricted to k-bit (say, as above, by defining[5] $\mathsf{F}'_\kappa(x) := \mathsf{F}_\kappa(x\|0^k)$).

Using Π, Π', and $\mathcal{F}'$ as building blocks, we construct the PKE scheme $\widehat{\Pi} = (\widehat{\mathsf{PKG}}, \widehat{\mathsf{Enc}}, \widehat{\mathsf{Dec}})$ as in Fig. 3 (right). Furthermore, we will again use $\widetilde{\mathcal{F}}$ (constructed based on $\mathcal{F}$ as in Fig. 3 (left-bottom)) as the function family $\widehat{\mathcal{F}}$ for the proof of this theorem. It is not hard to see that the pair $(\widehat{\Pi}, \widetilde{\mathcal{F}})$ is FO-compatible if so is the pair $(\Pi, \mathcal{F})$. In particular, $\widehat{\Pi}$ satisfies correctness, and preserves the large ciphertext cardinality property of Π.

[5] Padding inputs by some default value does not destroy the $\mathtt{UCE}[\mathtt{S}]$ security for $\mathtt{S}$ considered here. Namely, if $\mathcal{F}$ is $\mathtt{UCE}[\mathtt{S}]$ secure, then so is $\mathcal{F}'$.

The following lemmas, together with Lemmas 1 and 3, imply Theorem 2.

Lemma 5. *Assume that the PKE schemes Π and Π' are* CPA *secure, and the function family $\mathcal{F}'$ is* $\mathtt{UCE}[\mathsf{S}^{\mathsf{cup}}_{O(\mathsf{t}_{\mathsf{PKG}}+\mathsf{t}_{\mathsf{Enc}}),1}]$ *secure. Then the PKE scheme $\widehat{\Pi}$ constructed as in Fig. 3 (right) is* CPA *secure.*

Lemma 6. *If $\mathsf{Smth}_{\widetilde{\mathcal{F}}}(k)$ is negligible, then the FO construction $\Pi_{\mathtt{FO}}[\widehat{\Pi}, \widetilde{\mathcal{F}}]$ is* not CCA1 *secure. Furthermore, the* CCA1 *attack succeeds even if an adversary uses two uniformly random plaintexts as its challenge plaintexts and makes only one decryption query.*

We give intuitive explanations for the proofs of the above lemmas. Regarding Lemma 5, note that in the PKE scheme $\widehat{\Pi}$, an encryption $c^\star$ of the randomness $r^\star$ used to generate the "main" public key pk is publicized as part of a public key of $\widehat{\Pi}$. Furthermore, the randomness r' for generating $c^\star$ is computed also from $r^\star$ by using the function family $\mathcal{F}'$. However, the correlation among $r^\star$, pk, and $c^\star$ is dealt with by the $\mathtt{UCE}[\mathsf{S}^{\mathsf{cup}}_{O(t_{\mathsf{PKG}}+t_{\mathsf{Enc}}),1}]$ security of the function family $\mathcal{F}'$ and the CPA security of Π', and then the CPA security of $\widehat{\Pi}$ follows from the CPA security of Π.

Regarding Lemma 6, recall that a public key $PK_{\mathtt{FO}}$ of the FO construction $\Pi_{\mathtt{FO}}[\widehat{\Pi}, \widetilde{\mathcal{F}}]$ is of the form $PK_{\mathtt{FO}} = (PK = (pk, pk', \kappa', c^\star), \widetilde{\kappa} = (\kappa, v^\star))$. Here, observe that if we decrypt the following "critical ciphertext" $C^\star_{\mathtt{FO}} = (1\|v^\star\|c^\star)$ which can be constructed once $PK_{\mathtt{FO}}$ is given, then the decryption result is $r^\star$ (which is the last k-bit of $\widehat{\mathsf{Dec}}(SK, C^\star_{\mathtt{FO}})$ and is the randomness used to generate sk). This follows from the properties of $\widehat{\Pi}$, $\widetilde{\mathcal{F}}$, and $\Pi_{\mathtt{FO}}[\widehat{\Pi}, \widetilde{\mathcal{F}}]$ such that

(1) $\widehat{\mathsf{Dec}}(SK, C^\star_{\mathtt{FO}}) = \widehat{\mathsf{Dec}}(SK, (1\|v^\star\|c^\star)) = (v^\star\|\mathsf{Dec}'(sk', c^\star)) = (v^\star\|r^\star)$,
(2) $\widetilde{\mathsf{F}}_{\widetilde{\kappa}}(v^\star\|r^\star) = 0^k$, and
(3) $\widehat{\mathsf{Enc}}(PK, (v^\star\|r^\star); \widetilde{\mathsf{F}}_{\widetilde{\kappa}}(v^\star\|r^\star)) = \widehat{\mathsf{Enc}}(PK, (v^\star\|r^\star); 0^k)$
$= (1\|v^\star\|\mathsf{Enc}'(pk', r^\star; \mathsf{F}'_{\kappa'}(r^\star))) = (1\|v^\star\|c^\star) = C^\star_{\mathtt{FO}}$.

Then, from $r^\star$ we can recover sk, which is the "main" secret key. This means that a CCA1 adversary can submit the "critical ciphertext" $C^\star_{\mathtt{FO}}$ as its decryption query, and obtain sk. Since with overwhelming probability the challenge ciphertext $C^*_{\mathtt{FO}}$ is of the form $C^*_{\mathtt{FO}} = (0\|\mathsf{Enc}(pk, (r^*\|m_b); \mathsf{F}_\kappa(r^*\|m_b)))$ due to the negligible smoothness of $\widetilde{\mathcal{F}}$, knowing sk allows the adversary to decrypt and tell the challenge bit, no matter what plaintexts are used (and thus even if they are public-key-independent). □

4 Puncturable Tag-Based Encryption

In our proposed constructions in Section 5, we will use the "core" structure that appears in the DDN construction [19]. To ease the notation and reduce the description complexity of our proposed constructions, here we introduce and formalize an abstraction of the structure in the DDN construction as a special type of TBE [28,25], which we call *puncturable tag-based encryption* (PTBE).[6] We remark that there would be several possible

[6] The name "puncturable" is borrowed from the name of the primitive "puncturable" pseudorandom function [38].

ways to formalize the "core" structure of the DDN construction, and our formalization here is one which is convenient for our purpose.

Intuitively, a PTBE scheme is a TBE scheme that has two modes for decryption: The normal mode and the punctured mode. The normal mode is just the normal decryption process of a TBE scheme. In the punctured mode, we can generate a "punctured" secret key $\widehat{sk}_{\mathsf{tag}^*}$ which can be used to decrypt all ciphertexts that are generated under tags tag that are different from tag^*, while the information of plaintexts does not leak from ciphertexts that are generated under the "punctured point" tag tag^*, even given the punctured secret key $\widehat{sk}_{\mathsf{tag}^*}$. (This is as if we can "puncture" the tag space, and hence the name of the primitive.)

More formally, a PTBE scheme consists of the five PPTAs $(\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Punc}, \widehat{\mathsf{TDec}})$ among which the latter three algorithms are deterministic, with the following interface:

$$\underline{\textbf{Key Generation:}} \quad (pk, sk) \leftarrow \mathsf{TKG}(1^k) \qquad \underline{\textbf{Encryption:}} \quad c \leftarrow \mathsf{TEnc}(pk, \mathsf{tag}, m) \qquad \underline{\textbf{Decryption:}} \quad m \ (\text{or } \perp) \leftarrow \mathsf{TDec}(sk, \mathsf{tag}, c)$$

$$\underline{\textbf{Puncturing:}} \quad \widehat{sk}_{\mathsf{tag}^*} \leftarrow \mathsf{Punc}(sk, \mathsf{tag}^*) \qquad \underline{\textbf{Punctured Decryption:}} \quad m \ (\text{or } \perp) \leftarrow \widehat{\mathsf{TDec}}(\widehat{sk}_{\mathsf{tag}^*}, \mathsf{tag}, c)$$

where (pk, sk) is a public/secret key pair, c is a ciphertext of a plaintext m under pk and a tag $\mathsf{tag} \in \{0,1\}^k$, and $\widehat{sk}_{\mathsf{tag}^*}$ is a "punctured" secret key corresponding to a tag $\mathsf{tag}^* \in \{0,1\}^k$.

Correctness. We require for all $k \in \mathbb{N}$, all tags $\mathsf{tag}^*, \mathsf{tag} \in \{0,1\}^k$ such that $\mathsf{tag}^* \neq \mathsf{tag}$, all (pk, sk) output by $\mathsf{TKG}(1^k)$, all m, and all c output by $\mathsf{TEnc}(pk, \mathsf{tag}, m)$, it holds that $\mathsf{TDec}(sk, \mathsf{tag}, c) = \widehat{\mathsf{TDec}}(\mathsf{Punc}(sk, \mathsf{tag}^*), \mathsf{tag}, c) = m$.

We stress that the above correctness is only guaranteed for the case in which a ciphertext c is generated from $\mathsf{TEnc}(pk, \mathsf{tag}, \cdot)$ and $\mathsf{tag} \neq \mathsf{tag}^*$. We do not specify anything when these conditions are not guaranteed.

Extended CPA Security: CPA *Security in the Presence of a Punctured Secret Key.* As a security requirement for a PTBE scheme, we define *extended CPA security* (eCPA security, for short) which requires that CPA security hold even in the presence of a punctured secret key.

Definition 5. *We say that a PTBE scheme $\mathcal{T}$ is* eCPA *secure if for all PPTAs $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{eCPA}}_{\mathcal{T},\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{eCPA}}_{\mathcal{T},\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the experiment $\mathsf{Expt}^{\mathtt{eCPA}}_{\mathcal{T},\mathcal{A}}(k)$ is defined as follows:*

$$\mathsf{Expt}^{\mathtt{eCPA}}_{\mathcal{T},\mathcal{A}}(k) : [\ (\mathsf{tag}^*, \mathsf{st}) \leftarrow \mathcal{A}_0(1^k);\ (pk, sk) \leftarrow \mathsf{TKG}(1^k);$$
$$\widehat{sk}_{\mathsf{tag}^*} \leftarrow \mathsf{Punc}(sk, \mathsf{tag}^*);\ (m_0, m_1, \mathsf{st}') \leftarrow \mathcal{A}_1(\mathsf{st}, pk, \widehat{sk}_{\mathsf{tag}^*});\ b \leftarrow \{0,1\};$$
$$c^* \leftarrow \mathsf{TEnc}(pk, \mathsf{tag}^*, m_b);\ b' \leftarrow \mathcal{A}_2(\mathsf{st}', c^*);\ \text{Return } (b' \stackrel{?}{=} b).\ ],$$

where in the experiment it is required that $|m_0| = |m_1|$.

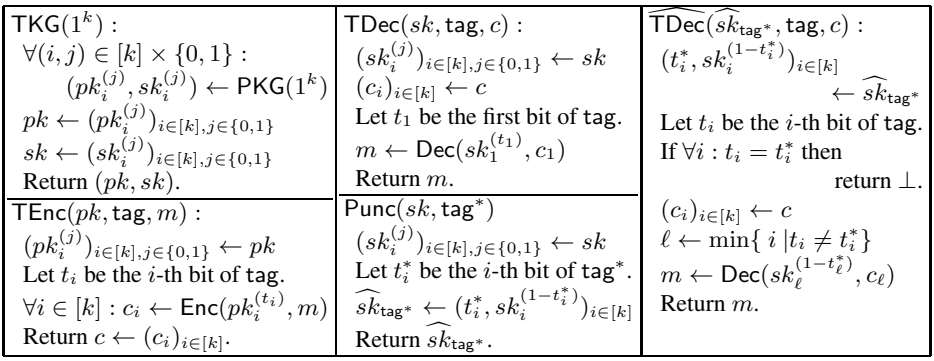

Fig. 4. A concrete instantiation of a PTBE scheme $\mathcal{T}$ based on a CPA secure PKE Π

Concrete Instantiation of PTBE. Since PTBE is intended to abstract the structure that appears in the DDN construction [19], the concrete instantiation of PTBE is exactly one that is used in [19], which is constructed from any CPA secure PKE scheme. Specifically, given a CPA secure PKE scheme $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$, we construct a PTBE scheme $\mathcal{T} = (\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Punc}, \widehat{\mathsf{TDec}})$ as in Fig. 4. In the full version of our paper, we will give the proof for the eCPA security of $\mathcal{T}$.

One of the merits of considering PTBE as a stand-alone primitive would be that it can be instantiated from other primitives, such as broadcast encryption and a multi-user PKE scheme/KEM. A potential advantage of instantiations with these alternative building blocks is that the public key and/or ciphertext size could be much shorter than the simplest construction from a CPA secure PKE scheme. For example, if we use a broadcast encryption scheme by Boneh, Gentry, and Waters [11] to instantiate a PTBE scheme, then a ciphertext consists of a constant number of group elements (in bilinear groups), regardless of the security parameter k.

5 Chosen Ciphertext Security via UCE

In this section, we show our positive results: Specifically, in Section 5.1, we show the proposed CCA secure KEM based on a PTBE scheme, a commitment scheme, and a UCE secure function family (for which we will specify the class of sources shortly). Since the first two building blocks can be constructed from CPA secure PKE, our KEM can be constructed only from CPA secure PKE and a UCE secure function family.

Due to space limitations, our result on a DPKE scheme is not included in this proceedings version, and we refer the reader to the full version. In Section 5.2, we instead give brief overview of the result, as well as several extensions of our positive results.

5.1 CCA Secure KEM

Let $\mathcal{T} = (\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Punc}, \widehat{\mathsf{TDec}})$ be a PTBE scheme and $\mathcal{C} = (\mathsf{CKG}, \mathsf{Com})$ be a commitment scheme. We assume the plaintext/message space of both $\mathcal{T}$ and $\mathcal{C}$ to be $\{0,1\}^k$, and the randomness space of TEnc in $\mathcal{T}$ and Com in $\mathcal{C}$ to be $\{0,1\}^\ell$ and

$\mathsf{KKG}(1^k)$:	$\mathsf{Encap}(PK)$:	$\mathsf{Decap}(SK, C)$:
$(pk, sk) \leftarrow \mathsf{TKG}(1^k)$	$(pk, ck, \kappa) \leftarrow PK$	$(sk, PK) \leftarrow SK$; $(pk, ck, \kappa) \leftarrow PK$
$ck \leftarrow \mathsf{CKG}(1^k)$	$\alpha \leftarrow \{0,1\}^k$	$(\mathsf{tag}, c) \leftarrow C$
$\kappa \leftarrow \mathsf{FKG}(1^k)$	$\beta \leftarrow \mathsf{F}_\kappa(\alpha)$	$\alpha \leftarrow \mathsf{TDec}(sk, \mathsf{tag}, c)$
$PK \leftarrow (pk, ck, \kappa)$	Parse β as (r, r', K)	If $\alpha = \perp$ then return $\perp$.
$SK \leftarrow (sk, PK)$	$\in \{0,1\}^{\ell+\ell'+k}$.	$\beta \leftarrow \mathsf{F}_\kappa(\alpha)$
Return (PK, SK).	$\mathsf{tag} \leftarrow \mathsf{Com}(ck, \alpha; r')$	Parse β as $(r, r', K) \in \{0,1\}^{\ell+\ell'+k}$.
	$c \leftarrow \mathsf{TEnc}(pk, \mathsf{tag}, \alpha; r)$	If $\mathsf{TEnc}(pk, \mathsf{tag}, \alpha; r) = c$
	$C \leftarrow (\mathsf{tag}, c)$	and $\mathsf{Com}(ck, \alpha; r') = \mathsf{tag}$
	Return (C, K).	then return K else return $\perp$.

Fig. 5. The proposed CCA secure KEM Γ

$\{0,1\}^{\ell'}$, respectively, for some positive polynomials $\ell = \ell(k)$ and $\ell' = \ell'(k)$. Let $\mathcal{F} = (\mathsf{FKG}, \mathsf{F})$ be a function family with input length k and output length $\ell(k) + \ell'(k) + k$. Then, our proposed KEM $\Gamma = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ is constructed as in Fig. 5.

Alternative Decapsulation Algorithm. To show the CCA security of the proposed KEM Γ, it is useful to consider the following alternative decapsulation algorithm $\mathsf{AltDecap}$. For a k-bit string $\mathsf{tag}^* \in \{0,1\}^k$ and a key pair (PK, SK) output by $\mathsf{KKG}(1^k)$, where $PK = (pk, ck, \kappa)$ and $SK = (sk, PK)$, we define an "alternative" secret key $\widehat{SK}_{\mathsf{tag}^*}$ associated with $\mathsf{tag}^* \in \{0,1\}^k$ by $\widehat{SK}_{\mathsf{tag}^*} = (\mathsf{tag}^*, \widehat{sk}_{\mathsf{tag}^*}, PK)$, where $\widehat{sk}_{\mathsf{tag}^*} = \mathsf{Punc}(sk, \mathsf{tag}^*)$. $\mathsf{AltDecap}$ takes an "alternative" secret key $\widehat{SK}_{\mathsf{tag}^*}$ defined as above and a ciphertext $C = (\mathsf{tag}, c)$ as input, and runs as follows:

$\mathsf{AltDecap}(\widehat{SK}_{\mathsf{tag}^*}, C)$**:** If $\mathsf{tag}^* = \mathsf{tag}$, then return $\perp$. Otherwise, run in exactly the same way as $\mathsf{Decap}(SK, C)$, except that "$\alpha \leftarrow \widehat{\mathsf{TDec}}(\widehat{sk}_{\mathsf{tag}^*}, \mathsf{tag}, c)$" is executed instead of "$\alpha \leftarrow \mathsf{TDec}(sk, \mathsf{tag}, c)$."

The following lemma is easy to see due to the correctness of the underlying PTBE scheme $\mathcal{T}$ and the validity check of c by re-encryption performed at the last step.

Lemma 7. *Let* $\mathsf{tag}^* \in \{0,1\}^k$ *be a string and let* (PK, SK) *be a key pair output by* $\mathsf{KKG}(1^k)$*. Furthermore, let* $\widehat{SK}_{\mathsf{tag}^*}$ *be an alternative secret key as defined above. Then, for any ciphertext* $C = (\mathsf{tag}, c)$ *(which could be outside the range of* $\mathsf{Encap}(PK)$*) satisfying* $\mathsf{tag} \neq \mathsf{tag}^*$*, it holds that* $\mathsf{Decap}(SK, C) = \mathsf{AltDecap}(\widehat{SK}_{\mathsf{tag}^*}, C)$.

Security of Γ. The security of Γ is guaranteed by the following theorem.

Theorem 3. *Assume that the PTBE scheme $\mathcal{T}$ is* eCPA *secure, the commitment scheme $\mathcal{C}$ is hiding and target-binding, and the function family $\mathcal{F}$ is* $\mathtt{UCE}[\mathbf{S}^{\mathrm{cup}}_{t,1}]$ *secure with* $t = O(\mathsf{t}_{\mathsf{TKG}} + \mathsf{t}_{\mathsf{TEnc}} + \mathsf{t}_{\mathsf{Punc}} + \mathsf{t}_{\mathsf{CKG}} + \mathsf{t}_{\mathsf{Com}})$*. Then, the KEM Γ constructed as in Fig. 5 is* CCA *secure.*

Proof Sketch of Theorem 3. Let $\mathcal{A}$ be any PPTA adversary that attacks the KEM Γ in the sense of CCA security. Consider the following sequence of games: (Here, the values with asterisk (*) represent those related to the challenge ciphertext for $\mathcal{A}$.)

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

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

Game 3: Same as Game 2, except that all decapsulation queries C are answered with $\mathsf{AltDecap}(\widehat{SK}_{\mathsf{tag}^*}, C)$, where $\widehat{SK}_{\mathsf{tag}^*}$ is the alternative secret key corresponding to (PK, SK) and tag^*.

Game 4: Same as Game 3, except that r^*, r'^*, K_1^* are picked uniformly at random, independently of $\beta^* = \mathsf{F}_\kappa(\alpha^*)$. That is, the steps "$\beta^* \leftarrow \mathsf{F}_\kappa(\alpha^*)$; Parse β^* as $(r^*, r'^*, K_1^*) \in \{0,1\}^{\ell+\ell'+k}$" in Game 3 are replaced with the step "$(r^*, r'^*, K_1^*) \leftarrow \{0,1\}^{\ell+\ell'+k}$," and we do not compute β^* anymore.

For $i \in [4]$, let S_i denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game i. Note that $\mathsf{Adv}^{\mathtt{CCA}}_{\Gamma,\mathcal{A}}(k) = 2 \cdot |\Pr[\mathsf{S}_1] - 1/2| \leq 2 \cdot \sum_{i\in[3]} |\Pr[\mathsf{S}_i] - \Pr[\mathsf{S}_{i+1}]| + 2 \cdot |\Pr[\mathsf{S}_4] - 1/2|$. We will show that $|\Pr[\mathsf{S}_i] - \Pr[\mathsf{S}_{i+1}]|$ is negligible for each $i \in [3]$ and that $\Pr[\mathsf{S}_4] = 1/2$, which proves the theorem.

Firstly, notice that $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]|$ can be upperbounded by the probability of $\mathcal{A}$ making a decapsulation query $C = (\mathsf{tag}, c)$ satisfying $\mathsf{tag} = \mathsf{tag}^*$, $c \neq c^*$, and $\mathsf{Decap}(SK, C) \neq \perp$. In the full proof, we will show that such a query can be used to break the target-binding property of the commitment scheme $\mathcal{C}$, and hence $\mathcal{A}$ will submit a query of this type only with negligible probability, due to the target-binding property of the commitment scheme $\mathcal{C}$.

It is easy to see that $\Pr[\mathsf{S}_2] = \Pr[\mathsf{S}_3]$ holds, because the behavior of the oracle in Game 2 and that in Game 3 are identical due to Lemma 7.

To show the upperbound of $|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$, we need to use the $\mathtt{UCE}[\mathsf{S}^{\mathtt{cup}}_{t,1}]$ security of the function family $\mathcal{F}$. Define the source $\mathcal{S}$ that takes 1^k as input, expects to have access to an oracle $\mathcal{O} \in \mathsf{Func}_{k\to(\ell+\ell'+k)}$, and computes an output (leakage) $L = (pk, ck, \mathsf{tag}^*, \widehat{sk}_{\mathsf{tag}^*}, c^*, K^*)$ in the following way:

$$\mathcal{S}^{\mathcal{O}}(1^k) : [\ (pk, sk) \leftarrow \mathsf{TKG}(1^k);\ ck \leftarrow \mathsf{CKG}(1^k);\ \alpha^* \leftarrow \{0,1\}^k;\ \beta^* \leftarrow \mathcal{O}(\alpha^*);$$
$$\text{Parse } \beta^* \text{ as } (r^*, r'^*, K^*).;\ \mathsf{tag}^* \leftarrow \mathsf{Com}(ck, \alpha^*; r'^*);\ \widehat{sk}_{\mathsf{tag}^*} \leftarrow \mathsf{Punc}(sk, \mathsf{tag}^*);$$
$$c^* \leftarrow \mathsf{TEnc}(pk, \mathsf{tag}^*, \alpha^*; r^*);\ \text{Return } L \leftarrow (pk, ck, \mathsf{tag}^*, \widehat{sk}_{\mathsf{tag}^*}, c^*, K^*).\ ].$$

Defined as above, it is obvious that $\mathcal{S}$ satisfies the restrictions on the running time and the number of queries. Furthermore, due to the hiding property of the commitment scheme $\mathcal{C}$ and the $\mathtt{eCPA}$ security of the PTBE scheme $\mathcal{T}$, it is straightforward to see that $\mathcal{S}$ is computationally unpredictable, and thus it holds that $\mathcal{S} \in \mathsf{S}^{\mathtt{cup}}_{t,1}$. Then, in the full proof, we will show that there exists a PPTA $\mathcal{B}_{\mathtt{u}}$ that takes as input a function index κ, a leakage $L = (pk, ck, \mathsf{tag}^*, \widehat{sk}_{\mathsf{tag}^*}, c^*, K^*) \leftarrow \mathcal{S}^{\mathcal{O}}(1^k)$, where $\mathcal{O} \in \mathsf{Func}_{k\to(\ell+\ell'+k)}$ is either $\mathsf{F}_\kappa(\cdot)$ or a random function, simulates Game 3 or Game 4 perfectly for $\mathcal{A}$ depending on $\mathcal{B}_{\mathtt{u}}$'s challenge bit, and has the $\mathtt{UCE}$ advantage $\mathsf{Adv}^{\mathtt{UCE}}_{\mathcal{F},(\mathcal{S},\mathcal{B}_{\mathtt{u}})}(k) = |\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$. Hence, $|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$ is negligible by the $\mathtt{UCE}[\mathsf{S}^{\mathtt{cup}}_{t,1}]$ security of $\mathcal{F}$.

Finally, in Game 4, the "real" session-key K_1^* is independent of the challenge ciphertext C^* and is a uniformly random value, and thus the challenge bit b is information-theoretically hidden from $\mathcal{A}$'s view. This implies $\Pr[\mathsf{S}_4] = 1/2$. □

5.2 Further Results and Extensions

$\mathtt{CCA}$ *Secure DPKE for Block Sources with Bounded Running Time.* Note that our proposed KEM has the property that a randomness used to generate a ciphertext is entirely recovered in the decryption process. Here, by deriving the randomness r and r' (used for generating c and tag) from a plaintext m (instead of deriving them from the "seed" α picked randomly) by the UCE secure function family $\mathcal{F}$, we obtain a DPKE scheme. We can show that this DPKE scheme is $\mathtt{CCA}$ secure for block sources [10] (i.e. each plaintext sampled from the source has high min-entropy, even conditioned on all the previous plaintexts), as long as the sources satisfy an additional constraint that their running time is bounded by some predetermined polynomial $t' = t'(k)$ (we call such a block source t'*-bounded block source*). This additional constraint on the running time of the sources is due to our security proof in which the source for a UCE secure function family has to execute a t'-bounded block source for DPKE (that chooses the challenge plaintexts), and thus we have to rely on $\mathtt{UCE}[\mathtt{S}^{\mathtt{cup}}_{t,1}]$ security where t must be large enough to allow the execution of the t'-bounded block source for DPKE (and other algorithms that need to be run for the security proof).

Although $\mathtt{CCA}$ security for block sources with bounded running time is clearly weaker than that for ordinary block sources, the constraint on the running time of the sources would not be a severe limitation in practice, because in most cases messages that are going to be encrypted will be chosen by honest parties and we do not expect picking messages to be computationally expensive.[7] We stress that we do not put any restriction on the running time of the "main" adversary who may perform decryption queries and any computationally heavy operations, as long as it runs in polynomial time.

Function Families with Short Output Length. For our proposed KEM, we use a function family $\mathcal{F}$ with output length $\ell + \ell' + k$, which could be long (the actual length depends on how the PTBE scheme is instantiated). However, by employing a pseudorandom generator $\mathsf{G} : \{0,1\}^k \to \{0,1\}^{\ell+\ell'+k}$, we can replace $\mathcal{F}$ with a function family with output length k. This extension is however at the cost of using slightly stronger UCE security. Specifically, now we have to rely on the $\mathtt{UCE}[\mathtt{S}^{\mathtt{cup}}_{t',1}]$ security where $t' = t + \mathsf{t}_{\mathsf{G}}$ and t is as stated in Theorem 3. This extension is also applicable to our DPKE scheme.

Weakening the UCE Assumption Using Lossy Encryption. We notice that in the security proof of our proposed KEM, if the underlying PTBE scheme is instantiated using a *lossy encryption* scheme [6] and the underlying commitment scheme is statistically hiding (which can be constructed from any lossy encryption scheme), then the source $\mathcal{S}$ used in the proof of Theorem 3 can be modified to show that it is statistically unpredictable. Specifically, this can be shown by considering an additional game between Game 3 and Game 4 in which we use lossy public keys for public keys corresponding to tag^* when generating a challenge ciphertext. (For this, in the full version of our paper we will also introduce a lossy-encryption-analogue of PTBE.)

Therefore, at the cost of employing a stronger assumption on the underlying PKE scheme, we can weaken the assumption on $\mathcal{F}$ to be $\mathtt{UCE}[\mathtt{S}^{\mathtt{sup}}_{t',1}]$ security where t' is

[7] This observation is due to one of the anonymous reviewers.

dependent on the underlying lossy encryption scheme (and other building blocks). (We will specify t' in the full version.)

We note that similar tradeoffs about the assumptions among building blocks for constructing CCA secure PKE/KEM were shown in [29].

Acknowledgement. The authors would like to thank Pooya Farshim for giving us a detailed overview of their attack [12] on UCE security using indistinguishability obfuscation. The authors would also like to thank Jacob Schuldt, the members of the study group "Shin-Akarui-Angou-Benkyou-Kai," and the anonymous reviewers of PKC 2014 for their helpful comments and suggestions. In particular, the authors are grateful to one of the reviewers for pointing out some issue in the security proof of our DPKE scheme, and for suggesting considering CCA security of DPKE for block sources with bounded running time.

References

1. Abdalla, M., Bellare, M., Rogaway, P.: The oracle Diffie-Hellman assumptions and an analysis of DHIES. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 143–158. Springer, Heidelberg (2001)
2. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)
3. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
4. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 398–415. Springer, Heidelberg (2013)
5. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. Updated full version of [4] (2013), http://eprint.iacr.org/2013/424
6. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
7. Bellare, M., Keelveedhi, S., Ristenpart, T.: Message-locked encryption and secure deduplication. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 296–312. Springer, Heidelberg (2013)
8. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS 1993, pp. 62–73 (1993)
9. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
10. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
11. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
12. Brzuska, C., Farshim, P., Mittelbach, A.: Personal communication (December 2013)

13. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: STOC 1998, pp. 209–218 (1998)
14. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
15. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Black-box construction of a non-malleable encryption scheme from any semantically secure one. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 427–444. Springer, Heidelberg (2008)
16. Cramer, R., Hanaoka, G., Hofheinz, D., Imai, H., Kiltz, E., Pass, R., Shelat, A., Vaikuntanathan, V.: Bounded CCA2-secure encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 502–518. Springer, Heidelberg (2007)
17. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Computing 33(1), 167–226 (2003)
18. Dachman-Soled, D.: A black-box construction of a CCA2 encryption scheme from a plaintext aware (sPA1) encryption scheme. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 37–55. Springer, Heidelberg (2014), http://eprint.iacr.org/2013/680
19. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: STOC 1991, pp. 542–552 (1991)
20. Fujisaki, E., Okamoto, T.: How to enhance the security of public-key encryption at minimum cost. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 53–68. Springer, Heidelberg (1999)
21. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
22. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all curcuits. In: FOCS 2013, pp. 40–49 (2013)
23. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: FOCS 2003, pp. 102–113 (2003)
24. Hohenberger, S., Lewko, A., Waters, B.: Detecting dangerous queries: A new approach for chosen ciphertext security. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 663–681. Springer, Heidelberg (2012)
25. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
26. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive trapdoor functions and chosen-ciphertext security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)
27. Lin, H., Tessaro, S.: Amplification of chosen-ciphertext security. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 503–519. Springer, Heidelberg (2013)
28. MacKenzie, P.D., Reiter, M.K., Yang, K.: Alternatives to non-malleability: Definitions, constructions, and applications. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 171–190. Springer, Heidelberg (2004)
29. Matsuda, T., Hanaoka, G.: Chosen ciphertext security via point obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 95–120. Springer, Heidelberg (2014)
30. Matsuda, T., Matsuura, K.: Parallel decryption queries in bounded chosen ciphertext attacks. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 246–264. Springer, Heidelberg (2011)
31. Maurer, U.M., Renner, R.S., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004)

32. Myers, S., Shelat, A.: Bit encryption is complete. In: FOCS 2009, pp. 607–616 (2009)
33. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990, pp. 427–437 (1990)
34. Okamoto, T., Pointcheval, D.: REACT: Rapid enhanced-security asymmetric cryptosystem transform. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 159–174. Springer, Heidelberg (2001)
35. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008, pp. 187–196 (2008)
36. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
37. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
38. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: Deniable encryption, and more (2013), http://eprint.iacr.org/2013/454
39. Wee, H.: Efficient chosen-ciphertext security via extractable hash proofs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 314–332. Springer, Heidelberg (2010)