Chosen Ciphertext Security via Point Obfuscation*

Takahiro Matsuda and Goichiro Hanaoka

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

Abstract. In this paper, we show two new constructions of chosen ciphertext secure (CCA secure) public key encryption (PKE) from general assumptions. The key ingredient in our constructions is an obfuscator for point functions with multi-bit output (MBPF obfuscators, for short), that satisfies some (average-case) indistinguishability-based security, which we call AIND security, in the presence of hard-to-invert auxiliary input. Specifically, our first construction is based on a chosen plaintext secure PKE scheme and an MBPF obfuscator satisfying the AIND security in the presence of computationally hard-to-invert auxiliary input. Our second construction is based on a lossy encryption scheme and an MBPF obfuscator satisfying the AIND security in the presence of statistically hard-to-invert auxiliary input. To clarify the relative strength of AIND security, we show the relations among security notions for MBPF obfuscators, and show that AIND security with computationally (resp. statistically) hard-to-invert auxiliary input is implied by the average-case virtual black-box (resp. virtual grey-box) property with the same type of auxiliary input. Finally, we show that a lossy encryption scheme can be constructed from an obfuscator for point functions (point obfuscator) that satisfies re-randomizability and a weak form of composability in the worst-case virtual grey-box sense. This result, combined with our second generic construction and several previous results on point obfuscators and MBPF obfuscators, yields a CCA secure PKE scheme that is constructed *solely* from a re-randomizable and composable point obfuscator. We believe that our results make an interesting bridge that connects CCA secure PKE and program obfuscators, two seemingly isolated but important cryptographic primitives in the area of cryptography.

Keywords: public key encryption, lossy encryption, key encapsulation mechanism, chosen ciphertext security, point obfuscation.

* An extended abstract appears in the proceedings of TCC 2014. This is the full version.

Table of Contents

1 Introduction

1.1 Background and Motivation

One of the fundamental research themes in cryptography is to clarify what the minimal assumptions to realize various kinds of cryptographic primitives are, and up to now, a number of relationships among primitives have been investigated and established. Clarifying these relationships gives us a lot of insights for how to construct and/or prove the security of cryptographic primitives, enables us to understand the considered primitives more deeply, and leads to systematizing the research area in cryptography.

In this paper, we focus on the constructions of public key encryption (PKE) schemes secure against chosen ciphertext attacks (CCA) [65, 35] from general cryptographic assumptions. CCA secure PKE is one of the most important cryptographic primitives that has been intensively studied, due to its resilience against practical attacks such as [10], and its implication to many useful security notions, such as non-malleability [35] and universal composability [22].

The first successful result regarding this line of research is the construction by Dolev, Dwork, and Naor [35] that uses a chosen plaintext secure (CPA secure) PKE scheme [45] and a non-interactive zero-knowledge proof [11]. Since these two primitives can be constructed from (an enhanced variant of) trapdoor permutations (TDP) [42], CCA secure PKE can be constructed solely from TDPs. Canetti, Halevi, and Katz [24] showed that CCA secure PKE can be constructed from an identity-based encryption (IBE) [69, 12]. It was later shown that in fact, a weaker primitive called tag-based encryption suffices [58, 54]. Peikert and Waters [64] showed that CCA secure PKE can be constructed from any lossy trapdoor function (TDF), and subsequent works showed that injective TDFs with weaker properties suffice: injective TDFs secure for correlated inputs [66], slightly lossy TDFs [60], adaptive one-way TDFs [55], and adaptive one-way trapdoor relations [71]. (CPA secure) PKE schemes with additional security/functional properties have also turned out to be useful for constructing CCA secure PKE: Hemenway and Ostrovsky [49] showed that we can construct CCA secure PKE in several ways from homomorphic encryption with appropriate properties. The same authors [50] also showed that CCA secure PKE can be constructed from a lossy encryption scheme [6] if the plaintext space is larger than the randomness space. Hohenberger, Lewko, and Waters [51] showed that if one has a PKE scheme which satisfies the notion called detectable CCA security, which is somewhere between CCA1 and CCA2 security, then using it one can construct a CCA secure PKE scheme. Myers and Shelat [61] showed how to construct a CCA secure PKE scheme that can encrypt plaintexts with arbitrary length from a CCA secure one with 1-bit plaintext space. Lin and Tessaro [56] showed how to amplify weak CCA security. Very recently, Dachman-Soled [31] showed a construction of CCA secure PKE from PKE satisfying (standard model) plaintext-awareness together with some additional simulatability property, and Matsuda and Hanaoka [59] showed a construction from CPA secure PKE and a family of hash functions satisfying the security notion called *universal computational extractors* (UCE security) [5].

The main purpose of this work is to show that a different kind of cryptographic primitives is also useful for achieving CCA secure PKE. Specifically, we add new recipes for the construction of CCA secure PKE, based on the techniques and results from (cryptographic) *program obfuscation* [3] for the very simple classes of functions, *point functions* and *point functions with multi-bit output*. Despite the tremendous efforts, it is not known whether it is possible to construct CCA secure PKE only from CPA secure one (in fact, a partial negative result is known [40]). Clarifying new classes of primitives that serve as building blocks is important for tackling this problem. In particular, it was shown that there is no black-box construction of IBE and a TDF from (CCA secure) PKE [14, 41], and thus to tackle the "CPA-to-CCA" problem, the attempts to construct IBE or the above TDF-related primitives from a CPA secure PKE scheme seem hopeless (though there is a possibility

that some non-black-box construction exists). Our new constructions based on (multi-bit) point obfuscators do not seem to be covered by this negative result, and thus it could serve as a new target for building CCA secure PKE.

1.2 Our Contributions

In this paper, we show two new constructions of CCA secure PKE schemes from general assumptions, using the techniques and results from program obfuscation [3]. We actually construct CCA secure key encapsulation mechanisms (KEMs) [30], where a KEM is the "PKE"-part of hybrid encryption that encrypts a random "session-key" for symmetric key encryption (SKE). By combining a CCA secure KEM with a CCA secure SKE scheme, one obtains a CCA secure PKE scheme [30]. The key ingredient in our constructions is an obfuscator for point functions with multi-bit output (MBPF obfuscators) [57, 23, 33, 44, 25, 7], that satisfies a kind of average-case indistinguishability-based security in the presence of "hard-to-invert" auxiliary inputs. The formal definition of this security notion will be given in Section 3. For brevity, we call it AIND security.

Our first construction in Section 4.1 is based on a CPA secure PKE scheme and an MBPF obfuscator satisfying the above mentioned AIND security in the presence of computationally hard-to-invert auxiliary input. Our second construction in Section 4.2 is based on a lossy encryption scheme [6] and an MBPO satisfying the above mentioned AIND security in the presence of statistically hard-to-invert auxiliary input. Interestingly, the first and the second constructions are in fact exactly the same, and we show two different security analyses from different assumptions on building blocks. These two constructions add new recipes into the current picture of the constructions of CCA secure PKE schemes/KEMs from general assumptions.

In order to clarify where these AIND security definitions for MBPF obfuscators are placed, in Section 5 we show that AIND security with computationally (resp. statistically) hard-to-invert auxiliary inputs is implied by the (average-case) virtual black-box property [3] (resp. virtual grey-box property [7]) in the presence of the same auxiliary inputs. Besides these, we show the relations among several related worst-/average-case virtual black-/grey-box properties under several types of auxiliary inputs, and summarize them in Fig. 2, which we believe is useful for further research on this topic and might be of independent interest.

Finally, in Section 6, we show that a lossy encryption scheme can be constructed from an obfuscator for point functions (point obfuscator) that satisfies re-randomizability [7] and a weak form of composability [57, 23, 7] in the worst-case virtual grey-box sense. This result, combined with our second generic construction and the results on composable point obfuscators with the virtual grey-box property in [7], shows that a CCA secure PKE scheme can be constructed *solely* from a point obfuscator which is re-randomizable and composable.

We believe that our results make an interesting bridge that connects CCA secure PKE and program obfuscators,[1] two seemingly isolated but important cryptographic primitives that have been separately studied in the area of cryptography, and hope that our results motivate further studies on them.

1.3 Overview of Techniques

Our proposed constructions of KEMs are based on the "witness-recovering" technique [64, 66, 61, 51] in which a part of randomness used to generate a ciphertext is somehow embedded into the ciphertext itself, and is later recovered in the decryption process for checking the validity of the

[1] Recently, Sahai and Waters [68] showed how to construct (among other primitives) CCA secure PKE using *indistinguishability obfuscation* [3, 39]. We explain the difference with our results in Section 1.4.

ciphertext by re-encryption. What we believe is novel in our constructions is how to implement this mechanism of witness-recovering by using an MBPF obfuscator with an appropriate security property.

Let $\mathcal{I}_{\alpha\to\beta}$ denote an MBPF such that $\mathcal{I}_{\alpha\to\beta}(x)=\beta$ if $x=\alpha$ and $\perp$ otherwise, and let MBPO denotes an MBPF obfuscator which takes an MBPF $\mathcal{I}_{\alpha\to\beta}$ as input, and outputs an obfuscated circuit $\mathtt{DL}$ for $\mathcal{I}_{\alpha\to\beta}$. ("$\mathtt{DL}$" stands for "digital locker," the name due to [23].) Let $\Pi=(\mathsf{PKG},\mathsf{Enc},\mathsf{Dec})$ be a PKE scheme, where PKG, Enc, and Dec are the key generation, the encryption, and the decryption algorithms of Π, respectively.

Below we give a high level idea behind our main proposed constructions in Section 4 (and in Appendix E) by explaining how the "toy" version of our constructions $\Pi'=(\mathsf{PKG}',\mathsf{Enc}',\mathsf{Dec}')$, constructed using Π and MBPO, is proved $\mathtt{CCA1}$ secure based on the assumptions that Π is $\mathtt{CPA}$ secure and that MBPO satisfies the virtual black-box property with respect to dependent auxiliary input [43]. (As mentioned earlier, in this paper we actually construct KEMs rather than PKE schemes, but the intuition for our results are captured by the explanation here.) A public/secret key pair (PK,SK) of Π' is of the form $PK=(pk_1,pk_2)$, $SK=(sk_1,sk_2)$, where each (pk_i,sk_i) is an independently generated key pair by running PKG. To encrypt a plaintext m under PK, Enc' first picks a random string $\alpha\in\{0,1\}^k$ (where k is the security parameter) and two randomness r_1 and r_2 for Enc, and computes a ciphertext C in the following way:

$$C=(c_1,c_2,\mathtt{DL})=\Big(\mathsf{Enc}(pk_1,(m\|\alpha);r_1),\ \mathsf{Enc}(pk_2,(m\|\alpha);r_2),\ \mathsf{MBPO}(\mathcal{I}_{\alpha\to(r_1\|r_2)})\Big),$$

where "$\|$" denotes the concatenation of strings, and "$\mathsf{Enc}(pk,m;r)$" means to encrypt the plaintext m under the public key pk using the randomness r. To decrypt C, we first decrypt c_1 by using sk_1 to obtain $(m\|\alpha)$, then run $\mathtt{DL}(\alpha)$ to recover $(r_1\|r_2)$. Finally, m is returned if $c_i=\mathsf{Enc}(pk_i,(m\|\alpha);r_i)$ holds for both $i=1,2$, and otherwise we reject C. Here, it should be noted that due to the symmetric roles of pk_1 and pk_2 and the validity check by re-encryption performed in Dec', we can also decrypt C using sk_2, so that the decryption result of C using sk_1 and that using sk_2 always agree.

Now, recall the interface of a $\mathtt{CCA1}$ adversary $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$, where $\mathcal{A}_1$ and $\mathcal{A}_2$ represent an adversary's algorithm before and after the challenge, respectively. $\mathcal{A}_1$ is firstly given a public key PK, and can start using the decryption oracle $\mathsf{Dec}'(SK,\cdot)$. After that, $\mathcal{A}_1$ terminates with output two plaintexts (m_0,m_1) and some state information st that is passed to $\mathcal{A}_2$. $\mathcal{A}_2$ is given st and the challenge ciphertext $C^*=(c_1^*,c_2^*,\mathtt{DL}^*)$ which is an encryption of m_b (where b is the challenge bit), and outputs a bit as its guess for b.

The key observation is that $\mathcal{A}_2$ can be seen as an adversary for the MBPF obfuscator MBPO, by regarding $(\mathsf{st},c_1^*,c_2^*)$ as an auxiliary input z about the obfuscated circuit $\mathtt{DL}^*$ of the MBPF $\mathcal{I}_{\alpha^*\to(r_1^*\|r_2^*)}$. Then, if MBPO satisfies the virtual black-box property with respect to dependent auxiliary input [43], there exists a simulator $\mathcal{S}$ that takes only $z=(\mathsf{st},c_1^*,c_2^*)$ as input, has oracle access to $\mathcal{I}_{\alpha^*\to(r_1^*\|r_2^*)}$, and has the property that $\mathcal{A}$'s success probability (in guessing b) is negligibly close to the probability that $\mathcal{S}$ succeeds in guessing b. (For convenience, let us call the latter probability "$\mathcal{S}$'s success probability," although $\mathcal{S}$ is not a $\mathtt{CCA1}$ adversary and thus its task is not to guess a challenge bit.) This means that if $\mathcal{S}$'s success probability is close to $1/2$, then so is $\mathcal{A}$'s success probability, which will prove the $\mathtt{CCA1}$ security of Π'.

To show that $\mathcal{S}$'s success probability is close to $1/2$, we consider the hypothetical experiment for $\mathcal{S}$ in which the auxiliary input z is generated so that decryption queries from $\mathcal{A}_1$ are answered using sk_2, and both c_1^* and c_2^* are an encryption of a fixed value (say, $0^{|m_0|+k}$). Since z does not contain any information on b and α^*, in this hypothetical experiment $\mathcal{S}$'s success probability is exactly $1/2$ and the probability that $\mathcal{S}$ makes the query α^* (which is chosen randomly) is negligible. Next, we make

the experiment closer to the actual $\mathcal{S}$'s experiment, by changing c_1^* into an encryption of $(m_b \| \alpha^*)$. By the CPA security regarding pk_1, $\mathcal{S}$'s success probability as well as the probability of $\mathcal{S}$ making the query α^* is negligibly close to those in the hypothetical experiment. Then, we further modify the previous experiment by changing c_2^* into an encryption of $(m_b \| \alpha^*)$, but this time we use sk_1 for answering $\mathcal{A}_1$'s queries. Notice that this is exactly the actual experiment for $\mathcal{S}$. As mentioned above, switching sk_2 to/from sk_1 for answering $\mathcal{A}_1$'s queries does not affect $\mathcal{A}_1$'s behavior, and thus again by the CPA security regarding pk_2, $\mathcal{S}$'s success probability is negligibly close to $1/2$ and the probability that $\mathcal{S}$ makes the query α^* is negligible. Then, by the virtual black-box property of MBPO with auxiliary input, $\mathcal{A}$'s original success probability is negligibly close to $1/2$, meaning that $\mathcal{A}$ has negligible advantage in breaking the CCA1 security of the scheme Π'.

The above completes the proof sketch of how Π' is proved CCA1 secure. By encrypting a random K, Π' can be considered as a CCA1 secure KEM. Our proposed CCA2 secure KEMs are obtained by applying several optimizations and enhancement to this KEM:

- Firstly, we do not need the full virtual black-box property with auxiliary input of [43]. As mentioned earlier, an indistinguishability-based definition in the presence of only "hard-to-invert" auxiliary input is sufficient for a similar argument to work.
- Secondly, we need not include a plaintext into each of c_i. Instead, we pick a randomness $K \in \{0,1\}^k$ used as a plaintext of a KEM, and include this K into the output of the MBPF, i.e. now we obfuscate the MBPF $\mathcal{I}_{\alpha \to (r_1 \| r_2 \| K)}$. This is the actual basic version of our construction whose formal description and security proof are given in Appendix E.
- Lastly, note that the above construction cannot be proved to be CCA1 secure as it is. In particular, the obfuscated circuit DL could be malleable. To deal with this issue, instead of the Naor-Yung-style double encryption [63], we employ the Dolev-Dwork-Naor-style multiple encryption [35] together with the technique of the "unduplicatable set selection" [67]. Unlike the classical method of using a one-time signature scheme, in our proposed construction we employ a universal one-way hash function (UOWHF) [62], where a hash value of DL is used as a "selector" of the public key components (for multiple encryption). Another issue is that the second stage adversary $\mathcal{A}_2$ in the CCA2 experiment can also make decryption queries, and thus the above explained idea of replacing $\mathcal{A}_2$ with a simulator $\mathcal{S}$ does not work. However, our indistinguishability-based security definition for MBPF obfuscators enables us to work with an original CCA2 adversary, and we can avoid considering how a simulator deal with the queries from $\mathcal{A}_2$. For more details, see Section 4.

1.4 Related Work: Program Obfuscation

Roughly speaking, an obfuscator is an algorithm that takes a program (e.g. Turing machine or circuit) as input, and outputs another program with the same functionality, but otherwise "unintelligible."

After the impossibility of general-purpose program obfuscation satisfying the nowadays standard security notion called *virtual black-box* property shown in the seminal work by Barak et. al. [3], several subsequent works extended the impossibility in various other settings [43, 70, 46, 7]. The other line of research pursues possibilities of obfuscating a specific class of functions. Before 2013, most known positive results were about obfuscation for point functions and their variants, e.g. [57, 70, 23, 27, 7]. Relaxing the security requirements to "average-case" in which a program is sampled according to some distribution, several more complex tasks have been shown to be obfuscatable, such as proximity testing [34] and cryptographic tasks such as re-encryption [52, 28] and encrypted signatures [47]. The goal of these works was obfuscation itself, while our work uses a positive result

on obfuscation of (multi-bit) point functions as a tool to construct other cryptographic primitive. One of the previous works which has the same spirit as ours is the work by Bitansky and Paneth [9] who showed how to construct a three-round weak zero-knowledge protocol for NP, which is known to be impossible via black-box simulation, using a point obfuscator and an MBPF obfuscator as a part of building blocks. In fact, our work is partly inspired by their use of point obfuscation. We note that the security required for obfuscators in our proposed construction is weaker than one used in [9] to achieve their weak zero-knowledge protocol.

Since the first candidates of a cryptographic multilinear map have been proposed in 2013 [37, 29], the research field of (cryptographic) obfuscation has drastically changed and accelerated. Brakerski and Rothblum [17] showed how to construct an obfuscator for conjunctions from graded encoding schemes [37, 29], and the same authors showed a further extension [18]. Most recently, they showed a general-purpose obfuscator satisfying a virtual black-box property in an idealized model called the generic graded encoded scheme model [19]. Barak et al. [2] studied obfuscation for a class of functions called *evasive functions* which in particular includes point functions. A series of works [39, 68, 53, 38] (and many other recent works) have shown that a general-purpose obfuscator satisfying a security notion weaker than the virtual black-box property, called *indistinguishability obfuscator* [3], which seems to be too weak to be useful, is in fact surprisingly powerful and can be used as a building block for constructing a various kinds of cryptographic primitives. Garg et al. [39] constructed the first candidate of general-purpose indistinguishability obfuscation. A security notion stronger than indistinguishability obfuscation, called *differing-inputs obfuscation* [3, 1] (and its closely related notion of *extractability obfuscation* [15]), has also been shown to be quite powerful and useful [1, 15].

Among a number of recent fascinating results, especially relevant to our work is the work by Sahai and Waters [68] who showed (among several other primitives) how to construct CCA secure PKE and KEMs from an indistinguishability obfuscator (and a one-way function). Although our work and [68] have the common property that both works build CCA secure PKE using techniques and results from obfuscation, our use of obfuscators and that of [68] are quite different: We use an obfuscator for a specific class of functions, point functions and MBPFs, while [68] uses an obfuscator for all polynomial-sized circuits. Furthermore, the indistinguishability-based security notion for MBPF obfuscators used in our main result is about randomly chosen MBPFs, while that used in [68] is for the worst-case choice of circuits (that compute the same functions). We would also like to stress that our work and [68] were done concurrently and independently.

1.5 Paper Organization

The rest of the paper is organized as follows: In Section 2 (and Appendix A) we review the basic notations and definitions of primitives. In Section 3, we introduce the formal definitions of our new indistinguishability-based security notions for MBPF obfuscators. In Section 4, we show our main results: two CCA secure KEMs using an MBPF obfuscator. In Section 5, we investigate relations between our new security notions and other notions for MBPF obfuscators. In Section 6, we show how to construct a lossy encryption scheme from a point obfuscator with re-randomizability and composability. In Section 7, we discuss some issues on the MBPF obfuscators that we use.

2 Preliminaries

In this section, we review the basic notation and the definitions for lossy encryption and (cryptographic) obfuscation. The basic notation and the definitions for standard cryptographic primitives that are not given in this section are given in Appendix A, which include PKE, KEMs, UOWHFs, as well as other basic primitives.

Basic Notation. $\mathbb{N}$ denotes the set of all natural numbers, and if $n \in \mathbb{N}$ then $[n] = \{1, \ldots, n\}$. "$x \leftarrow y$" denotes that x is chosen uniformly at random from y if y is a finite set, x is output from y if y is a function or an algorithm, or y is assigned to x otherwise. If x and y are strings, then "$|x|$" denotes the bit-length of x, and "$x\|y$" denotes the concatenation x and y. "$x \stackrel{?}{=} y$" is the operation that returns 1 if $x = y$ and returns 0 otherwise. "PPTA" stands for a *probabilistic polynomial time algorithm*. 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 this paper, we use the character "k" to denote a security parameter.

2.1 Lossy Encryption

Definition 1. *A tuple of PPTAs* $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{LKG})$ *is said to be an* ϵ-lossy encryption scheme[2] *if the following properties are satisfied:*

- *(**Syntax**)* $(\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ *constitutes a PKE scheme. The algorithm* LKG *is called a lossy key generation algorithm, which takes* 1^k *as input, and outputs a "lossy" public key* pk.
- *(**Indistinguishability of ordinary/lossy keys**) For all PPTAs* $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{KEY}}_{\Pi,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{KEY}}_{\Pi,\mathcal{A}}(k) = 1] - 1/2|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{KEY}}_{\Pi,\mathcal{A}}(k)$ *is defined as follows:*

  $\mathsf{Expt}^{\mathtt{KEY}}_{\Pi,\mathcal{A}}(k) : [\ (pk_0, sk) \leftarrow \mathsf{PKG}(1^k);\ pk_1 \leftarrow \mathsf{LKG}(1^k);\ b \leftarrow \{0,1\};\ b' \leftarrow \mathcal{A}(pk_b);\ \text{Return}\ (b' \stackrel{?}{=} b)\ ].$

- *(**Statistical lossiness**) For all computationally unbounded algorithms* $\mathcal{A}$ *and for all sufficiently large* $k \in \mathbb{N}$ *it holds that* $\mathsf{Adv}^{\mathtt{LOS\text{-}CPA}}_{\Pi,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{LOS\text{-}CPA}}_{\Pi,\mathcal{A}}(k) = 1] - 1/2| \leq \epsilon(k)$, *where the experiment* $\mathsf{Expt}^{\mathtt{LOS\text{-}CPA}}_{\Pi,\mathcal{A}}(k)$ *is defined in the same way as the ordinary* $\mathtt{CPA}$ *experiment* $\mathsf{Expt}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k)$ *except that the public key* pk *is generated as* $pk \leftarrow \mathsf{LKG}(1^k)$. *We call* ϵ lossiness.

2.2 Obfuscation for Circuits and Worst-Case Security Definitions

Here, we recall the definition of circuit obfuscations, following the definitions given in [3, 57, 43, 8]. In the following, by $\mathcal{C}$ we denote an ensemble $\{\mathcal{C}_k\}_{k\in\mathbb{N}}$, where $\mathcal{C}_k$ is a collection of circuits whose input length is $k \in \mathbb{N}$ and whose size is bounded by some polynomial of k.

Definition 2. *We say that a PPTA* Obf *is an obfuscator for* $\mathcal{C}$ *if it satisfies the following:*

- *(**Functionality**) For every* $k \in \mathbb{N}$ *and every* $C \in \mathcal{C}_k$, *a circuit output from* $\mathsf{Obf}(C)$ *computes the same function as* C.
- *(**Polynomial blowup**) There exists a polynomial* $p = p(k) > 0$ *such that for every* $k \in \mathbb{N}$ *and every* $C \in \mathcal{C}_k$, *the size of a circuit output from* $\mathsf{Obf}(C)$ *is bounded by* $p(k)$.

Note that the above definition is only about the functionality requirements of obfuscators.

Next, we recal the security definitions for "worst-case" choice of circuits.: The *virtual black-box property* is due to Barak et al. [3], the *virtual black-box property with auxiliary input* is due to Goldwasser and Kalai [43], and *virtual "grey"-box (with auxiliary input)* is due to Bitansky and Canetti [7].

[2] In this paper, we consider the "exact security"-style definition for lossy encryption and $\mathtt{CPA}$ secure PKE. This is to quantify the "hardness" of inverting auxiliary input functions used in the security definitions of MBPF obfuscators. For details, see Section 3.

Definition 3. *We say that an obfuscator* Obf *for* $\mathcal{C}$ *satisfies:*

- *the* worst-case virtual black-box property *(*`WVB` *security, for short), if for every PPTA* $\mathcal{A}$ *(adversary) and every positive polynomial* q*, there exists a PPTA* $\mathcal{S}$ *(simulator) such that for all sufficiently large* $k \in \mathbb{N}$ *and all circuits* $C \in \mathcal{C}_k$*, it holds that*

$$|\Pr[\mathcal{A}(1^k, \mathsf{Obf}(C)) = 1] - \Pr[\mathcal{S}^C(1^k) = 1]| \leq 1/q,$$

- *the* worst-case virtual black-box property w.r.t. auxiliary input *(*`WVB-AI` *security, for short), if for every PPTA* $\mathcal{A}$ *and every positive polynomials* q *and* ℓ*, there exists a PPTA* $\mathcal{S}$ *such that all sufficiently large* $k \in \mathbb{N}$*, all circuits* $C \in \mathcal{C}_k$*, and all strings* $z \in \{0,1\}^{\ell(k)}$*, it holds that*

$$|\Pr[\mathcal{A}(1^k, z, \mathsf{Obf}(C)) = 1] - \Pr[\mathcal{S}^C(1^k, z) = 1]| \leq 1/q,$$

where the probabilities are over the randomness consumed by Obf*,* $\mathcal{A}$*, and* $\mathcal{S}$*. Furthermore, we define the* worst-case virtual grey-box property *(*`WVG` *security), and the* worst-case virtual grey-box property w.r.t. auxiliary input *(*`WVG-AI` *security) of* Obf*, in the same way as the definitions for the corresponding virtual black-box properties, except that we replace "a PPTA* $\mathcal{S}$*" in each definition with "a computationally unbounded algorithm* $\mathcal{S}$ *that makes only polynomially many queries."*

Note that in the above definitions, the simulator $\mathcal{S}$ can depend on the polynomial q which represents the hardness of obfuscation. Wee [70] refers to the simulators of this type as a "weak simulator."

We also define (t-)composability of obfuscations [57, 23, 7, 25]. Following [8], we only define the composability in the grey-box (`WVG`) notion, using a computationally unbounded simulator, which is sufficient for our purpose in this paper.

Definition 4. *([7]) Let* $t = t(k) > 0$ *be a polynomial. We say that an obfuscator* Obf *for* $\mathcal{C}$ *satisfies* t-composability*, if for every PPTA* $\mathcal{A}$ *and a positive polynomial* q*, there exists a computationally unbounded algorithm* $\mathcal{S}$ *that makes only polynomially many queries, such that for all sufficiently large* $k \in \mathbb{N}$ *and for all circuits* $C_1, \ldots, C_t \in \mathcal{C}_k$*, it holds that:*

$$|\Pr[\mathcal{A}(1^k, \mathsf{Obf}(C_1), \ldots, \mathsf{Obf}(C_t)) = 1] - \Pr[\mathcal{S}^{C_1,\ldots,C_t}(1^k) = 1]| \leq 1/q,$$

where the probabilities are over the randomness consumed by Obf*,* $\mathcal{A}$*, and* $\mathcal{S}$*.*

Notations for Point Obfuscators and MBPF Obfuscators. Let $\mathcal{X}$ be a finite set, $t \in \mathbb{N}$, $\alpha \in \mathcal{X}$, and $\beta \in \{0,1\}^t$. A *point function* $\mathcal{I}_\alpha$ and a *multi-bit point function* (MBPF) $\mathcal{I}_{\alpha\to\beta}$ are functions defined as follows:

$$\mathcal{I}_\alpha(x) = \begin{cases} \top & \text{if } x = \alpha \\ \bot & \text{otherwise} \end{cases} \quad \text{and} \quad \mathcal{I}_{\alpha\to\beta}(x) = \begin{cases} \beta & \text{if } x = \alpha \\ \bot & \text{otherwise} \end{cases}$$

We refer to α and β as the *point address* and the *point value*, respectively.

In this paper, we will only consider circuits for computing point functions/MBPFs with the properties that (1) the description is given in some canonical form and thus there is a one-to-one correspondence between a point address/value and the circuit for computing the point function/MBPF, and (2) the description of the circuits reveals the point address/value in the clear. Hereafter, we will identify a point function and an MBPF with circuits that compute them (with the above mentioned properties).

For an ensemble $\mathcal{X} = \{\mathcal{X}_k\}_{k\in\mathbb{N}}$, where each $\mathcal{X}_k$ is a set, we denote by $\mathsf{PF}(\mathcal{X})$ the ensemble of point functions $\{\mathcal{I}_\alpha\}_{\alpha\in\mathcal{X}_k}$. Similarly, for $\mathcal{X}$ and a polynomial t, we denote by $\mathsf{MBPF}(\mathcal{X}, t)$ the ensemble MBPFs $\{\mathcal{I}_{\alpha\to\beta}\}_{\alpha\in\mathcal{X}_k,\beta\in\{0,1\}^t}$.

Hereafter, we refer to an obfuscator for point functions as a *point obfuscator* and will denote it by PO. Furthermore, we refer to an obfuscator for MBPFs as an *MBPF obfuscator* and will denote it by MBPO. Moreover, we call an ensemble $\mathcal{X} = \{\mathcal{X}_k\}_{k\in\mathbb{N}}$ a *"domain ensemble" (for point functions and MBPFs)* if (1) for all $k \in \mathbb{N}$, each element of $\mathcal{X}_k$ is k-bit, (2) $|\mathcal{X}_k|$ is superpolynomially large in k (and thus $1/|\mathcal{X}_k|$ is negligible), and (3) we can efficiently sample an element from $\mathcal{X}_k$ uniformly at random.

Concrete Instantiations of a Composable Point Obfuscator and an MBPF Obfuscator. In Appendix B, we recall the concrete construction of a point obfuscator due to the results [21, 7], which is originally proposed by Canetti [21] as a perfectly one-way function and is later shown to be t-composable under the t-strong vector Diffie-Hellman (t-SVDDH) assumption, which is a stronger variant of the DDH assumption. There, we also recall the construction of an MBPF obfuscator based on a composable point obfuscator [23, 7].

3 New Security Definitions for MBPF Obfuscators

In this section, we introduce and formalize the new security notions for MBPF obfuscators that we call *average-case indistinguishability w.r.t. (computationally/statistically) partially uninvertible auxiliary input*, which will play a central role in our proposed KEMs given in Section 4. This security definition requires that obfuscated circuits of MBPFs hide the point values on average, even in the presence of "dependent" auxiliary inputs [43, 33], as long as the auxiliary input has some "hard-to-invert" property.

In the following, we formally define what we mean by "hard-to-invert" auxiliary input in Section 3.1. Then, in Section 3.2, we define the new indistinguishability-based notions. (Looking ahead, we will show the relations between the new security notions with the virtual black-/grey-box security notions in Section 5.)

For notational convenience, in this section, $\mathcal{X}$ will always denote a domain ensemble $\{\mathcal{X}_k\}_{k\in\mathbb{N}}$, and $t = t(k) > 0$ be a polynomial that will be used for MBPF obfuscators for $\mathsf{MBPF}(\mathcal{X}, t)$, and do not introduce them in each definition.

3.1 Auxiliary Input Functions and Partial Uninvertibility

For MBPF obfuscators, we will consider the average-case security in the presence of "dependent" auxiliary input [43] that depends on the description of an MBPF $\mathcal{I}_{\alpha\to\beta}$ being obfuscated. We will capture this by a probabilistic function ai that takes as input the point address/value pair $(\alpha, \beta) \in \mathcal{X}_k \times \{0,1\}^t$. Furthermore, we consider the (average-case) "partial uninvertibility" of the function ai. That is, given z output by $\mathsf{ai}(\alpha, \beta)$ for a randomly chosen (α, β), it is hard to find α. We consider computational and statistical partial uninvertibility.

Definition 5. *Let $\delta : \mathbb{N} \to [0, 1]$, and let $\mathsf{ai} : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^*$ be a (possibly probabilistic) two-input function. We say that ai is a* δ-computationally (resp. δ-statistically) partially uninvertible auxiliary input function (δ-$\mathtt{cPUAI}$ (resp. δ-$\mathtt{sPUAI}$) function, for short) *if (1) it is efficiently computable, and (2) for all PPTAs (resp. computationally unbounded algorithms) $\mathcal{F}$ and for all sufficiently large $k \in \mathbb{N}$, it holds that $\mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai},\mathcal{F}}(k) := \Pr[\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai},\mathcal{F}}(k) = 1] - 1/|\mathcal{X}_k| \le \delta(k)$,[3] where the experiment $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai},\mathcal{F}}(k)$ is defined as follows:*

$$\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai},\mathcal{F}}(k) : [\ \alpha \leftarrow \mathcal{X}_k;\ \beta \leftarrow \{0,1\}^t;\ z \leftarrow \mathsf{ai}(\alpha, \beta);\ \alpha' \leftarrow \mathcal{F}(1^k, z);\ \text{Return } (\alpha' \stackrel{?}{=} \alpha)\ ].$$

Furthermore, we say that ai is ℓ-bounded if the output length of ai is bounded by $\ell = \ell(k)$.

[3] Here, the subtraction of $1/|\mathcal{X}_k|$ is to offset the trivial success probability by a random guess.

3.2 Average-Case Indistinguishability of Point Values with Auxiliary Input

In our proposed KEM constructions, what we need for an MBPF obfuscator is that it hides the point value "on average," in the presence of auxiliary input that is *simultaneously* dependent on the point address and the point value. This indistinguishability-based definition, formalized below, enables us to avoid using simulator-based security notions, and helps to make the security analyses of our proposed constructions simpler.

Definition 6. *Let $\delta : \mathbb{N} \to [0,1]$. We say that an MBPF obfuscator* MBPO *satisfies* average-case indistinguishability w.r.t. δ-computationally (resp. δ-statistically) partially uninvertible auxiliary input *(* $\mathtt{AIND}$-δ-$\mathtt{cPUAI}$ *(resp.* $\mathtt{AIND}$-δ-$\mathtt{sPUAI}$*) secure, for short), if for all PPTAs $\mathcal{A}$ and all δ-*$\mathtt{cPUAI}$ *(resp. δ-*$\mathtt{sPUAI}$*) functions* ai*, $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the experiment $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k)$ is defined as follows:*

$$\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) : [\ \alpha \leftarrow \mathcal{X}_k;\ \beta_0, \beta_1 \leftarrow \{0,1\}^t;\ z \leftarrow \mathsf{ai}(\alpha, \beta_0);\ b \leftarrow \{0,1\};$$
$$\mathtt{DL} \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha \to \beta_b});\ b' \leftarrow \mathcal{A}(1^k, z, \mathtt{DL});\ \text{Return } (b' \stackrel{?}{=} b)\ ].$$

In the experiment, $\mathtt{DL}$ stands for a "digital locker" (the name is due to [23]).

The following is a simple fact that in order for the new definitions to be meaningful, δ has to be a negligible function. (The proof is given in Appendix D.1.)

Lemma 1. *Let $\delta : \mathbb{N} \to [0,1]$. If δ is non-negligible, then an MBPF obfuscator cannot be* $\mathtt{AIND}$-δ-$\mathtt{sPUAI}$ *secure (and hence it cannot be* $\mathtt{AIND}$-δ-$\mathtt{cPUAI}$ *secure, either).*

4 Chosen Ciphertext Security via MBPF Obfuscation

In this section, we show our main results: two constructions of $\mathtt{CCA2}$ secure KEMs. The first and the second constructions are given in Sections 4.1 and 4.2, respectively. We also explain several extensions applicable to our proposed constructions in Section 4.3.

4.1 First Construction

Let $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with the plaintext space $\{0,1\}^k$, the public key length $\ell_{\mathtt{PK}}(k)$, the randomness length $\ell_{\mathtt{R}}(k)$, and the ciphertext length $\ell_{\mathtt{C}}(k)$ (where the definitions of these are given in Appendix A. We define $t(k) = k \cdot \ell_{\mathtt{R}}(k) + k$ and $t'(k) = k \cdot \ell_{\mathtt{PK}}(k) + k \cdot \ell_{\mathtt{C}}(k) + k$. Let $\mathcal{X} = \{\mathcal{X}_k\}_{k \in \mathbb{N}}$ be a domain ensemble such that each element in $\mathcal{X}_k$ is of length k, and let MBPO be an MBPF obfuscator for $\mathsf{MBPF}(\mathcal{X}, t)$. Furthermore, let $\mathcal{H} = (\mathsf{HKG}, \mathsf{H})$ be a UOWHF. Then we construct a KEM $\Gamma = (\mathsf{KKG}, \mathsf{Encap}, \mathsf{Decap})$ as in Fig. 1.

Useful Properties of Γ. To show the $\mathtt{CCA2}$ security of the proposed KEM Γ, it is useful to note the following two simple properties, which are both due to the validity check of a ciphertext by re-encryption performed in the last step of Decap (and the correctness of the underlying PKE scheme Π). The first property states that in order to generate a valid ciphertext, an obfuscated circuit $\mathtt{DL}$ cannot be copied from other valid ciphertexts. (The formal proof is given in Appendix D.2.)

Lemma 2. *Let (PK, SK) be a key pair output by $\mathsf{KKG}(1^k)$, and $C = (c_1, \ldots, c_k, \mathtt{DL})$ be a ciphertext output by $\mathsf{Encap}(PK)$. Then, for any ciphertext $C' = (c'_1, \ldots, c'_k, \mathtt{DL}')$ satisfying $\mathtt{DL}' = \mathtt{DL}$ and $(c'_1, \ldots, c'_k) \neq (c_1, \ldots, c_k)$, it holds that $\mathsf{Decap}(SK, C') = \perp$.*

$\mathsf{KKG}(1^k)$:	$\mathsf{Encap}(PK)$:	$\mathsf{Decap}(SK, C)$:
$\kappa \leftarrow \mathsf{HKG}(1^k)$	Parse PK as $(\{pk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}, \kappa)$	Parse SK as $(\{sk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}, \kappa)$
$(pk_i^{(j)}, sk_i^{(j)}) \leftarrow \mathsf{PKG}(1^k)$ for $i \in [k]$ and $j \in \{0,1\}$	$\alpha \leftarrow \mathcal{X}_k$	Parse C as $(c_1, \ldots, c_k, \mathtt{DL})$
$PK \leftarrow (\{pk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}, \kappa)$	$\beta \leftarrow \{0,1\}^t$	$h \leftarrow \mathsf{H}_\kappa(\mathtt{DL})$
$SK \leftarrow (\{sk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}, \kappa)$	$\mathtt{DL} \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta})$	View h as $(h_1\| \ldots \|h_k) \in \{0,1\}^k$
Return (PK, SK)	$h \leftarrow \mathsf{H}_\kappa(\mathtt{DL})$	$\alpha \leftarrow \mathsf{Dec}(sk_1^{(h_1)}, c_1)$
	View h as $(h_1\| \ldots \|h_k) \in \{0,1\}^k$	If $\alpha = \perp$ then return $\perp$
	Parse β as $(r_1, \ldots, r_k, K) \in (\{0,1\}^{\ell_{\mathtt{R}}})^k \times \{0,1\}^k$	$\beta \leftarrow \mathtt{DL}(\alpha)$
	$c_i \leftarrow \mathsf{Enc}(pk_i^{(h_i)}, \alpha; r_i)$ for $i \in [k]$	If $\beta = \perp$ then return $\perp$
	$C \leftarrow (c_1, \ldots, c_k, \mathtt{DL})$	Parse β as $(r_1, \ldots, r_k, K) \in (\{0,1\}^{\ell_{\mathtt{R}}})^k \times \{0,1\}^k$
	Return (C, K)	If $\forall i \in [k] : \mathsf{Enc}(pk_i^{(h_i)}, \alpha; r_i) = c_i$ then return K else return $\perp$

Fig. 1. The proposed CCA2 secure KEM Γ.

The second property is the existence of the "alternative" decapsulation algorithm AltDecap. For a k-bit string $h^* = (h_1^*\| \ldots \|h_k^*) \in \{0,1\}^k$ and a key pair (PK, SK) output by $\mathsf{KKG}(1^k)$, where $SK = (\{sk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}, \kappa)$, we define the "alternative" secret key $\widehat{SK}_{h^*}$ associated with $h^* \in \{0,1\}^k$ by $\widehat{SK}_{h^*} = (h^*, PK, \{sk_i^{(1-h_i^*)}\}_{i\in[k]})$, where h_i^* is the i-th bit of h^*. AltDecap takes an "alternative" secret key $\widehat{SK}_{h^*}$ and a ciphertext $C = (c_1, \ldots, c_k, \mathtt{DL})$ as input, and runs as follows:

$\mathsf{AltDecap}(\widehat{SK}_{h^*}, C)$**:** First check if $\mathsf{H}_\kappa(\mathtt{DL}) = h^*$, and return $\perp$ if this is the case. Otherwise, let $h = \mathsf{H}_\kappa(\mathtt{DL})$ and let $\ell \in [k]$ be the smallest index such that $h_\ell = 1 - h_\ell^*$, where h_ℓ is the ℓ-th bit of h. (Note that such ℓ must exist because $h \neq h^*$ in this case.) Run in exactly the same way as $\mathsf{Decap}(SK, C)$, except that it executes $\mathsf{Dec}(sk_\ell^{(1-h_\ell^*)}, c_\ell)$ in the fifth step, instead of executing $\mathsf{Dec}(sk_1^{(h_1)}, c_1)$.

Regarding AltDecap, the following lemma is easy to see due to the symmetric role of each of $sk_i^{(j)}$ and the validity check of each c_i by re-encryption performed at the last step. (The formal proof is given in Appendix D.3.)

Lemma 3. *Let $h^* \in \{0,1\}^k$ be a string, (PK, SK) be a key pair output by $\mathsf{KKG}(1^k)$, and $\widehat{SK}_{h^*}$ be an alternative secret key corresponding to h^* and (PK, SK) as defined above. Then, for any ciphertext $C = (c_1, \ldots, c_k, \mathtt{DL})$ (which could be outside the range of $\mathsf{Encap}(PK)$) satisfying $\mathsf{H}_\kappa(\mathtt{DL}) \neq h^*$, it holds that $\mathsf{Decap}(SK, C) = \mathsf{AltDecap}(\widehat{SK}_{h^*}, C)$.*

CCA2 *Security of Γ.* The security of Γ is guaranteed by the following theorem.

Theorem 1. *Assume that Π is ϵ-CPA secure with negligible ϵ, $\mathcal{H}$ is a UOWHF, and* MBPO *is* AIND-δ-cPUAI *secure with $\delta(k) \geq k\epsilon(k)$. Then, the KEM Γ constructed as in Fig. 1 is* CCA2 *secure.*

Basic ideas for the proof of this theorem are explained in Section 1.3. Thus, we directly proceed to the proof.

Proof of Theorem 1. We will show that for any PPTA adversary $\mathcal{A}$ attacking the CCA2 security of the KEM Γ, there exist PPTAs $\mathcal{B}_h$ and $\mathcal{B}_o$ and a $(k\epsilon)$-cPUAI function $\mathsf{ai}_\Gamma : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^{t'}$ such that

$$\mathsf{Adv}^{\mathtt{CCA2}}_{\Gamma,\mathcal{A}}(k) \leq 2 \cdot \left(\frac{q}{|\mathcal{X}_k|} + \mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{B}_h}(k) + \mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_\Gamma,\mathcal{B}_o}(k) \right), \tag{1}$$

where Π^k is the k-repetition construction of the underlying PKE scheme Π (see Section C for the explanation on it). Combined with our assumptions on the building blocks and Lemma 7 (stated in Appendix C), this inequality implies that $\mathsf{Adv}^{\mathtt{CCA2}}_{\Gamma,\mathcal{A}}(k)$ is negligible, and proves the theorem.

Fix arbitrarily a $\mathtt{CCA2}$ adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against Γ that makes in total q decapsulation queries. (Since $\mathcal{A}$ is a PPTA, q is some polynomial.) 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 $\mathsf{Expt}^{\mathtt{CCA2}}_{\Gamma,\mathcal{A}}(k)$ itself. Without loss of generality, we generate the challenge ciphertext $C^* = (c^*_1, \ldots, c^*_k, \mathtt{DL}^*)$ and the challenge session-key K^*_b for $\mathcal{A}$, where b is the challenge bit for $\mathcal{A}$, before running $\mathcal{A}_1$. (Note that this does not affect $\mathcal{A}$'s behavior.)

Game 2: Same as Game 1, except that all decapsulation queries $C = (c_1, \ldots, c_k, \mathtt{DL})$ satisfying $\mathtt{DL} = \mathtt{DL}^*$ are answered with $\perp$.

Game 3: Same as Game 2, except that all decapsulation queries $C = (c_1, \ldots, c_k, \mathtt{DL})$ satisfying $\mathsf{H}_\kappa(\mathtt{DL}) = h^* = \mathsf{H}_\kappa(\mathtt{DL}^*)$ are answered with $\perp$.

Game 4: Same as Game 3, except that all decapsulation queries C are answered with $\mathsf{AltDecap}(\widehat{SK}_{h^*}, C)$, where $\widehat{SK}_{h^*}$ is the alternative secret key corresponding to (PK, SK) and $h^* = \mathsf{H}_\kappa(\mathtt{DL}^*) \in \{0,1\}^k$.

Game 5: Same as Game 4, except that $\mathtt{DL}^*$ is replaced with an obfuscation of the MBPF $\mathcal{I}_{\alpha^* \to \beta'}$ with an independently chosen random value $\beta' \in \{0,1\}^t$. That is, the step "$\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha^* \to \beta^*})$" in Game 4 is replaced with the steps "$\beta' \leftarrow \{0,1\}^t$; $\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha^* \to \beta'})$." (Note that each r^*_i and K^*_1 are still generated from β^*.)

For $i \in [5]$, let Succ_i be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game i. Using the above notation, $\mathcal{A}$'s $\mathtt{CCA2}$ advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{CCA2}}_{\Gamma,\mathcal{A}}(k) = 2 \cdot |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \leq 2 \cdot \sum_{i \in [4]} |\Pr[\mathsf{Succ}_i] - \Pr[\mathsf{Succ}_{i+1}]| + 2 \cdot |\Pr[\mathsf{Succ}_5] - \frac{1}{2}|. \quad (2)$$

In the following, we show the upperbounds of the terms that appear in the right hand side of the above inequality.

Claim 1 $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| \leq q/|\mathcal{X}_k|$.

Proof of Claim 1. For $i \in \{1,2\}$, let DLColl_i be the event that $\mathcal{A}$ submits at least one decapsulation query $C = (c_1, \ldots, c_k, \mathtt{DL})$ such that $\mathtt{DL} = \mathtt{DL}^*$ and $\mathsf{Decap}(SK, C) \neq \perp$. The difference between Game 1 and Game 2 is how $\mathcal{A}$'s decapsulation query $C = (c_1, \ldots, c_k, \mathtt{DL})$ satisfying $\mathtt{DL} = \mathtt{DL}^*$ is answered, and these games proceed identically unless DLColl_1 or DLColl_2 occurs in the corresponding games. Hence, we have

$$|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| \leq \Pr[\mathsf{DLColl}_1] = \Pr[\mathsf{DLColl}_2]. \quad (3)$$

Thus, it is sufficient to show the upperbound of $\Pr[\mathsf{DLColl}_1]$. Moreover, recall that according to the rule of the $\mathtt{CCA2}$ experiment, $\mathcal{A}_2$'s queries $C = (c_1, \ldots, c_k, \mathtt{DL})$ must satisfy $C \neq C^*$, and thus if $\mathtt{DL} = \mathtt{DL}^*$, then it must be the case that $(c_1, \ldots, c_k) \neq (c^*_1, \ldots, c^*_k)$. Then, by Lemma 2, for any decapsulation query C by $\mathcal{A}_2$ satisfying $\mathtt{DL} = \mathtt{DL}^*$, we must have $\mathsf{Decap}(SK, C) = \perp$. This means that $\mathcal{A}_2$'s queries are answered identically in both Game 1 and Game 2.

Therefore, to show the upperbound of $\Pr[\mathsf{DLColl}_1]$, it is sufficient to show the upperbound of the probability that $\mathcal{A}_1$ makes a query that causes the event DLColl_1. Instead of directly considering the event, we show the upperbound of the probability that $\mathcal{A}_1$ makes a decapsulation query that contains $\mathtt{DL}^*$. (Clearly, if $\mathcal{A}_1$ does not make such a query, then no query made by $\mathcal{A}_1$ causes DLColl_1.) However, it is easy to see that in Game 1, the probability that $\mathcal{A}_1$ makes a decapsulation query that contains $\mathtt{DL}^*$ is at most $q/|\mathcal{X}_k|$. (This holds even if $\mathcal{A}_1$ is computationally unbounded.) This is because in order for $\mathcal{A}_1$ to make such a query, it is at least necessary that $\mathcal{A}_1$ succeeds in guessing the randomly chosen point value $\alpha^* \in \mathcal{X}_k$, without seeing any information on α^*. (Note that if $\mathtt{DL}^*$

is an obfuscation of the MBPF $\mathcal{I}_{\alpha^*\to\beta^*}$ for some α^* and β^*, then $\mathtt{DL}^*$ cannot be an obfuscation of another MBPF $\mathcal{I}_{\alpha\to\beta}$ with $\alpha \neq \alpha^*$.) Therefore, the probability (over the choice of $\alpha^* \in \mathcal{X}_k$ and the choice of $\mathcal{A}$'s internal randomness) that $\mathtt{DL}^*$ is contained in one particular decapsulation query made by $\mathcal{A}_1$ is exactly $1/|\mathcal{X}_k|$. By the union bound over all of $\mathcal{A}_1$'s possible q queries, the probability that $\mathcal{A}_1$ makes a decapsulation query containing $\mathtt{DL}^*$ is at most $q/|\mathcal{X}_k|$.

In summary, we have seen that $\Pr[\mathsf{DLColl}_1]$ is upperbounded by $q/|\mathcal{X}_k|$. Then, by the inequality (3), we have $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| \leq q/|\mathcal{X}_k|$. This completes the proof of Claim 1. □

Claim 2 *There exists a PPTA $\mathcal{B}_h$ such that* $\mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{B}_h}(k) \geq |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$.

Proof of Claim 3. For $i \in \{2,3\}$, let HColl_i be the event that $\mathcal{A}$ submits at least one decapsulation query $C = (c_1, \ldots, c_k, \mathtt{DL})$ satisfying $\mathsf{H}_\kappa(\mathtt{DL}) = h^* = \mathsf{H}_\kappa(\mathtt{DL}^*)$, $\mathtt{DL} \neq \mathtt{DL}^*$, and $\mathsf{Decap}(SK, C) \neq \perp$. Note that the difference between Game 2 and Game 3 is how $\mathcal{A}$'s decapsulation query $C = (c_1, \ldots, c_k, \mathtt{DL})$ satisfying $\mathsf{H}_\kappa(\mathtt{DL}) = \mathsf{H}_\kappa(\mathtt{DL}^*)$ and $\mathtt{DL} \neq \mathtt{DL}^*$ are answered. (Note that the queries with $\mathtt{DL} = \mathtt{DL}^*$ are answered with $\perp$ in both games.) These games proceed identically unless HColl_2 or HColl_3 occurs in the corresponding games, and hence we have

$$|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]| \leq \Pr[\mathsf{HColl}_2] = \Pr[\mathsf{HColl}_3]. \tag{4}$$

We show how to construct a PPTA adversary $\mathcal{B}_h$ that attacks the universal one-wayness of $\mathcal{H}$ with the advantage $\mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{B}_h}(k) \geq \Pr[\mathsf{HColl}_3]$. The description of $\mathcal{B}_h = (\mathcal{B}_{h1}, \mathcal{B}_{h2})$ is as follows:

$\mathcal{B}_{h1}(1^k)$**:** $\mathcal{B}_{h1}$ picks $\alpha^* \in \mathcal{X}_k$ and $\beta^* = (r_1^* \| \ldots \| r_k^* \| K_1^*) \in \{0,1\}^t$ uniformly at random, and computes $\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha^*\to\beta^*})$. Then $\mathcal{B}_{h1}$ prepares the state information $\mathsf{st}_{\mathcal{B}}$ consisting of all information known to $\mathcal{B}_{h1}$, and terminates with output $(\mathtt{DL}^*, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{h2}(\mathsf{st}_{\mathcal{B}}, \kappa)$**:** $\mathcal{B}_{h2}$ generates a key pair (PK, SK) in the same way as $\mathsf{KKG}(1^k)$ does, except that $\mathcal{B}_{h2}$ uses the hash-key κ that it receives as a hash-key in (PK, SK). $\mathcal{B}_{h2}$ next runs $h^* = (h_1^* \| \ldots \| h_k^*) \leftarrow \mathsf{H}_\kappa(\mathtt{DL}^*)$ and $c_i^* \leftarrow \mathsf{Enc}(pk_i^{(h_i^*)}, \alpha^*; r_i^*)$ for all $i \in [k]$, sets $C^* \leftarrow (c_1^*, \ldots, c_k^*, \mathtt{DL}^*)$, and also chooses $b \in \{0,1\}$ and $K_0^* \in \{0,1\}^k$ uniformly at random. Then, $\mathcal{B}_{h2}$ runs $\mathcal{A}_1$ and $\mathcal{A}_2$ as Game 3 runs (which is possible because $\mathcal{B}_{h2}$ holds SK). When $\mathcal{A}_2$ terminates, $\mathcal{B}_{h2}$ checks if $\mathcal{A}_1$ or $\mathcal{A}_2$ made a decapsulation query $C = (c_1, \ldots, c_k, \mathtt{DL})$ satisfying $\mathsf{H}_\kappa(\mathtt{DL}) = h^*$ and $\mathtt{DL} \neq \mathtt{DL}^*$. If such query is found, then $\mathcal{B}_{h2}$ terminates with output $\mathtt{DL}$. Otherwise, $\mathcal{B}_{h2}$ simply gives up and aborts.

The above completes the description of $\mathcal{B}_h$. It is easy to see that $\mathcal{B}_h$ does perfect simulation of Game 3 for $\mathcal{A}$, and whenever $\mathcal{A}$ makes a query that causes the event HColl_3, $\mathcal{B}_{h2}$ can find such a query and output a colliding value $\mathtt{DL}$ satisfying $\mathsf{H}_\kappa(\mathtt{DL}) = \mathsf{H}_\kappa(\mathtt{DL}^*)$ and $\mathtt{DL} \neq \mathtt{DL}^*$. Therefore, we have $\mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{B}_h}(k) \geq \Pr[\mathsf{HColl}_3]$. Then, by the inequality (4), we have $\mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{B}_h}(k) \geq |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$. This completes the proof of Claim 2. □

Claim 3 $\Pr[\mathsf{Succ}_3] = \Pr[\mathsf{Succ}_4]$.

Proof of Claim 3. It is sufficient to show that the behavior of the oracle given to $\mathcal{A}$ in Game 3 and that in Game 4 are identical. Let $C = (c_1, \ldots, c_k, \mathtt{DL})$ be a decapsulation query that $\mathcal{A}$ makes. If $\mathsf{H}_\kappa(\mathtt{DL}) = h^* = \mathsf{H}_\kappa(\mathtt{DL}^*)$, then the query is answered with $\perp$ in Game 3 by definition, while the oracle $\mathsf{AltDecap}(\widehat{SK}_{h^*}, C)$ that is given access to $\mathcal{A}$ in Game 4 also returns $\perp$ by definition. Otherwise (i.e. $\mathsf{H}_\kappa(\mathtt{DL}) \neq h^*$), by Lemma 3, the result of $\mathsf{Decap}(SK, C)$ and that of $\mathsf{AltDecap}(\widehat{SK}_{h^*}, C)$ agree. This completes the proof of Claim 3. □

Next, we would like to show the upperbound of $|\Pr[\mathsf{Succ}_4] - \Pr[\mathsf{Succ}_5]|$. To this end, we need to use the $\mathtt{AIND}$-δ-$\mathtt{cPUAI}$ security of the MBPF obfuscator MBPO. We therefore first specify the auxiliary input function that we are going to consider. Define the probabilistic function $\mathsf{ai}_\Gamma : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^{t'}$ that takes $(\alpha, \beta) \in \mathcal{X}_k \times \{0,1\}^t$ as input, and computes $z = (\{pk_i\}_{i\in[k]}, c_1^*, \ldots, c_k^*, K^*) \in \{0,1\}^{t'}$ in the following way:

$$\mathsf{ai}_\Gamma(\alpha,\beta) : [\ (pk_i, sk_i) \leftarrow \mathsf{PKG}(1^k) \text{ for } i \in [k];\ \text{Parse } \beta \text{ as } (r_1^*, \ldots, r_k^*, K^*) \in (\{0,1\}^{\ell_{\mathtt{R}}})^k \times \{0,1\}^k;$$
$$c_i^* \leftarrow \mathsf{Enc}(pk_i, \alpha; r_i^*) \text{ for } i \in [k];\ \text{Return } z \leftarrow (\{pk_i\}_{i\in[k]}, c_1^*, \ldots, c_k^*, K^*)\ ],$$

where the randomness used by ai_Γ is the randomness for executing PKG for k times. Note that ai_Γ is efficiently computable. The following claim guarantees that ai_Γ is computationally partially uninvertible.

Claim 4 *ai_Γ is a $(k\epsilon)$-$\mathtt{cPUAI}$ function.*

Proof of Claim 4. As noted above, ai_Γ is efficiently computable. We will show that for any PPTA $\mathcal{F}$ which runs in the experiment $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k)$, there exists a PPTA adversary $\mathcal{B}_p$ that attacks the k-repetition construction Π^k with the advantage $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi^k,\mathcal{B}_p}(k) = \mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k)$. Once this is shown, by using the ϵ-$\mathtt{CPA}$ security of Π and Lemma 7 we have that for any PPTA $\mathcal{F}$ and for all sufficiently large $k \in \mathbb{N}$, it holds that $\mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k) \le k\epsilon$, which implies that ai_Γ is a $(k\epsilon)$-$\mathtt{cPUAI}$ function, as claimed.

To show the above, fix an arbitrary PPTA $\mathcal{F}$ that runs in $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k)$. The description of the $\mathtt{CPA}$ adversary $\mathcal{B}_p = (\mathcal{B}_{p1}, \mathcal{B}_{p2})$ against the k-repetition construction Π^k is as follows:

$\mathcal{B}_{p1}(PK' = (pk_1, \ldots, pk_k))$**:** $\mathcal{B}_{p1}$ picks $\alpha \in \mathcal{X}_k$ uniformly at random, and sets $M_0 \leftarrow \alpha$ and $M_1 \leftarrow 0^k$. Then $\mathcal{B}_{p1}$ prepares the state information $\mathsf{st}_\mathcal{B}$ consisting of all information known to $\mathcal{B}_{p1}$, and terminates with output $(M_0, M_1, \mathsf{st}_\mathcal{B})$.

$\mathcal{B}_{p2}(\mathsf{st}_\mathcal{B}, C'^* = (c_1^*, \ldots, c_k^*))$**:** $\mathcal{B}_{p2}$ picks $K^* \in \{0,1\}^k$ uniformly, sets $z \leftarrow (\{pk_i\}_{i\in[k]}, c_1^*, \ldots, c_k^*, K^*)$, runs $\alpha' \leftarrow \mathcal{F}(1^k, z)$, and terminates with output $b' \leftarrow (\alpha' \stackrel{?}{=} \alpha)$.

The above completes the description of $\mathcal{B}_p$. Let b be the challenge bit for $\mathcal{B}_p$. $\mathcal{B}_p$'s $\mathtt{CPA}$ advantage can be estimated as follows:

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

Consider the case when $b = 0$. It is easy to see that in this case, $\mathcal{B}_p$ perfectly simulates $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k)$ for $\mathcal{F}$. In particular, each c_i^* is an encryption of $M_0 = \alpha$ for a uniformly chosen α, and the randomness r_i^* used for generating c_i^* is also chosen uniformly by $\mathcal{B}_p$'s experiment (recall that the experiment $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k)$ chooses $\beta = (r_1^* \| \ldots \| r_k^* \| K^*) \in \{0,1\}^t$ uniformly at random), as is done in $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k)$. Under this situation, the probability that $\alpha' = \alpha$ occurs is exactly the same as the probability that $\mathcal{F}$ outputs α in $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k)$, i.e., $\Pr[\alpha' = \alpha|b = 0] = \mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k) + 1/|\mathcal{X}_k|$.

When $b = 1$, on the other hand, each c_i^* in z is an encryption of $M_1 = 0^k$, and thus z is completely independent of α. Therefore, α is information-theoretically hidden from $\mathcal{F}$. This must mean that in this case, the probability of $\mathcal{F}$ outputting α is exactly $1/|\mathcal{X}_k|$. That is, $\Pr[\alpha' = \alpha|b = 1] = 1/|\mathcal{X}_k|$. (This holds even if $\mathcal{F}$ is computationally unbounded.)

In summary, we have $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi^k,\mathcal{B}_p}(k) = \mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_\Gamma,\mathcal{F}}(k)$. Since the choice of $\mathcal{F}$ was arbitrarily, the above works for any PPTA $\mathcal{F}$. Hence, ai_Γ is $(k\epsilon)$-computationally partially uninvertible. This completes the proof of Claim 4. □

Now, we turn to showing the upperbound of $|\Pr[\mathsf{Succ}_4] - \Pr[\mathsf{Succ}_5]|$.

Claim 5 *There exists a PPTA $\mathcal{B}_o$ such that* $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_\Gamma,\mathcal{B}_o}(k) = |\Pr[\mathsf{Succ}_4] - \Pr[\mathsf{Succ}_5]|$.

Proof of Claim 5. We show how to construct a PPTA adversary $\mathcal{B}_o$ with the claimed advantage. $\mathcal{B}_o$ is given as input 1^k, $z \leftarrow \mathsf{ai}_\Gamma(\alpha, \beta_0)$, and $\mathtt{DL}^*$ which is output from either $\mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_0})$ or $\mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_1})$ (where $\alpha \in \mathcal{X}_k$ and $\beta_0, \beta_1 \in \{0,1\}^t$ are chosen uniformly at random), and runs as follows:

$\mathcal{B}_o(1^k, z, \mathtt{DL}^*)$**:** $\mathcal{B}_o$ first parses z as $(\{pk_i\}_{i\in[k]}, c_1^*, \ldots, c_k^*, K^*)$, and runs $\kappa \leftarrow \mathsf{HKG}(1^k)$ and $h^* \leftarrow \mathsf{H}_\kappa(\mathtt{DL}^*)$. Let $h^* = (h_1^* \| \ldots \| h_k^*) \in \{0,1\}^k$. For each $i \in [k]$, $\mathcal{B}_o$ sets $pk_i^{(h_i^*)} \leftarrow pk_i$ and runs $(pk_i^{(1-h_i^*)}, sk_i^{(1-h_i^*)}) \leftarrow \mathsf{PKG}(1^k)$. $\mathcal{B}_o$ then picks $\gamma \in \{0,1\}$ and $K_0^* \in \{0,1\}^k$ uniformly, and sets $PK \leftarrow (\{pk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}, \kappa)$, $\widehat{SK}_{h^*} \leftarrow (h^*, PK, \{sk_i^{(1-h_i^*)}\}_{i\in[k]})$, $C^* \leftarrow (c_1^*, \ldots, c_k^*, \mathtt{DL}^*)$, and $K_1^* \leftarrow K^*$. Then, $\mathcal{B}_o$ runs $\mathsf{st} \leftarrow \mathcal{A}_1^{\mathsf{AltDecap}(\widehat{SK}_{h^*},\cdot)}(PK)$ and $\gamma' \leftarrow \mathcal{A}_2^{\mathsf{AltDecap}(\widehat{SK}_{h^*},\cdot)}(\mathsf{st}, C^*, K_\gamma^*)$, and terminates with output $b' \leftarrow (\gamma' \stackrel{?}{=} \gamma)$.

The above completes the description of $\mathcal{B}_o$. Let b be the challenge bit for $\mathcal{B}_o$. $\mathcal{B}_o$'s $\mathtt{AIND\text{-}AI}$ advantage is estimate as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_\Gamma,\mathcal{B}_o}(k) &= 2 \cdot |\Pr[b' = b] - \frac{1}{2}| = |\Pr[b' = 1 | b = 0] - \Pr[b' = 1 | b = 1]| \\ &= |\Pr[\gamma' = \gamma | b = 0] - \Pr[\gamma' = \gamma | b = 1]|.\end{aligned}$$

Consider the case when $b = 0$ (i.e. $\mathtt{DL}^*$ is computed as $\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_0})$). Note that by the definition of the experiment $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_\Gamma,\mathcal{B}_o}(k)$, if we regard α and β_0 in $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_\Gamma,\mathcal{B}_o}(k)$ as α^* and β^* in Game 4, respectively, then the values in z (i.e. $\{pk_i\}_{i\in[k]}$ which are used as $\{pk_i^{(h_i^*)}\}_{i\in[k]}$, the ciphertexts $\{c_i^*\}_{i\in[k]}$, and the value K^* which is used as K_1^*), are generated/chosen in exactly same way as those in Game 4. Therefore, since γ is chosen randomly by $\mathcal{B}_o$, the challenge ciphertext $C^* = (c_1^*, \ldots, c_k^*, \mathtt{DL}^*)$ and the challenge session-key K_γ^* for $\mathcal{A}$ is distributed identically to those in Game 4 in which the challenge bit for $\mathcal{A}$ is γ. Moreover, decapsulation queries from $\mathcal{A}$ are answered by using $\mathsf{AltDecap}(\widehat{SK}_{h^*}, \cdot)$, as is done in Game 4. Hence, $\mathcal{B}_o$ simulates Game 4 perfectly for $\mathcal{A}$ in which the challenge bit for $\mathcal{A}$ is γ. Under this situation, the probability that $\gamma' = \gamma$ occurs is exactly the same as the probability that $\mathcal{A}$ succeeds in guessing the challenge bit in Game 4, i.e. $\Pr[\gamma' = \gamma | b = 0] = \Pr[\mathsf{Succ}_4]$.

Next, consider the case when $b = 1$. In this case, $\mathtt{DL}^*$ is computed as $\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_1})$, where β_1 is also chosen uniformly at random from $\{0,1\}^t$, independently of β_0. Under this situation, if we regard α, β_0, and β_1 in $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_\Gamma,\mathcal{B}_o}(k)$ as α^*, β^*, and β' in Game 5, respectively, then $\mathcal{A}$'s challenge ciphertext/session-key pair (C^*, K_γ^*) is generated in such a way that it is distributed identically to that in Game 5 in which the challenge bit for $\mathcal{A}$ is γ, and thus $\mathcal{B}_o$ simulates Game 5 perfectly for $\mathcal{A}$ in which the challenge bit is γ. Therefore, with a similar argument to the above, we have $\Pr[\gamma' = \gamma | b = 1] = \Pr[\mathsf{Succ}_5]$.

In summary, we have $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_\Gamma,\mathcal{B}_o}(k) = |\Pr[\mathsf{Succ}_4] - \Pr[\mathsf{Succ}_5]|$. This completes the proof of Claim 5. □

Claim 6 $\Pr[\mathsf{Succ}_5] = 1/2$.

Proof of Claim 6. This is obvious because K_1^*, which is contained in β^*, is independent of the challenge ciphertext C^*, and the distribution of K_1^* and that of K_0^* are exactly the same in Game 5. Therefore, the distribution of the challenge ciphertext/session-key pair (C^*, K_b^*) as well as all other values (public key PK and the responses to decapsulation queries) are identically distributed

in both cases $b = 0$ and $b = 1$. This must mean that the probability that $\mathcal{A}$ succeeds in guessing the challenge bit is exactly $1/2$. This completes the proof of Claim 6. □

Claims 1 to 6 and the inequality (2) guarantee that there exist PPTAs $\mathcal{B}_h$ and $\mathcal{B}_o$, and a $(k\epsilon)$-$\mathtt{cPUAI}$ function ai_Γ satisfying the inequality (1), as required. Recall that the choice of the PPTA $\mathtt{CCA2}$ adversary $\mathcal{A}$ was arbitrarily, and thus for any PPTA $\mathtt{CCA2}$ adversary we can show its negligible advantage. Hence, Γ is $\mathtt{CCA2}$ secure. This completes the proof of Theorem 1. □

4.2 Second Construction

In the first construction shown in the previous subsection, we used an ordinary $\mathtt{CPA}$ secure PKE scheme for Π. Now, we consider the construction of the KEM Γ in which Π is replaced with a lossy encryption scheme. Π now has the lossy key generation algorithm LKG, and thus is of the form $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{LKG})$. (The lossy key generation algorithm LKG is actually not used in the construction, and is used only in the security proof.) Because of this change, we can now relax the requirement for the MBPF obfuscator MBPO to be secure in the presence of only statistically partially uninvertible auxiliary input. This result is captured by the following theorem.

Theorem 2. *Assume Π is an ϵ-lossy encryption scheme with negligible ϵ, $\mathcal{H}$ is a UOWHF, and MBPO is $\mathtt{AIND}$-δ-$\mathtt{sPUAI}$ secure with $\delta(k) \geq k\epsilon(k)$. Then, the KEM Γ constructed as in Fig. 1 is $\mathtt{CCA2}$ secure.*

The proof proceeds very similarly to that of Theorem 1, and the main difference is that we introduce an additional game between Game 4 and Game 5 (in the proof of Theorem 1) for switching the public keys $\{pk_i^{(h_i^*)}\}_{i\in[k]}$ (corresponding to $\mathcal{A}$'s challenge ciphertext C^*) into lossy public keys.

Proof of Theorem 2. We will show that for any PPTA adversary $\mathcal{A}$ attacking the $\mathtt{CCA2}$ security of the KEM Γ, there exist PPTAs $\mathcal{B}_h$, $\mathcal{B}_\ell$, and $\mathcal{B}_o$, and a $(k\epsilon)$-$\mathtt{sPUAI}$ function $\mathsf{ai}'_\Gamma : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^{t'}$ such that

$$\mathsf{Adv}^{\mathtt{CCA2}}_{\Gamma,\mathcal{A}}(k) \leq 2 \cdot \left(\frac{q}{|\mathcal{X}_k|} + \mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{B}_h}(k) + \mathsf{Adv}^{\mathtt{KEY}}_{\Pi^k,\mathcal{B}_\ell}(k) + \mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}'_\Gamma,\mathcal{B}_o}(k) \right), \tag{5}$$

where Π^k is the k-repetition construction of the underlying lossy encryption scheme Π (see Section C). Combined with our assumptions on the building blocks and Lemma 8 (stated in Appendix C), this inequality implies that $\mathsf{Adv}^{\mathtt{CCA2}}_{\Gamma,\mathcal{A}}(k)$ is negligible, which proves the theorem.

Fix arbitrarily a $\mathtt{CCA2}$ adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against Γ that make in total q decapsulation queries. (Since $\mathcal{A}$ is a PPTA, q is some polynomial.) Consider the following sequence of games: (Here, the values with asterisk (*) represent those related to the challenge ciphertext for $\mathcal{A}$.)

Games 1, 2, 3, and 4: These games are exactly the same as those in the proof of Theorem 1.

Game 5: Same as Game 4, except that each of $pk_i^{(h_i^*)}$ is generated by the lossy key generation algorithm LKG, where h_i^* is the i-th bit of $h^* = \mathsf{H}_\kappa(\mathtt{DL}^*)$. Note that the corresponding secret keys $\{sk_i^{(h_i^*)}\}_{i\in[k]}$ are not at all used in Game 5 (in fact they are already not required in Game 4), and thus this game is well-defined.

Game 6: Same as Game 5, except that $\mathtt{DL}^*$ is replaced with an obfuscation of the MBPF $\mathcal{I}_{\alpha^*\to\beta'}$ with an independently chosen random value $\beta' \in \{0,1\}^t$. That is, the step "$\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha^*\to\beta^*})$" is replaced with the steps "$\beta' \leftarrow \{0,1\}^t$; $\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha^*\to\beta'})$." (Note that each r_i^* and K_1^* are still generated from β^*.)

For $i \in [6]$, let Succ_i be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game i. Using the above notation, $\mathcal{A}$'s CCA2 advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{CCA2}}_{\Gamma,\mathcal{A}}(k) = 2 \cdot |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \leq 2 \cdot \sum_{i \in [5]} |\Pr[\mathsf{Succ}_i] - \Pr[\mathsf{Succ}_{i+1}| + 2 \cdot |\Pr[\mathsf{Succ}_6] - \frac{1}{2}| \quad (6)$$

In the following, we show the upperbounds of the terms that appear in the right hand side of the above inequality.

Claim 7 $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| \leq q/|\mathcal{X}_k|$.

Claim 8 *There exists a PPTA* $\mathcal{B}_h$ *such that* $\mathsf{Adv}^{\mathtt{UOW}}_{\mathcal{H},\mathcal{B}_h} \geq |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$.

Claim 9 $\Pr[\mathsf{Succ}_3] = \Pr[\mathsf{Succ}_4]$.

The proofs of these claims are identical to those of Claims 1, 2, and 3 in the proof of Theorem 1, respectively, and thus omitted.

Claim 10 *There exists a PPTA* $\mathcal{B}_\ell$ *such that* $\mathsf{Adv}^{\mathtt{KEY}}_{\Pi^k,\mathcal{B}_\ell}(k) = |\Pr[\mathsf{Succ}_4] - \Pr[\mathsf{Succ}_5]|$.

Proof of Claim 10. We show how to construct a PPTA distinguisher $\mathcal{B}_\ell$ that has the claimed advantage in distinguishing ordinary/lossy keys of the k-repetition lossy encryption scheme Π^k. $\mathcal{B}_\ell$ is given as input PK' which is output from either $\mathsf{PKG}^k(1^k)$ or $\mathsf{LKG}^k(1^k)$, and runs as follows:

$\mathcal{B}_\ell(PK' = (pk_1, \ldots, pk_k))$**:** $\mathcal{B}_\ell$ first picks $\alpha^* \in \{0,1\}^k$ and $\beta^* = (r_1^* \| \ldots \| r_k^* \| K_1^*) \in \{0,1\}^t$ uniformly at random, and runs $\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha^* \to \beta^*})$, $\kappa \leftarrow \mathsf{HKG}(1^k)$, and $h^* \leftarrow \mathsf{H}_\kappa(\mathtt{DL}^*)$. Let $h^* = (h_1^* \| \ldots \| h_k^*) \in \{0,1\}^k$. For each $i \in [k]$, $\mathcal{B}_\ell$ sets $pk_i^{(h_i^*)} \leftarrow pk_i$ and runs $(pk_i^{(1-h_i^*)}, sk_i^{(1-h_i^*)}) \leftarrow \mathsf{PKG}(1^k)$. $\mathcal{B}_\ell$ next picks $K_0^* \in \{0,1\}^k$ and $\gamma \in \{0,1\}$ uniformly at random, and then runs $c_i^* \leftarrow \mathsf{Enc}(pk_i^{(h_i^*)}, \alpha^*; r_i^*)$ for every $i \in [k]$. Then, $\mathcal{B}_\ell$ sets $PK \leftarrow (\{pk_i^{(j)}\}_{i \in [k], j \in \{0,1\}}, \kappa)$, $\widehat{SK}_{h^*} \leftarrow (h^*, PK, \{sk_i^{(1-h_i^*)}\}_{i \in [k]})$, and $C^* \leftarrow (c_1^*, \ldots, c_k^*, \mathtt{DL}^*)$, and runs $\mathsf{st} \leftarrow \mathcal{A}_1^{\mathsf{AltDecap}(\widehat{SK}_{h^*}, \cdot)}(PK)$ and $\gamma' \leftarrow \mathcal{A}_2^{\mathsf{AltDecap}(\widehat{SK}_{h^*}, \cdot)}(\mathsf{st}, C^*, K_\gamma^*)$. Finally, $\mathcal{B}_\ell$ terminates with output $b' \leftarrow (\gamma' \stackrel{?}{=} \gamma)$.

The above completes the description of $\mathcal{B}_\ell$. Let b be the challenge bit for $\mathcal{B}_\ell$. $\mathcal{B}_\ell$'s advantage in distinguishing ordinary/lossy keys can be estimated as follows:

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

Consider the case when $b = 0$ (i.e. PK' given to $\mathcal{B}_\ell$ is generated from PKG^k). Then it is easy to see that $\mathcal{B}_\ell$ perfectly simulates Game 4 for $\mathcal{A}$ in which the challenge bit for $\mathcal{A}$ is γ. Under the situation, the probability that $\gamma' = \gamma$ occurs is exactly the same as the probability that $\mathcal{A}_2$ succeeds in guessing its challenge bit in Game 4, i.e. $\Pr[\gamma' = \gamma | b = 0] = \Pr[\mathsf{Succ}_4]$.

When $b = 1$ (i.e. PK' given to $\mathcal{B}_\ell$ is generated from LKG^k), on the other hand, it is also easy to see that $\mathcal{B}_\ell$ perfectly simulates Game 5 for $\mathcal{A}$ in which the challenge bit for $\mathcal{A}$ is γ. With a similar argument to the above, we have $\Pr[\gamma' = \gamma | b = 1] = \Pr[\mathsf{Succ}_5]$.

In summary, we have $\mathsf{Adv}^{\mathtt{KEY}}_{\Pi^k,\mathcal{B}_\ell}(k) = |\Pr[\mathsf{Succ}_4] - \Pr[\mathsf{Succ}_5]|$. This completes the proof of Claim 10. □

Next, we would like to show the upperbound of $|\Pr[\mathsf{Succ}_5] - \Pr[\mathsf{Succ}_6]|$. To this end, we need to use the $\mathtt{AIND}$-δ-$\mathtt{sPUAI}$ security of the MBPF obfuscator MBPO. We therefore first specify the auxiliary input function that we are going to consider. Define a probabilistic function $\mathsf{ai}'_\Gamma : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^{t'}$ that takes $(\alpha, \beta) \in \mathcal{X}_k \times \{0,1\}^t$ as input, and computes $z = (\{pk_i\}_{i\in[k]}, c_1^*, \ldots, c_k^*, K^*) \in \{0,1\}^{t'}$ in the following way:

$$\mathsf{ai}'_\Gamma(\alpha,\beta) : [\ pk_i \leftarrow \mathsf{LKG}(1^k) \text{ for } i \in [k];\ \text{Parse } \beta \text{ as } (r_1^*, \ldots, r_k^*, K^*) \in (\{0,1\}^{\ell_{\mathtt{R}}})^k \times \{0,1\}^k;$$
$$c_i^* \leftarrow \mathsf{Enc}(pk_i, \alpha; r_i^*) \text{ for } i \in [k];\ \text{Return } z \leftarrow (\{pk_i\}_{i\in[k]}, c_1^*, \ldots, c_k^*, K^*)],$$

where the randomness used by ai'_Γ is the randomness for executing LKG for k times. Note that ai'_Γ is efficiently computable. The following claim guarantees that ai'_Γ is statistically partially uninvertible.

Claim 11 *ai'_Γ is a $(k\epsilon)$-$\mathtt{sPUAI}$ function.*

Proof of Claim 11. This claim is shown in essentially the same way as the proof of Claim 4, by considering not the ordinary $\mathtt{CPA}$ experiment but the $\mathtt{LOS}$-$\mathtt{CPA}$ experiment regarding the k-repetition lossy encryption scheme Π^k. Namely, we can show that for any computationally unbounded algorithm $\mathcal{F}$ that runs in the experiment $\mathsf{Expt}^{\mathtt{P}\text{-}\mathtt{Inv}}_{\mathsf{ai}'_\Gamma,\mathcal{F}}(k)$, there exists a computationally unbounded algorithm $\mathcal{B}_p$ such that $\mathsf{Adv}^{\mathtt{LOS}\text{-}\mathtt{CPA}}_{\Pi^k,\mathcal{B}_p}(k) = \mathsf{Adv}^{\mathtt{P}\text{-}\mathtt{Inv}}_{\mathsf{ai}'_\Gamma,\mathcal{F}}(k)$. (The description of $\mathcal{B}_p$ is exactly the same as that of $\mathcal{B}_p$ that we used in the proof of Claim 4.) Here, due to Lemma 8 and the assumption that Π is an ϵ-lossy encryption scheme, we have that $\mathsf{Adv}^{\mathtt{LOS}\text{-}\mathtt{CPA}}_{\Pi^k,\mathcal{B}_p}(k) \le k\epsilon(k)$ for all sufficiently large $k \in \mathbb{N}$. Therefore, for any computationally unbounded algorithm $\mathcal{F}$ and for all sufficiently large $k \in \mathbb{N}$, we have $\mathsf{Adv}^{\mathtt{P}\text{-}\mathtt{Inv}}_{\mathsf{ai}'_\Gamma,\mathcal{F}}(k) \le k\epsilon(k)$. That is, ai'_Γ is $(k\epsilon)$-statistically partially uninvertible. This completes the proof of Claim 11. □

The rest of the proof proceeds almost identically to that proof of Theorem 1. More specifically, the following claims can be shown in exactly the same way as Claims 5 and 6, respectively, and thus we omit the proofs. (The only difference is that the reduction algorithm $\mathcal{B}_o$ uses LKG, instead of PKG, to generate $\{pk_i^{(h_i^*)}\}_{i\in[k]}$.)

Claim 12 *There exists a PPTA $\mathcal{B}_o$ such that $\mathsf{Adv}^{\mathtt{AIND}\text{-}\mathtt{AI}}_{\mathsf{MBPO},\mathsf{ai}'_\Gamma,\mathcal{B}_o}(k) = |\Pr[\mathsf{Succ}_5] - \Pr[\mathsf{Succ}_6]|$.*

Claim 13 $\Pr[\mathsf{Succ}_6] = 1/2$.

Claims 7 to 13 and the inequality (6) guarantee that there exist PPTAs $\mathcal{B}_h$, $\mathcal{B}_\ell$, and $\mathcal{B}_o$, and a $(k\delta)$-$\mathtt{sPUAI}$ function $\mathsf{ai}'_\Gamma : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^{t'}$ satisfying the inequality (5), as required. Recall that the choice of the PPTA $\mathtt{CCA2}$ adversary $\mathcal{A}$ was arbitrarily, and thus for any PPTA $\mathtt{CCA2}$ adversary we can show its negligible advantage. Hence, Γ is $\mathtt{CCA2}$ secure. This completes the proof of Theorem 2. □

4.3 Extensions

A-priori Fixed Auxiliary Input Function. Note that for both of our constructions in Section 4, the auxiliary input functions under which the building block MBPF obfuscator MBPO needs to be secure, are dependent only on the building block PKE/lossy encryption scheme Π, which is fixed when Π is fixed. In particular, MBPO is required to satisfy $\mathtt{AIND}$-δ-$\mathtt{cPUAI}$ (resp. $\mathtt{AIND}$-δ-$\mathtt{sPUAI}$) security only for t'-bounded δ-$\mathtt{cPUAI}$ (resp. δ-$\mathtt{sPUAI}$) functions with $t'(k) = k \cdot \ell_{\mathtt{PK}}(k) + k \cdot \ell_{\mathtt{C}}(k) + k$. (This t' can be further shortened by using the technique of [36] to reduce the ciphertext size of the Dolev-Dwork-Naor construction [35].) This a-priori bounded output length for auxiliary input functions might make it easier to achieve $\mathtt{AIND}$-δ-$\mathtt{cPUAI}$ (and $\mathtt{AIND}$-δ-$\mathtt{sPUAI}$) secure MBPF obfuscators. We note that a similar observation on the possibility of weakening the requirement regarding auxiliary input by bounding its length is also given in [9].

Using MBPF Obfuscators with Short Point Values. In our constructions, the MBPF obfuscator MBPO needs to obfuscate an MBPF $\mathcal{I}_{\alpha\to\beta}$ whose point value β is relatively long (which consists of k randomness $\{r_i\}_{i\in[k]}$ and a k-bit string K). For our first construction, however, we can shorten the length of a point value of MBPFs that need to be obfuscated by utilizing a pseudorandom generator (PRG). More specifically, let $G : \{0,1\}^k \to \{0,1\}^t$ be a PRG, where $t(k) = k \cdot \ell_{\mathtt{R}}(k) + k$. Then instead of picking $\{r_i\}_{i\in[k]}$ and $K \in \{0,1\}^k$, these values can be generated from a short seed $s \in \{0,1\}^k$ by $\beta = (r_1\| \dots \|r_k\|K) \leftarrow G(s)$, and now we only need to obfuscate $\mathcal{I}_{\alpha\to s}$, instead of $\mathcal{I}_{\alpha\to\beta}$. However, this modification is at the cost of a stronger requirement for $\mathtt{AIND}$-δ-$\mathtt{cPUAI}$ security of MBPO. That is, now δ has to be large enough to incorporate the security of the used PRG. Specifically, if the PRG is ϵ_g-secure, then it is required that $\delta \geq k\epsilon + \epsilon_g$ (where a PRG is said to be ϵ-secure if all PPTA adversaries have at most advantage $\epsilon = \epsilon(k)$ in distinguishing a pseudorandom value from a truly random value for all sufficiently large $k \in \mathbb{N}$). We note that this idea of using a PRG does not work for our second construction, because we cannot use a pseudorandom string as a randomness in the encryption algorithm of a lossy encryption scheme. Using a pseudorandomness violates the statistical lossiness property in general.

A Simpler Construction with $\mathtt{CCA1}$ *Security.* In Appendix E, we show a simpler variant of the proposed construction which employs the Naor-Yung construction-style double encryption [63] (instead of the Dolev-Dwork-Naor-style multiple encryption), leads to a $\mathtt{CCA1}$ secure KEM. (This is the construction partly explained in Introduction.) Interestingly, unlike our $\mathtt{CCA2}$ secure constructions, in the proof of this $\mathtt{CCA1}$ secure variant, we need to use an auxiliary input function that internally runs (a part of) a $\mathtt{CCA1}$ adversary, and thus its output length cannot be a-priori bounded. (Our treatment of an adversary as a part of an auxiliary input function for an MBPF obfuscator might be of independent interest.) For more details, see Appendix E.

5 Relations among Security Notions for MBPF Obfuscators

In this section, we investigate the relations between our new indistinguishability-based security notions for MBPF obfuscators, $\mathtt{AIND}$-δ-$\mathtt{cPUAI}$/$\mathtt{sPUAI}$, and the worst-/average-case virtual black-/grey-box properties in the presence of auxiliary inputs. For the average-case virtual black-/grey-box properties, we consider the auxiliary input functions defined in Section 3.1, and show that our new security notions are implied by the average-case virtual black-/grey-box properties with the same type of auxiliary inputs.

Average-Case Security Definitions. We first formally define the average-case virtual black-/grey-box properties for MBPF obfuscators. As in the worst-case security definitions and $\mathtt{AIND}$ security, for auxiliary input notions we consider the "dependent" auxiliary inputs that depend on the circuit being obfuscated (i.e. the point address and the point value in the case of MBPF obfuscation). For notational convenience, for an MBPF obfuscator MBPO, a probabilistic algorithm $\mathcal{M}$ whose output is restricted to be a bit, and a two-input probabilistic function $\mathsf{ai} : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^*$, we define the following three experiments:

$\mathsf{Expt}^{\mathtt{Real}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{M}}(k)$:	$\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai},\mathcal{M}}(k)$:	$\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai},\mathcal{M}}(k)$:
$\alpha \leftarrow \mathcal{X}_k$	$\alpha \leftarrow \mathcal{X}_k$	$\alpha \leftarrow \mathcal{X}_k$
$\beta \leftarrow \{0,1\}^t$	$\beta \leftarrow \{0,1\}^t$	$\beta \leftarrow \{0,1\}^t$
$z \leftarrow \mathsf{ai}(\alpha,\beta)$	$z \leftarrow \mathsf{ai}(\alpha,\beta)$	$z \leftarrow \mathsf{ai}(\alpha,\beta)$
$\mathtt{DL} \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta})$	Return $b \leftarrow \mathcal{M}^{\mathcal{I}_{\alpha\to\beta}}(1^k, z)$	Return $b \leftarrow \mathcal{M}(1^k, z)$
Return $b \leftarrow \mathcal{M}(1^k, z, \mathtt{DL})$		

(Note that in $\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai},\mathcal{M}}(k)$, the algorithm $\mathcal{M}$ does not have access to any oracle.)

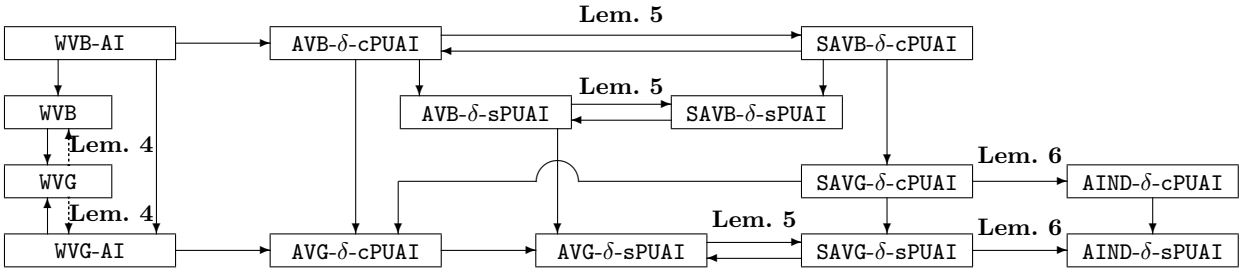

Fig. 2. Relations among security notions for MBPF obfuscators defined in this paper. The arrow "$\mathtt{X} \to \mathtt{Y}$" indicates that X-security implies Y-security. The dotted arrows indicate the implications that hold only for the non-uniform setting in which an adversary (and a simulator) are non-uniform algorithms. In the figure, δ is a negligible function.

Definition 7. *We say that an MBPF obfuscator* MBPO *satisfies*

– *the* average-case virtual black-box property w.r.t. δ-computationally (resp. δ-statistically) partially uninvertible auxiliary input *(*AVB-δ-cPUAI *(resp.* AVB-δ-sPUAI*) security, for short), if for every PPTA $\mathcal{A}$ and all positive polynomials $q = q(k)$ and $\ell = \ell(k)$, there exists a PPTA $\mathcal{S}$ such that for every ℓ-bounded δ-cPUAI (resp. δ-sPUAI) function* ai *and all sufficiently large $k \in \mathbb{N}$, it holds that*

$$\mathsf{Adv}^{\mathtt{A\text{-}MBPO\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A},\mathcal{S}}(k) := |\Pr[\mathsf{Expt}^{\mathtt{Real}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai},\mathcal{S}}(k) = 1]| \leq 1/q.$$

– *the* strong average-case virtual black-box property w.r.t. δ-computationally (resp. δ-statistically) partially uninvertible auxiliary input *(*SAVB-δ-cPUAI *(resp.* SAVB-δ-sPUAI*) security, for short), if for every PPTA $\mathcal{A}$ and all positive polynomials $q = q(k)$ and $\ell = \ell(k)$, there exists a PPTA $\mathcal{S}$ such that for every ℓ-bounded δ-cPUAI (resp. δ-sPUAI) function* ai *and all sufficiently large $k \in \mathbb{N}$, it holds that*

$$\mathsf{Adv}^{\mathtt{SA\text{-}MBPO\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A},\mathcal{S}}(k) := |\Pr[\mathsf{Expt}^{\mathtt{Real}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai},\mathcal{S}}(k) = 1]| \leq 1/q.$$

Furthermore, we define the (strong) average-case virtual grey-box property w.r.t. δ-computationally (resp. δ-statistically) partially uninvertible auxiliary input *((*S*)*AVG-δ-cPUAI *(resp.* (S)AVG-δ-sPUAI*) security for short) for an MBPF obfuscator* MBPO*, in the same way as the definitions for the corresponding virtual black-box properties, except that we replace "a PPTA $\mathcal{S}$" in each definition with "a computationally unbounded algorithm $\mathcal{S}$ that makes only polynomially many queries."*

Relations among Security Notions. Now, we show the relations among security notions, which are summarized in Fig. 2. Most of the relations are obvious. Namely, the virtual black-box properties always imply the virtual grey-box properties for the same class of auxiliary inputs. Furthermore, WVB-AI security implies AVB-δ-cPUAI security for arbitrary (not necessarily negligible) δ, and AVB-δ-cPUAI security implies AVB-δ-sPUAI security because the class of δ-sPUAI functions are smaller than the class of δ-cPUAI functions for the same δ. Moreover, by definition, for both $\mathtt{X} \in \{\delta\text{-cPUAI}, \delta\text{-sPUAI}\}$, SAVB-X and SAVG-X imply AVB-X and AVG-X, respectively, because the former notions consider simulators that do not make any oracle queries and thus can also be used as simulator for the latter.

In the following, we show the implications of the non-trivial directions. The following equivalence is due to the result by Bitansky and Canetti [7]. (Note that the following results are only for non-uniform PPTA adversaries, while our default notions in this paper are with respect to uniform PPTA adversaries.)

Lemma 4. *([7, Propositions 7.3 and A.3]) For MBPF obfuscators,* WVB *security for non-uniform PPTA adversaries with non-uniform PPTA simulators,* WVG *security for non-uniform PPTA adversaries, and* WVG-AI *security for PPTA non-uniform adversaries, are equivalent.*

The next relation is useful for showing the implication to the AIND security notions that we will show later. (The formal proof is given in Appendix D.4.)

Lemma 5. *Let $\delta : \mathbb{N} \to [0,1]$ be a negligible function. For MBPF obfuscators, for both* $\mathtt{X} \in \{\delta\text{-}\mathtt{cPUAI}, \delta\text{-}\mathtt{sPUAI}\}$*,* AVB-X *security and* SAVB-X *security are equivalent. Furthermore,* AVG-δ-sPUAI *security and* SAVG-δ-sPUAI *security are equivalent.*

Intuition. For both cPUAI and sPUAI cases, the implication from the latter to the former is trivial by definition. The implications of the opposite directions can be shown because the partial uninvertibility of an auxiliary input function guarantees that a simulator cannot find the point address of the MBPF being obfuscated and thus having oracle access to an MBPF does not give much advantage. The computational uninvertibility and statistical uninvertibility naturally correspond to the uninvertibility of auxiliary input functions against a PPTA simulator and that against a computationally unbounded simulator, respectively.

Finally, the following implications clarify that AIND notions introduced in Section 3.2 are indeed implied by the average-case virtual black-box/grey-box properties. (The formal proof is given in Appendix D.5.)

Lemma 6. *Let $\delta : \mathbb{N} \to [0,1]$ be a negligible function. For both* $\mathtt{X} \in \{\delta\text{-}\mathtt{cPUAI}, \delta\text{-}\mathtt{sPUAI}\}$*, if an MBPF obfuscator is* SAVG-X *secure, then it is* AIND-X *secure.*

Intuition. This lemma is shown by considering a hybrid experiment in which a (computationally unbounded) simulator $\mathcal{S}$ (due to SAVG-δ-cPUAI/sPUAI security) is given only an auxiliary input $\mathsf{ai}(\alpha, \beta)$ (for randomly chosen (α, β)) as input, and outputs a bit.; By the SAVG-δ-cPUAI/sPUAI security, for both cases $b \in \{0,1\}$, the probability that an adversary (attacking the AIND-δ-cPUAI/sPUAI security) on input $\mathsf{ai}(\alpha, \beta_0)$ and $\mathsf{MBPO}(\mathcal{I}_{\alpha \to \beta_b})$ (for randomly chosen α, β_0, β_1) outputs 1 can be shown to be negligibly close to the probability that the simulator $\mathcal{S}$ outputs 1 in the hybrid experiment, which proves the lemma.

6 Lossy Encryption from Re-randomizable Point Obfuscation

In this section, we show that a re-randomizable point obfuscator yields a lossy encryption scheme. We first recall the definition of re-randomizability [7].

Definition 8. *([7]) Let $\mathcal{X} = \{\mathcal{X}_k\}_{k \in \mathbb{N}}$ be a domain ensemble and let* PO *be a point obfuscator for* $\mathsf{PF}(\mathcal{X})$ *whose randomness space is $\{0,1\}^{\ell(k)}$. We say that* PO *is* re-randomizable *if there exists a PPTA* ReRand *(called the re-randomization algorithm) such that for all $k \in \mathbb{N}$, all $\alpha \in \mathcal{X}_k$, and for all $r \in \{0,1\}^{\ell}$, the distribution of* $\mathsf{ReRand}(\mathsf{PO}(\mathcal{I}_\alpha; r))$ *and the distribution of* $\mathsf{PO}(\mathcal{I}_\alpha)$ *are identical.*

We note that the point obfuscator based on the perfect one-way hash function by Canetti [21] is re-randomizable. (We review the construction in Appendix B.)

Now, we formally describe our proposed lossy encryption scheme. Let $\mathcal{X} = \{\mathcal{X}_k\}_{k \in \mathbb{N}}$ be a domain ensemble, and let PO be a re-randomizable point obfuscator for $\mathsf{PF}(\mathcal{X})$ with the re-randomization algorithm ReRand, and let $t = t(k) > 0$ be a polynomial. Then we construct a lossy encryption scheme $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{LKG})$ whose plaintext space is $\{0,1\}^t$ as in Fig. 3.

Our construction is inspired partly by the construction of a PKE scheme from a re-randomizable point obfuscator due to Bitansky and Canetti [7], and partly by the construction of lossy encryption

$\mathsf{PKG}(1^k)$:	$\mathsf{LKG}(1^k)$:	$\mathsf{Enc}(pk, m)$:	$\mathsf{Dec}(sk, c)$:
$\alpha_0 \leftarrow \mathcal{X}_k$	$\alpha \leftarrow \mathcal{X}_k$	Parse pk as $(\widehat{\mathtt{P}}_0, \widehat{\mathtt{P}}_1)$	Parse c as $(\mathtt{P}_1, \ldots, \mathtt{P}_t)$
$\alpha_1 \leftarrow \mathcal{X}_k \backslash \{\alpha_0\}$	$\widehat{\mathtt{P}}_i \leftarrow \mathsf{PO}(\mathcal{I}_\alpha)$	View m as $(m_1 \Vert \ldots \Vert m_t) \in \{0,1\}^t$	For $i \in [t]$:
$\widehat{\mathtt{P}}_i \leftarrow \mathsf{PO}(\mathcal{I}_{\alpha_i})$	for $i \in \{0,1\}$	$\mathtt{P}_i \leftarrow \mathsf{ReRand}(\widehat{\mathtt{P}}_{m_i})$ for $i \in [t]$	$m_i \leftarrow \begin{cases} 0 & \text{if } \mathtt{P}_i(sk) = \top \\ 1 & \text{otherwise} \end{cases}$
for $i \in \{0,1\}$	$pk \leftarrow (\widehat{\mathtt{P}}_0, \widehat{\mathtt{P}}_1)$	$c \leftarrow (\mathtt{P}_1, \ldots, \mathtt{P}_t)$	End For
$pk \leftarrow (\widehat{\mathtt{P}}_0, \widehat{\mathtt{P}}_1)$	Return pk	Return c	$m \leftarrow (m_1 \Vert \ldots \Vert m_t)$
$sk \leftarrow \alpha_0$			Return m
Return (pk, sk)			

Fig. 3. Lossy encryption from a re-randomizable point obfuscator.

from a re-randomizable encryption scheme due to Hemenway et al. [48]. The following theorem guarantees that Π constructed as above is indeed a lossy encryption scheme. (The formal proof is given in Appendix D.6.)

Theorem 3. *If* PO *is re-randomizable and* 2*-composable, then* Π *constructed as in Fig. 3 is a* 0*-lossy encryption scheme.*

Intuition. Theorem 3 is shown by using the equivalence of t-composability and *t-distributional indistinguishability for coordinate-wise well-spread (CWS) distributions*, established by Bitansky and Canetti [8]. The latter property roughly states that if a set of points $(\alpha_1, \ldots, \alpha_t)$ is chosen from a distribution so that each point α_i has high min-entropy (but α_i's could be arbitrarily correlated), $(\mathsf{PO}(\alpha_1), \ldots, \mathsf{PO}(\alpha_t))$ is computationally indistinguishable from $(\mathsf{PO}(u_1), \ldots, \mathsf{PO}(u_t))$ where each u_i is chosen uniformly at random (see Appendix D.6 for the formal definition). This property can be used to show the indistinguishability of keys, which is easy to see due to the design of PKG and LKG. Moreover, note that a lossy key consists of a pair of obfuscated circuits of point functions with a same point address. Therefore, due to the re-randomizability, an encryption of any plaintext under a lossy key has identical distribution, which implies 0-statistical lossiness.

$\mathtt{CCA2}$ *Secure PKE/KEM Based Solely on Re-randomizable, Composable Point Obfuscators.* Recall that when considering non-uniform PPTA adversaries, $\mathtt{WVB}$ security (with non-uniform PPTA simulators), $\mathtt{WVG}$ security, and $\mathtt{WVG\text{-}AI}$ security for MBPF obfuscators are equivalent (see Lemma 4). Therefore, the $\mathtt{WVG}$ secure MBPF obfuscator for t-bit point values due to [23, 7] based on a $(t+1)$-composable point obfuscator can be used as an $\mathtt{AIND}$-δ-$\mathtt{sPUAI}$ secure MBPF obfuscator (with any negligible δ). Note that if we denote by ℓ the length of the randomness used by ReRand, then the randomness length $\ell_{\mathtt{R}}$ of the lossy encryption scheme Π for the k-bit plaintext space is $\ell_{\mathtt{R}}(k) = k \cdot \ell(k)$. Combining these results with our second generic construction, we obtain the following.

Theorem 4. *Assume there exists a point obfuscator which is (1) re-randomizable where* ReRand *uses $\ell(k)$-bit randomness, and (2) $(k^2 \cdot \ell(k) + k + 1)$-composable for non-uniform PPTA adversaries. Then there exists a* $\mathtt{CCA2}$ *secure PKE scheme/KEM.*

7 Discussion

On Replacing MBPF Obfuscators with SKE. As has been clarified in several previous works [23, 33, 44, 25], there is a strong connection between MBPF obfuscators and SKE schemes. More specifically, an MBPF obfuscator can always be used as a SKE scheme. In order for the opposite direction to be true, among other things regarding security, it is necessary that a SKE scheme has the property called the *unique-key* property [33, 44, 25]. (We recall the formal definition of this property in Appendix A.) Therefore, a variant of our KEM Γ in Section 4 in which an MBPF obfuscator

is replaced with a SKE scheme that has the unique-key property and satisfies the security that we call AIND-δ-cPUAI (and AIND-δ-sPUAI) security (which is defined similarly to that for MBPF obfuscator), can also be proved CCA2 secure.

Since the unique-key property is not satisfied by SKE schemes in general, it may be the case that a SKE scheme is in general a weaker primitive than an MBPF obfuscator, and is potentially easier to achieve. Motivated by this observation, we show another variant of the proposed KEM based on a SKE scheme without the unique-key property. We discuss more about this in Appendix F.

On the Difficulty of Achieving AIND-δ-cPUAI *Security.* We have shown that AIND-δ-sPUAI security is implied by the virtual grey-box properties (see Fig. 2), and thus by the results established by [23, 7] we can construct an AIND-δ-sPUAI secure MBPF obfuscator (or SKE) from any composable point obfuscator. Unfortunately, however, we could not come up with a natural assumption that is sufficient to realize an AIND-δ-cPUAI secure MBPF obfuscator, and we would like to leave it as an interesting open problem. In Appendix F.3, we show that constructing it is at least as difficult as constructing a SKE scheme which is one-time chosen plaintext secure in the presence of hard-to-invert leakage where leakage occurs only from a key. We note that if the random oracle model is allowed, then the MBPF obfuscator by Lynn et al. [57] can be shown to be AIND-δ-cPUAI secure for any negligible δ. This at least suggests that it can be achieved under a strong assumption. For more details, see Appendix G. We conjecture that the MBPF obfuscator by Lynn et al. can be shown to be AIND-δ-cPUAI secure for any negligible δ if we instantiate the random oracle as a family of hash functions satisfying the notion of UCE recently introduced by Bellare et al. [5].

We see that the difficulty of achieving AIND-δ-cPUAI security is that it allows a leakage from a random point address/value (α, β) (or a key/message pair in the context of SKE) that could be arbitrarily correlated, as long as partial uninvertibility is satisfied. This definition allows β to be (a part of) the source of the hardness of the partial uninvertibility. For example, we could consider an auxiliary input function $\mathsf{ai}(\alpha, \beta)$ that returns an encryption of the "plaintext" α under the "key" β, using some SKE scheme, which will be a δ-cPUAI function under a reasonable assumption on the SKE scheme. This situation is quite different from a usual indistinguishability-based security definition (e.g. CPA security of a SKE scheme) in which a point value (or a message in SKE) is chosen by an adversary, and thus cannot be a source of hardness. This is one of the reasons why we cannot straightforwardly use the existing results on MBPF obfuscators/SKE [33, 25] (or a stronger primitive of PKE secure under hard-to-invert leakage [32, 16]). We notice that the formulation of AIND-δ-cPUAI security looks close to the security definition for deterministic encryption in the hard-to-invert auxiliary input setting [20, 72, 73], which considers leakage from a plaintext (as opposed to a key). This setting is in some sense a "dual" of the settings that consider leakage occurring only from a key. We also notice the similarity to the notion called security under *chosen distribution attacks* [4] that considers the security under a correlated leakage occurring from a message and randomness simultaneously (this is a security notion for PKE but can be considered for SKE as well), but this setting does not consider a leakage from a key or leakage with computational uninvertibility. It would be worth clarifying further whether it is possible to leverage techniques from these various kinds of "leakage resilient" cryptography for achieving AIND-δ-cPUAI/sPUAI secure MBPF obfuscators/SKE schemes.

Acknowledgement

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

References

1. P. Ananth, D. Boneh, S. Garg, A. Sahai, and M. Zhandry. Differing-inputs obfuscation and applications, 2013. http://eprint.iacr.org/2013/689.pdf.
2. B. Barak, N. Bitansky, R. Canetti, Y.T. Kalai, O. Paneth, and A. Sahai. Obfuscation for evasive functions. In *Proc. of TCC 2014*, volume 8349 of *LNCS*, pages 26–51. Springer, 2014.
3. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S.P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *Proc. of CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, 2001.
4. M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249. Springer, 2009.
5. M. Bellare, V.T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs. In *Proc. of CRYPTO 2013(2)*, volume 8043 of *LNCS*, pages 398–415. Springer, 2013.
6. M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Proc. of EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, 2009.
7. N. Bitansky and R. Canetti. On strong simulation and composable point obfuscation, 2010. Full version of [8]. http://eprint.iacr.org/2010/414.
8. N. Bitansky and R. Canetti. On strong simulation and composable point obfuscation. In *Proc. of CRYPTO 2010*, volume 6223 of *LNCS*, pages 520–537. Springer, 2010.
9. N. Bitansky and O. Paneth. Point obfuscation and 3-round zero-knowledge. In *Proc. of TCC 2012*, volume 7194 of *LNCS*, pages 190–208. Springer, 2012.
10. D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *Proc. of CRYPTO 1998*, volume 1462 of *LNCS*, pages 1–12. Springer, 1998.
11. M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *Proc. of STOC 1988*, pages 103–112. ACM, 1988.
12. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Proc. of CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.
13. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In *Proc. of CT-RSA 2005*, volume 3376 of *LNCS*, pages 87–103. Springer, 2005.
14. D. Boneh, P.A. Papakonstantinou, C. Rackoff, Y. Vahlis, and B. Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *Proc. of FOCS 2008*, pages 283–292. IEEE Computer Society, 2008.
15. E. Boyle, K.-M. Chung, and R. Pass. On extractability obfuscation. In *Proc. of TCC 2014*, volume 8349 of *LNCS*, pages 52–73. Springer, 2014.
16. Z. Brakerski and S. Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *Proc. of CRYPTO 2010*, volume 6223 of *LNCS*, pages 1–20. Springer, 2010.
17. Z. Brakerski and G.N. Rothblum. Obfuscating conjunctions. In *Proc. of CRYPTO 2013(2)*, volume 8043 of *LNCS*, pages 416–434. Springer, 2013.
18. Z. Brakerski and G.N. Rothblum. Black-box obfuscation for d-CNFs. In *Proc. of ITCS 2014*, pages 235–250. ACM, 2014.
19. Z. Brakerski and G.N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In *Proc. of TCC 2014*, volume 8349 of *LNCS*, pages 1–25. Springer, 2014.
20. Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In *Proc. of CRYPTO 2011*, 6841, pages 543–560. Springer, 2011.
21. R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Proc. of CRYPTO 1997*, volume 1294 of *LNCS*, pages 455–469. Springer, 1997.
22. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proc. of FOCS 2001*, pages 136–145. IEEE Computer Society, 2001.
23. R. Canetti and R.R. Dakdouk. Obfuscating point functions with multibit output. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 489–508. Springer, 2008.
24. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Proc. of EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.
25. R. Canetti, Y.T. Kalai, M. Varia, and D. Wichs. On symmetric encryption and point obfuscation. In *Proc. of TCC 2010*, volume 5978 of *LNCS*, pages 52–71. Springer, 2010.
26. R. Canetti, Y.T. Kalai, M. Varia, and D. Wichs. On symmetric encryption and point obfuscation, 2010. Full version of [25]. http://eprint.iacr.org/2010/049.
27. R. Canetti, G.N. Rothblum, and M. Varia. Obfuscation of hyperplane membership. In *Proc. of TCC 2010*, volume 5978 of *LNCS*, pages 72–89. Springer, 2010.

28. N. Chandran, M. Chase, and V. Vaikuntanathan. Functional re-encryption and collusion-resistant obfuscation. In *Proc. of TCC 2012*, volume 7194 of *LNCS*, pages 404–421. Springer, 2012.
29. J.S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In *Proc. of CRYPTO 2013(1)*, volume 8042 of *LNCS*. Springer, 2013.
30. 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.
31. D. Dachman-Soled. A black-box construction of a CCA2 encryption scheme from a plaintext aware (sPA1) encryption scheme. In *Proc. of PKC 2014*, volume 8383 of *LNCS*, pages 37–55. Springer, 2014.
32. Y. Dodis, S. Goldwasser, Y.T. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption with auxiliary inputs. In *Proc. of TCC 2010*, volume 5978 of *LNCS*, pages 361–381. Springer, 2010.
33. Y. Dodis, Y.T. Kalai, and S. Lovett. On cryptography with auxiliary input. In *Proc. of STOC 2009*, pages 621–630. ACM, 2009.
34. Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *Proc. of STOC 2005*, pages 654–663. ACM, 2005.
35. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proc. of STOC 1991*, pages 542–552. ACM, 1991.
36. R. Dowsley, G. Hanaoka, H. Imai, and A.C.A. Nascimento. Reducing the ciphertext size of Dolev-Dwork-Naor like public key cryptosystems, 2009. http://eprint.iacr.org/2009/271.
37. S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *Proc. of EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, 2013.
38. S. Garg, C. Gentry, S. Halevi, and M. Raykova. Two-round secure MPC from indistinguishability obfuscation. In *Proc. of TCC 2014*, volume 8349 of *LNCS*, pages 74–94. Springer, 2014.
39. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B Waters. Candidate indistinguishability obfuscation and functional encryption for all curcuits. In *Proc. of FOCS 2013*, pages 40–49. IEEE Computer Society, 2013.
40. Y. Gertner, T. Malkin, and S. Myers. Towards a separation of semantic and CCA security for public key encryption. In *Proc. of TCC 2007*, volume 4392 of *LNCS*, pages 434–455. Springer, 2007.
41. Y. Gertner, T. Malkin, and O. Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *Proc. of FOCS 2001*, pages 126–135. IEEE Computer Society, 2001.
42. O. Goldreich and R.D. Rothblum. Enhancements of trapdoor permutations. *J. of Cryptology*, 26(3):484–512, 2013.
43. S. Goldwasser and Y.T. Kalai. On the impossibility of obfuscation with auxiliary input. In *Proc. of FOCS 2005*, pages 553–562. IEEE Computer Society, 2005.
44. S. Goldwasser, Y.T. Kalai, C. Peikart, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In *Proc. of ICS 2010*, pages 230–240. Tsinghua University Press, 2010.
45. S. Goldwasser and S. Micali. Probabilistic encryption. *J. of Computer and System Sciences*, 28(2):270–299, 1984.
46. S. Goldwasser and G.N. Rothblum. On best-possible obfuscation. In *Proc. of TCC 2007*, volume 4392 of *LNCS*, pages 194–213. Springer, 2007.
47. S. Hada. Secure obfuscation for encrypted signatures. In *Proc. of EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 92–112. Springer, 2010.
48. B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *Proc. of ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 70–88. Springer, 2011.
49. B. Hemenway and R. Ostrovsky. On homomorphic encryption and chosen-ciphertext security. In *Proc. of PKC 2012*, volume 7293 of *LNCS*, pages 52–65. Springer, 2012.
50. B. Hemenway and R. Ostrovsky. Building lossy trapdoor functions from lossy encryption. In *Proc. of ASIACRYPT 2013(2)*, volume 8270 of *LNCS*, pages 241–260. Springer, 2013.
51. 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.
52. S. Hohenberger, G.N. Rothblum, A. Shelat, and V. Vaikuntanathan. Securely obfuscating re-encryption. In *Proc. of TCC 2007*, volume 4392 of *LNCS*, pages 233–252. Springer, 2007.
53. S. Hohenberger, A. Sahai, and B. Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation, 2013. http://eprint.iacr.org/2013/509. To appear in EUROCRYPT 2014.
54. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *Proc. of TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, 2006.
55. E. Kiltz, P. Mohassel, and A. O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *Proc. of EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 673–692. Springer, 2010.
56. H. Lin and S. Tessaro. Amplification of chosen-ciphertext security. In *Proc. of EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 503–519. Springer, 2013.
57. B. Lynn, M. Prabhakaran, and A. Sahai. Positive results and techniques for obfuscation. In *Proc. of EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 20–39. Springer, 2004.

58. 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.
59. T. Matsuda and G. Hanaoka. Chosen ciphertext security via UCE. In *Proc. of PKC 2014*, volume 8383 of *LNCS*, pages 56–76. Springer, 2014.
60. P. Mol and S. Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. In *Proc. of PKC 2010*, volume 6056 of *LNCS*, pages 296–311. Springer, 2010.
61. S. Myers and A. Shelat. Bit encryption is complete. In *FOCS 2009*, pages 607–616. IEEE Computer Society, 2009.
62. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proc. of STOC 1989*, pages 33–43. ACM, 1989.
63. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proc. of STOC 1990*, pages 427–437. ACM, 1990.
64. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *Proc. of STOC 2008*, pages 187–196. ACM, 2008.
65. C. Rackoff and D.R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Proc. of CRYPTO 1991*, volume 576 of *LNCS*, pages 433–444. Springer, 1992.
66. A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *Proc. of TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, 2009.
67. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *Proc. of FOCS 1999*, pages 543–553. IEEE Computer Society, 1999.
68. A. Sahai and B. Waters. How to use indistinguishability obfuscation: Deniable encryption, and more, 2013. http://eprint.iacr.org/2013/454. To appear in STOC 2014.
69. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53. Springer, 1985.
70. H. Wee. On obfuscating point functions. In *Proc. of STOC 2005*, pages 523–532. ACM, 2005.
71. H. Wee. Efficient chosen-ciphertext security via extractable hash proofs. In *Proc. of CRYPTO 2010*, volume 6223 of *LNCS*, pages 314–332. Springer, 2010.
72. H. Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In *Proc. of EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 246–262. Springer, 2012.
73. X. Xie, R. Xue, and R. Zhang. Deterministic public key encryption and identity-based encryption from lattices in the auxiliary-input setting. In *Proc. of SCN 2012*, volume 7485 of *LNCS*, pages 1–18. Springer, 2012.

A Basic Cryptographic Primitives

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

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

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

We define the "*public key length*" $\ell_{\mathtt{PK}}(k)$ as the length of pk output by $\mathsf{PKG}(1^k)$. Moreover, if Enc can encrypt k-bit plaintexts (for security parameter k), we define the "*randomness length*" $\ell_{\mathtt{R}}(k)$ and the "*ciphertext length*" $\ell_{\mathtt{C}}(k)$, respectively, as the length of randomness used by Enc and the length of ciphertexts output from Enc.

We say that a PKE scheme Π is ϵ-$\mathtt{CPA}$ secure[4] if for all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and for all sufficiently large $k \in \mathbb{N}$, it holds that $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k) = 1] - 1/2| \leq \epsilon(k)$, where the $\mathtt{CPA}$ experiment $\mathsf{Expt}^{\mathtt{CPA}}_{\Pi,\mathcal{A}}(k)$ is defined as follows:

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

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

[4] See Footnote 2.

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

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

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

For $\mathtt{ATK} \in \{\mathtt{CCA1}, \mathtt{CCA2}\}$, we say that a KEM Γ is $\mathtt{ATK}$ secure if for all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{ATK}}_{\Gamma,\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the $\mathtt{CCA2}$ experiment $\mathsf{Expt}^{\mathtt{CCA2}}_{\Gamma,\mathcal{A}}(k)$ is defined as follows:

$$\mathsf{Expt}^{\mathtt{CCA2}}_{\Gamma,\mathcal{A}}(k) : [\ (pk, sk) \leftarrow \mathsf{KKG}(1^k);\ \mathsf{st} \leftarrow \mathcal{A}_1^{\mathsf{Decap}(sk,\cdot)}(pk);\ (c^*, K_1^*) \leftarrow \mathsf{Encap}(pk);$$
$$K_0^* \leftarrow \{0,1\}^k;\ b \leftarrow \{0,1\};\ b' \leftarrow \mathcal{A}_2^{\mathsf{Decap}(sk,\cdot)}(\mathsf{st}, c^*, K_b^*);\ \text{Return } (b' \stackrel{?}{=} b)\ ],$$

where $\mathcal{A}_2$ is not allowed to query c^*. The $\mathtt{CCA1}$ experiment $\mathsf{Expt}^{\mathtt{CCA1}}_{\Gamma,\mathcal{A}}(k)$ is defined similarly to the $\mathtt{CCA2}$ experiment, except that $\mathcal{A}_2$ is not allowed to ask any query.

Tag-Based Key Encapsulation Mechanism. A tag-based key encapsulation mechanism (TBKEM) is the KEM-analogue of tag-based encryption [58, 54], and consists of the three PPTAs $(\mathsf{TKG}, \mathsf{TEncap}, \mathsf{TDecap})$ with the following interface:

$$\underline{\textbf{Key Generation:}} \quad (pk, sk) \leftarrow \mathsf{TKG}(1^k) \qquad \underline{\textbf{Encapsulation:}} \quad (c, K) \leftarrow \mathsf{TEncap}(pk, \mathsf{tag}) \qquad \underline{\textbf{Decapsulation:}} \quad K \text{ (or } \perp) \leftarrow \mathsf{TDecap}(sk, \mathsf{tag}, c)$$

where TDecap 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 and a "tag" tag. We require for all $k \in \mathbb{N}$, all (pk, sk) output by $\mathsf{TKG}(1^k)$, all tags tag, and all $(c, K) \leftarrow \mathsf{TEncap}(pk, \mathsf{tag})$, it holds that $\mathsf{TDecap}(sk, \mathsf{tag}, c) = K$.

We say that a TBKEM $\mathcal{T}$ is secure against selective-tag, weak chosen ciphertext attacks [54] (for short, $\mathtt{wCCA}$ secure) if for all PPTAs $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the $\mathtt{wCCA}$ experiment $\mathsf{Expt}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(k)$ is defined as follows:

$$\mathsf{Expt}^{\mathtt{wCCA}}_{\mathcal{T},\mathcal{A}}(k) : [\ (\mathsf{tag}^*, \mathsf{st}) \leftarrow \mathcal{A}_0(1^k);\ (pk, sk) \leftarrow \mathsf{TKG}(1^k);\ \mathsf{st}' \leftarrow \mathcal{A}_1^{\mathsf{TDecap}(sk,\cdot,\cdot)}(\mathsf{st}, pk);\ b \leftarrow \{0,1\};$$
$$(c^*, K_1^*) \leftarrow \mathsf{TEncap}(pk, \mathsf{tag}^*);\ K_0^* \leftarrow \{0,1\}^k;\ b' \leftarrow \mathcal{A}_2^{\mathsf{TDecap}(sk,\cdot,\cdot)}(\mathsf{st}', c^*, K_b^*);\ \text{Return } (b' \stackrel{?}{=} b)\ ],$$

where $\mathcal{A}_1$ and $\mathcal{A}_2$ are not allowed to submit a tag/ciphertext pair (tag, c) satisfying $\mathsf{tag} = \mathsf{tag}^*$ to the oracle.

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

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

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

$$\underline{\textbf{Encryption:}} \qquad \underline{\textbf{Decryption:}}$$
$$c \leftarrow \mathsf{SEnc}(K, m) \qquad m \text{ (or } \perp) \leftarrow \mathsf{SDec}(K, c)$$

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 require for all $k \in \mathbb{N}$, all $K \in \{0,1\}^k$, and all plaintexts m, it holds that $\mathsf{SDec}(K, \mathsf{SEnc}(K, m)) = m$.

We say that a SKE scheme E satisfies *indistinguishability under one-time encryption* ($\mathtt{OT}$ security, for short) if for all PPTAs $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathsf{Adv}^{\mathtt{OT}}_{E,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{OT}}_{E,\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the $\mathtt{OT}$ experiment $\mathsf{Expt}^{\mathtt{OT}}_{E,\mathcal{A}}(k)$ is defined as follows:

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

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

We say that a SKE scheme E has the *unique-key property* [33, 44, 25] if there exists a negligible function ϵ such that for all keys $K, K' \in \{0,1\}^k$ satisfying $K \neq K'$ and all plaintexts m (in the plaintext space supported by E), it holds that

$$\Pr[\mathsf{SDec}(K', \mathsf{SEnc}(K, m)) \neq \perp] \leq \epsilon(k),$$

where the probability is over the randomness consumed by SEnc.

B Concrete Instantiations of Point/MBPF Obfuscators

Composable Point Obfuscator. Here we recall the point obfuscator due to Canetti, which was originally introduced as a perfectly one-way hash function [21]. Let $\mathbb{G}$ be a cyclic group with prime order p (where the size of p is determined by the security parameter k). Then, consider the following point obfuscator PO for $\mathsf{PF}(\mathbb{Z}_p)$:

$\mathsf{PO}(\mathcal{I}_\alpha)$**:** (where $\alpha \in \mathbb{Z}_p$) Pick a group element $r \leftarrow \mathbb{G}$ uniformly at random, and outputs the circuit $\mathcal{C}_{r,r^\alpha}(\cdot) : \mathbb{Z}_p \to \{\top, \perp\}$, where $\mathcal{C}_{A,B}$ is the circuit which takes $x \in \mathbb{Z}_p$ as input, and outputs $\top$ if $A^x = B$ and otherwise outputs $\perp$.

Bitansky and Canetti [7] showed that the above point obfuscator is t-composable, under a strong variant of the decisional Diffie-Hellman (DDH) assumption, called the t-strong vector DDH (t-SVDDH) assumption (see (see [7] for a formal definition).

We remark that as mentioned in [7], the point obfuscator based on the t-SVDDH assumption described here satisfies the re-randomizability in the sense of Definition 8. Specifically, we can just re-randomize two group elements in an obfuscated circuit output from PO without changing the point address.

$\mathtt{WVG}$ *Secure MBPF Obfuscator from Composable Point Obfuscator.* We recall the construction of an MBPF obfuscator based on a composable point obfuscator, due to Canetti and Dakdouk [23] and Bitansky and Canetti [7]. Let PO be a point obfuscator for $\mathsf{PF}(\mathcal{X})$ and let $t = t(k) > 0$ be a polynomial. Then an MBPF obfuscator MBPO for $\mathsf{MBPF}(\mathcal{X}, t)$ is constructed as in Fig. 4.

Based on the result of [23], Bitansky and Canetti [7] showed that if PO is $(t+1)$-composable, then the MBPF obfuscator MBPO constructed as in Fig. 4 is $\mathtt{WVG}$ secure. By instantiating this conversion with the above mentioned point obfuscator, we obtain a $\mathtt{WVG}$ secure t-bit-output MBPF obfuscator under the $(t+1)$-SVDDH assumption.

$\mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta})$:	$\mathcal{C}_{\mathtt{P}_0,\ldots,\mathtt{P}_t}(x)$:
$\mathtt{P}_0 \leftarrow \mathsf{PO}(\mathcal{I}_\alpha)$ View β as $(\beta_1 \| \ldots \| \beta_t) \in \{0,1\}^t$ $\alpha' \leftarrow \mathcal{X}_k \backslash \{\alpha\}$ For $i \in [t]$: $\mathtt{P}_i \leftarrow \begin{cases} \mathsf{PO}(\mathcal{I}_\alpha) & \text{if } \beta_i = 1 \\ \mathsf{PO}(\mathcal{I}_{\alpha'}) & \text{otherwise} \end{cases}$ End For Return $\mathtt{DL} \leftarrow \mathcal{C}_{\mathtt{P}_0,\ldots,\mathtt{P}_t}$.	If $\mathtt{P}_0(x) = \perp$ then return $\perp$ For $i \in [t]$: $\beta_i \leftarrow \begin{cases} 1 & \text{if } \mathtt{P}_i(x) = \top \\ 0 & \text{otherwise} \end{cases}$ End For Return $\beta \leftarrow (\beta_1 \| \ldots \| \beta_t)$.

Fig. 4. The construction of an MBPF obfuscator MBPO from a composable point obfuscator PO [23, 7]. MBPO takes an MBPF $\mathcal{I}_{\alpha\to\beta}$ as input, and returns a circuit $\mathtt{DL} = \mathcal{C}_{\mathtt{P}_0,\ldots,\mathtt{P}_t}$ that is described in the right column.

$\mathsf{PKG}^k(1^k)$:	$\mathsf{LKG}^k(1^k)$:	$\mathsf{Enc}^k(PK, m)$:	$\mathsf{Dec}^k(SK, C)$:
$(pk_i, sk_i) \leftarrow \mathsf{PKG}(1^k)$ for $i \in [k]$ $PK \leftarrow \{pk_i\}_{i\in[k]}$ $SK \leftarrow \{sk_i\}_{i\in[k]}$ Return (PK, SK)	$pk_i \leftarrow \mathsf{LKG}(1^k)$ for $i \in [k]$ $PK \leftarrow \{pk_i\}_{i\in[k]}$ Return PK	Parse PK as $\{pk_i\}_{i\in[k]}$ $c_i \leftarrow \mathsf{Enc}(pk_i, m)$ for $i \in [k]$ $C \leftarrow \{c_i\}_{i\in[k]}$ Return C	Parse SK as $\{sk_i\}_{i\in[k]}$ Parse C as $\{c_i\}_{i\in[k]}$ $m_i \leftarrow \mathsf{Dec}(sk_i, c_i)$ for $i \in [k]$ If $m_1 = \cdots = m_k$ then return m_1 else return $\perp$

Fig. 5. The k-repetition construction Π^k based on a PKE scheme/lossy encryption scheme Π. (In the former case, we ignore the algorithms LKG and LKG^k.)

C k-Repetition Construction of PKE/Lossy Encryption and Its Security

In our proposed constructions, we will use the k-repetition construction of a PKE scheme and lossy encryption scheme, and thus we review them for self-containment of the paper.

Let $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme. Then the k-repetition PKE construction $\Pi^k = (\mathsf{PKG}^k, \mathsf{Enc}^k, \mathsf{Dec}^k)$ is as in Fig. 5. In case Π is a lossy encryption, we naturally define the lossy key generation algorithm LKG^k for Π^k based on LKG of Π.

The security properties of the k-repetition construction are guaranteed by the following lemmas (which can be proved by applying a standard hybrid argument, and thus omitted).

Lemma 7. *If Π is a ϵ-$\mathtt{CPA}$ secure PKE scheme, then the k-repetition construction Π^k based on Π is $(k\epsilon)$-$\mathtt{CPA}$ secure.*

Lemma 8. *If Π is a ϵ-lossy encryption scheme, then the k-repetition construction Π^k based on Π is a $(k\epsilon)$-lossy encryption scheme.*

D Postponed Proofs

D.1 Proof of Lemma 1

Let $\delta : \mathbb{N} \to [0,1]$ be any non-negligible function and $t = t(k) \geq 1$ be a polynomial. Let MBPO be an MBPF obfuscator for $\mathsf{MBPF}(\mathcal{X}, t)$ We show that for this δ and MBPO, there exist a PPTA $\mathcal{A}$ and a δ-$\mathtt{sPUAI}$ function ai such that $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) \geq \delta(k)/2$, which means that MBPO is not $\mathtt{AIND}$-δ-$\mathtt{sPUAI}$ secure, and hence will prove the lemma.

First, define the following probabilistic function ai:

$$\mathsf{ai}(\alpha, \beta) = \begin{cases} (\alpha, \beta) & \text{with probability } \delta(k) \\ (\perp, \perp) & \text{with probability } 1 - \delta(k) \end{cases}$$

It is straightforward to see that ai is a δ-$\mathtt{sPUAI}$ function. In particular, given $z \leftarrow \mathsf{ai}(\alpha, \beta)$ for a randomly chosen (α, β), even a computationally unbounded adversary can output α with probability at most $\delta + 1/|\mathcal{X}_k|$.

Next, consider the following adversary $\mathcal{A}$ that runs in the experiment $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k)$:

$\mathcal{A}(1^k, z = (\alpha', \beta'), \mathtt{DL})$**:** (where $z \leftarrow \mathsf{ai}(\alpha, \beta_0)$ and $\mathtt{DL} \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha \to \beta_b})$ and b is the challenge bit for $\mathcal{A}$) Firstly $\mathcal{A}$ checks whether $z = (\bot, \bot)$. If this is the case, then $\mathcal{A}$ outputs a random bit $b' \leftarrow \{0,1\}$ and terminates. Otherwise ($(\alpha', \beta') = (\alpha, \beta_0)$), $\mathcal{A}$ runs $\beta_b \leftarrow \mathtt{DL}(\alpha')$. If $\beta' = \beta_b$, then $\mathcal{A}$ sets $b' \leftarrow 0$, otherwise sets $b' \leftarrow 1$, and terminates with output b'.

Let Good be the event that $z \neq (\bot, \bot)$ occurs. Clearly, we have $\Pr[\mathsf{Good}] = \delta(k)$. By definition, if Good does not occur, then $\mathcal{A}$ outputs a random bit, and thus we have $\Pr[b' = b | \overline{\mathsf{Good}}] = 1/2$. Now consider the case when Good occurs. If $b = 0$, then $\mathcal{A}$ always outputs $b' = 0$, while if $b = 1$, $\mathcal{A}$ outputs 0 only when $\beta_0 = \beta_1$, which occurs with probability exactly 2^{-t}. Therefore, we have $\Pr[b' = 0 | \mathsf{Good} \wedge b = 0] = 1$ and $\Pr[b' = 0 | \mathsf{Good} \wedge b = 1] = 2^{-t}$.

Using the above, $\mathcal{A}$'s $\mathtt{AIND\text{-}AI}$ advantage can be calculated as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) &= 2 \cdot |\Pr[b' = b] - \frac{1}{2}| \\
&= 2 \cdot |\Pr[b' = b | \mathsf{Good}] \cdot \Pr[\mathsf{Good}] + \Pr[b' = b | \overline{\mathsf{Good}}] \cdot \Pr[\overline{\mathsf{Good}}] - \frac{1}{2}| \\
&= 2 \cdot |\Pr[b' = b | \mathsf{Good}] \cdot \delta(k) + \frac{1}{2} \cdot (1 - \delta(k)) - \frac{1}{2}| \\
&= 2\delta(k) \cdot |\Pr[b' = b | \mathsf{Good}] - \frac{1}{2}| \\
&= \delta(k) \cdot |\Pr[b' = 0 | \mathsf{Good} \wedge b = 0] - \Pr[b' = 0 | \mathsf{Good} \wedge b = 1]| \\
&= \delta(k) \cdot (1 - 2^{-t}) \geq \frac{\delta(k)}{2},
\end{aligned}$$

where the last inequality is due to $t \geq 1$. This completes the proof of Lemma 1. □

D.2 Proof of Lemma 2

Let $PK = (\{pk_i^{(j)}\}_{i \in [k], j \in \{0,1\}}, \kappa)$ and $SK = (\{sk_i^{(j)}\}_{i \in [k], j \in \{0,1\}}, \kappa)$ be a key pair output by $\mathsf{KKG}(1^k)$, and let $C = (c_1, \ldots, c_k, \mathtt{DL})$ be a ciphertext output by $\mathsf{Encap}(PK)$. Let $h = (h_1 \| \ldots \| h_k) = \mathsf{H}_\kappa(\mathtt{DL})$.

Now, fix arbitrarily a ciphertext $C' = (c'_1, \ldots, c'_k, \mathtt{DL}')$ satisfying $\mathtt{DL}' = \mathtt{DL}$ and $(c'_1, \ldots, c'_k) \neq (c_1, \ldots, c_k)$. We show that for this C', it holds that $\mathsf{Decap}(SK, C') = \bot$. Let $\alpha = \mathsf{Dec}(sk_1^{(h_1)}, c'_1)$. Consider the following two cases:

Case $\alpha \neq \bot$ and $\mathtt{DL}(\alpha) = \beta = (r_1 \| \ldots \| r_k \| K) \neq \bot$: Since $\mathtt{DL}$ is an obfuscation of an MBPF, when it is executed, the output value is not $\bot$ if and only if the input is the point address. Thus, the fact that $\mathtt{DL}(\alpha) = \beta \neq \bot$ must mean that $\mathtt{DL}$ is indeed an obfuscation of $\mathcal{I}_{\alpha \to \beta}$. However, recall that by the definition of Encap and the ciphertext C, we have that $c_i = \mathsf{Enc}(pk_i^{(h_i)}, \alpha; r_i)$ for all $i \in [k]$. Furthermore, the second condition on C' implies that there exists at least one index $\ell \in [k]$ such that $c'_\ell \neq c_\ell$. Therefore, under this index ℓ, we have $c'_\ell \neq \mathsf{Enc}(pk_\ell^{(h_\ell)}, \alpha; r_\ell)$, and thus the validity check by re-encryption performed at the last step of Decap cannot be satisfied, and $\mathsf{Decap}(SK, C')$ outputs $\bot$.

Otherwise (i.e. $\alpha = \bot$ or $\mathtt{DL}(\alpha) = \bot$): In this case, $\mathsf{Decap}(SK, C')$ clearly outputs $\bot$.

We have shown that $\mathsf{Decap}(SK, C') = \bot$ holds for both cases. This completes the proof of Lemma 2. □

D.3 Proof of Lemma 3

Let $h^* = (h_1^* \| \dots \| h_k^*) \in \{0,1\}^k$, PK, SK, and $\widehat{SK}_{h^*}$ be as stated in the lemma. Fix arbitrarily a (possibly invalid) ciphertext $C = (c_1, \dots, c_k, \mathtt{DL})$ satisfying $\mathsf{H}_\kappa(\mathtt{DL}) = h \neq h^*$ and let $h = (h_1 \| \dots \| h_k) \in \{0,1\}^k$. Let $\ell \in [k]$ be the smallest index such that $h_\ell = 1 - h_\ell^*$. (Since $h \neq h^*$, there must exist $\ell \in [k]$ such that $h_\ell \neq h_\ell^*$, and hence $h_\ell = 1 - h_\ell^*$.) For notational convenience, let $\alpha_1 = \mathsf{Dec}(sk_1^{(h_1)}, c_1)$ and $\alpha_\ell = \mathsf{Dec}(sk_\ell^{(1-h_\ell^*)}, c_\ell) (= \mathsf{Dec}(sk^{(h_\ell)}, c_\ell))$. Let us consider the following two cases:

Case $\alpha_1 = \alpha_\ell$: Both Decap and $\mathsf{AltDecap}$ proceed identically after the fifth step, and thus the outputs from both algorithms must agree, regardless of the validity of C.

Case $\alpha_1 \neq \alpha_\ell$: In this case, both Decap and $\mathsf{AltDecap}$ return $\bot$. Specifically, $\alpha_1 \neq \alpha_\ell$, $h_\ell = 1 - h_\ell^*$, and the correctness of the PKE scheme Π imply that there does not exist r_ℓ such that $\mathsf{Enc}(pk_\ell^{(h_\ell)}, \alpha_1; r_\ell) = c_\ell$, and thus Decap returns $\bot$ when it performs the validity check of c_ℓ by re-encryption at its last step at the latest (it could return $\bot$ earlier if $\alpha_1 = \bot$ or $\mathtt{DL}(\alpha_1) = \bot$). Symmetrically, there does not exist r_1 such that $\mathsf{Enc}(pk_1^{(h_1)}, \alpha_\ell; r_1) = c_1$, and thus $\mathsf{AltDecap}$ returns $\bot$ in its last step at the latest (it could return $\bot$ earlier as above).

We have seen that $\mathsf{Decap}(SK, C) = \mathsf{AltDecap}(\widehat{SK}_{h^*}, C)$ holds for all ciphertexts $C = (c_1, \dots, c_k, \mathtt{DL})$ satisfying $\mathsf{H}_\kappa(\mathtt{DL}) \neq h^*$. This completes the proof of Lemma 3. □

D.4 Proof of Lemma 5

As already mentioned, the implications from $\mathtt{SAVB\text{-}X}$ security to $\mathtt{AVB\text{-}X}$ security for both $\mathtt{X} \in \{\delta\mathtt{\text{-}cPUAI}, \delta\mathtt{\text{-}sPUAI}\}$, and the implication from $\mathtt{SAVG}$-δ-$\mathtt{sPUAI}$ security to $\mathtt{AVG}$-δ-$\mathtt{sPUAI}$ security, are trivial by definition. Therefore, in the following we consider the opposite directions.

Let MBPO be an MBPF obfuscator for $\mathsf{MBPF}(\mathcal{X}, t)$ and let δ be a negligible function. We first show the implication from $\mathtt{AVB}$-δ-$\mathtt{cPUAI}$ security to $\mathtt{SAVB}$-δ-$\mathtt{cPUAI}$ security. Assume that MBPO is $\mathtt{AVB}$-δ-$\mathtt{cPUAI}$ secure.

Now, fix any PPTA adversary $\mathcal{A}$ against MBPO in the sense of $\mathtt{SAVB}$-δ-$\mathtt{cPUAI}$ security, and also fix any positive polynomials $q = q(k)$ and $\ell = \ell(k)$. We will show that for these $\mathcal{A}$, q and ℓ, there exists a PPTA simulator $\mathcal{S}$ such that for all ℓ-bounded δ-$\mathtt{cPUAI}$ functions ai and for all sufficiently large $k \in \mathbb{N}$, it holds that $\mathsf{Adv}^{\mathtt{SA\text{-}MBPO\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A},\mathcal{S}}(k) \leq 1/q(k)$.

Note that $\mathcal{A}$ can be viewed as an adversary in the sense of the $\mathtt{AVB}$-δ-$\mathtt{cPUAI}$ security as well. Then, since MBPO is assumed to be $\mathtt{AVB}$-δ-$\mathtt{cPUAI}$ secure, for this $\mathcal{A}$ and the polynomials $2q$ and ℓ, there exists a PPTA simulator $\mathcal{S}'$ (corresponding to $\mathcal{A}$, $2q$, and ℓ) such that for any ℓ-bounded δ-$\mathtt{cPUAI}$ function ai' and for all sufficiently large $k \in \mathbb{N}$, it holds that

$$\mathsf{Adv}^{\mathtt{A\text{-}MBPO\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}',\mathcal{A},\mathcal{S}'}(k) = |\Pr[\mathsf{Expt}^{\mathtt{Real}}_{\mathsf{MBPO},\mathsf{ai}',\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai}',\mathcal{S}'}(k) = 1]| \leq \frac{1}{2q(k)}. \tag{7}$$

Now, using this $\mathcal{S}'$, we construct a PPTA simulator $\mathcal{S}$ (corresponding to the above $\mathcal{A}$, q, and ℓ) that does *not* use the oracle for the MBPF $\mathcal{I}_{\alpha\to\beta}$, as follows.

$\mathcal{S}(1^k, z)$: (where $z \leftarrow \mathsf{ai}(\alpha, \beta)$ and ai is any δ-$\mathtt{cPUAI}$ function) $\mathcal{S}$ runs $\mathcal{S}'(1^k, z)$. $\mathcal{S}$ responds to all oracle queries from $\mathcal{S}'$ (to $\mathcal{I}_{\alpha\to\beta}$) with $\bot$. When $\mathcal{S}'$ terminates with output a bit b, $\mathcal{S}$ outputs this b and terminates.

Fix an arbitrary ℓ-bounded δ-$\mathtt{cPUAI}$ function ai. Let Bad be the event that $\mathcal{S}'$ makes a query α in the experiment $\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai},\mathcal{S}'}(k)$. By definition, unless $\mathcal{S}'$ issues the query that causes the event Bad, $\mathcal{S}$ perfectly simulates $\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai},\mathcal{S}'}(k)$ for $\mathcal{S}'$, and $\mathcal{S}$ uses the output of $\mathcal{S}'$. Therefore, we have

$$|\Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai},\mathcal{S}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai},\mathcal{S}'}(k) = 1]| \leq \Pr[\mathsf{Bad}]. \tag{8}$$

We show the following claim.

Claim 14 $\Pr[\mathsf{Bad}]$ *is negligible.*

Proof of Claim 14. Let $Q = Q(k) > 0$ be the number of queries made by $\mathcal{S}'$. (Since $\mathcal{S}'$ is a PPTA, Q is some polynomial.) We first show that there exists a PPTA $\mathcal{F}$ such that $\mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai},\mathcal{F}}(k) \geq (1/Q(k)) \cdot \Pr[\mathsf{Bad}] - 1/|\mathcal{X}_k|$. The description of $\mathcal{F}$ is as follows:

$\mathcal{F}(1^k, z)$**:** (where $z \leftarrow \mathsf{ai}(\alpha, \beta)$, and α and β are chosen uniformly at random) $\mathcal{F}$ runs $\mathcal{S}'(1^k, z)$. $\mathcal{F}$ responds to all oracle queries from $\mathcal{S}'(1^k, z)$ (to $\mathcal{I}_{\alpha\to\beta}$) with $\bot$. When $\mathcal{S}'$ terminates, $\mathcal{F}$ picks one of the queries made by $\mathcal{S}'$ uniformly at random, outputs it as its guess for α, and terminates.

The above completes the description of $\mathcal{F}$. It is easy to see that $\mathcal{F}$ perfectly simulates the experiment $\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai},\mathcal{S}'}(k)$ for $\mathcal{S}'$ until the point $\mathcal{S}'$ makes the query α. Therefore, the probability that $\mathcal{S}'$ makes the query α is exactly $\Pr[\mathsf{Bad}]$. Furthermore, once $\mathcal{S}'$ makes the query α, it is picked by $\mathcal{F}$ with probability at least $1/Q$. Therefore, $\mathcal{F}$'s advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai},\mathcal{F}}(k) = \Pr[\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai},\mathcal{F}}(k) = 1] - \frac{1}{|\mathcal{X}_k|} \geq \frac{1}{Q(k)} \cdot \Pr[\mathsf{Bad}] - \frac{1}{|\mathcal{X}_k|}.$$

Now, recall that ai is a δ-$\mathtt{cPUAI}$ function and δ is negligible. and thus for all sufficiently large $k \in \mathbb{N}$, we have $\Pr[\mathsf{Bad}] \leq Q(k)(\delta(k) + 1/|\mathcal{X}_k|)$, and the right hand side is negligible. This completes the proof of Claim 14. □

Let $\mu(k)$ be a negligible function such that $\mu(k) = \Pr[\mathsf{Bad}]$. We have:

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{SA\text{-}MBPO\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A},\mathcal{S}}(k) &= |\Pr[\mathsf{Expt}^{\mathtt{Real}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai},\mathcal{S}}(k) = 1]| \\
&\leq |\Pr[\mathsf{Expt}^{\mathtt{Real}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai},\mathcal{S}'}(k) = 1]| \\
&\quad + |\Pr[\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai},\mathcal{S}'}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai},\mathcal{S}}(k) = 1]| \\
&\leq \frac{1}{2q(k)} + \mu(k) \\
&\leq \frac{1}{q(k)} \qquad (9)
\end{aligned}$$

where the first inequality follows from the triangle inequality, the second inequality holds for all sufficiently large $k \in \mathbb{N}$ due to the inequalities (7), (8) and Claim 14, and the last inequality holds for all sufficiently large $k \in \mathbb{N}$ (because $\mu(k) \leq \frac{1}{2q(k)}$ holds for all sufficiently large $k \in \mathbb{N}$). Recall that the choice of ai was arbitrarily, and thus the inequality (9) holds for any ℓ-bounded δ-$\mathtt{cPUAI}$ fuction ai.

Recall also that the choice of the PPTA adversary $\mathcal{A}$ and the positive polynomials q and ℓ was also arbitrarily, and thus the above works for any PPTA $\mathcal{A}$ and any positive polynomials q and ℓ. This means that MBPO is $\mathtt{SAVB}$-δ-$\mathtt{cPUAI}$ secure.

The implication from $\mathtt{AVB}$-δ-$\mathtt{sPUAI}$ security to $\mathtt{SAVB}$-δ-$\mathtt{sPUAI}$ security can be proved identically to the above.

The proof for showing the implication from $\mathtt{AVG}$-δ-$\mathtt{sPUAI}$ security to $\mathtt{SAVG}$-δ-$\mathtt{sPUAI}$ security also proceeds in almost the same way, and thus we omit it. The difference is that the simulator $\mathcal{S}'$ due to $\mathtt{AVG}$-δ-$\mathtt{sPUAI}$ security is computationally unbounded (and makes only polynomially many queries). Correspondingly, however, we only need to construct a computationally unbounded simulator $\mathcal{S}$, to show the $\mathtt{SAVG}$-δ-$\mathtt{sPUAI}$ security. Furthermore, the proof that $\mathcal{S}$ can simulate the experiment

$\mathsf{Expt}^{\mathtt{Sim}}_{\mathsf{ai},\mathcal{S}'}(k)$ for all δ-$\mathtt{sPUAI}$ secure functions follows from the fact that $\mathcal{S}'$ makes only polynomially many queries and ai is statistically partially uninvertible, and thus we can similarly derive the upperbound of the probability $\Pr[\mathsf{Bad}]$ to be negligible by constructing a computationally unbounded inverter $\mathcal{F}$ for ai. The rest of the analysis is exactly the same. This completes the proof of Lemma 5. □

D.5 Proof of Lemma 6

Let δ be any negligible function, and let MBPO be an MBPF obfuscator for $\mathsf{MBPF}(\mathcal{X}, t)$. Since the proof is essentially the same for both δ-$\mathtt{cPUAI}$ and δ-$\mathtt{sPUAI}$ cases, below we only show that $\mathtt{SAVG}$-δ-$\mathtt{cPUAI}$ security implies $\mathtt{AIND}$-δ-$\mathtt{cPUAI}$ security.

Fix arbitrarily a PPTA adversary $\mathcal{A}$ and a δ-$\mathtt{cPUAI}$ function ai. We will show that $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k)$ is negligible, namely, for any positive polynomial q and for all sufficiently large $k \in \mathbb{N}$, it holds that $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) < 1/q(k)$. To this end, fix any positive polynomial $q = q(k)$. Let $\ell = \ell(k)$ be the maximum length of the output of ai when it takes an input from $\mathcal{X}_k \times \{0,1\}^t$. Note that since ai is efficiently computable, ℓ must be some polynomial. Consider the following slightly modified probabilistic function $\mathsf{ai}' : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^\ell$:

$\mathsf{ai}'(\alpha, \beta)$**:** Pick $\beta' \in \{0,1\}^t$ uniformly at random, and return $z \leftarrow \mathsf{ai}(\alpha, \beta')$. (Here, ai' ignores the input β.)

Note that when α and β are chosen uniformly at random, the distribution of z output from $\mathsf{ai}(\alpha, \beta)$ and that from $\mathsf{ai}'(\alpha, \beta)$ are identical. This directly implies that if ai is a δ-$\mathtt{cPUAI}$ function, then so is ai'. (And if ai is a δ-$\mathtt{sPUAI}$ function, then so is ai'.) Furthermore, it also implies that for any (even computationally unbounded) algorithm $\mathcal{M}$, it holds that

$$\Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai},\mathcal{M}}(k) = 1] = \Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai}',\mathcal{M}}(k) = 1]. \tag{10}$$

Note also that by definition, both ai and ai' are ℓ-bounded.

Now, due to our assumption that MBPO is $\mathtt{SAVG}$-δ-$\mathtt{cPUAI}$ secure, for the adversary $\mathcal{A}$ and the polynomial $4q$ and the polynomial ℓ, there exists a computationally unbounded simulator $\mathcal{S}$ such that for the ℓ-bounded δ-$\mathtt{cPUAI}$ functions ai and ai', and for all sufficiently large $k \in \mathbb{N}$, the following two inequalities simultaneously hold:

$$\mathsf{Adv}^{\mathtt{SA\text{-}MBPO\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A},\mathcal{S}}(k) \le \frac{1}{4q(k)} \quad \text{and} \quad \mathsf{Adv}^{\mathtt{SA\text{-}MBPO\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}',\mathcal{A},\mathcal{S}}(k) \le \frac{1}{4q(k)}.$$

Let $\nu(k) = \Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai},\mathcal{S}}(k) = 1]$. Then by the equation (10), $\nu(k) = \Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai}',\mathcal{S}}(k) = 1]$ holds. Using this, the above two inequalities can be rewritten as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{SA\text{-}MBPO\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A},\mathcal{S}}(k) &= |\Pr[\mathsf{Expt}^{\mathtt{Real}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai},\mathcal{S}}(k) = 1]| \\ &= |\Pr[\mathcal{A}(1^k, \mathsf{ai}(\alpha,\beta), \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta})) = 1] - \nu(k)| \le \frac{1}{4q(k)} \end{aligned} \tag{11}$$

$$\begin{aligned}\mathsf{Adv}^{\mathtt{SA\text{-}MBPO\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}',\mathcal{A},\mathcal{S}}(k) &= |\Pr[\mathsf{Expt}^{\mathtt{Real}}_{\mathsf{MBPO},\mathsf{ai}',\mathcal{A}}(k) = 1] - \Pr[\mathsf{Expt}^{\mathtt{s\text{-}Sim}}_{\mathsf{ai}',\mathcal{S}}(k) = 1]| \\ &= |\Pr[\mathcal{A}(1^k, \mathsf{ai}(\alpha,\beta'), \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta})) = 1] - \nu(k)| \le \frac{1}{4q(k)} \end{aligned} \tag{12}$$

where the probabilities (in the right hand side of the above equalities) are over the choice of α, β, and β' uniformly at random and also over the choice of randomness consumed by Obf, ai, $\mathcal{A}$, and $\mathcal{S}$.

Now, we show that the adversary $\mathcal{A}$'s $\texttt{AIND-AI}$ advantage (with respect to the probabilistic function ai) is smaller than $1/q(k)$. For all sufficiently large $k \in \mathbb{N}$, it holds that

$$\begin{aligned}
\mathsf{Adv}^{\texttt{AIND-AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) &= 2 \cdot |\Pr[\mathsf{Expt}^{\texttt{AIND-AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k) = 1] - 1/2| \\
&= |\Pr[\mathcal{A}(1^k, \mathsf{ai}(\alpha,\beta_0), \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_0})) = 1] - \Pr[\mathcal{A}(1^k, \mathsf{ai}(\alpha,\beta_0), \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_1})) = 1]| \\
&\leq |\Pr[\mathcal{A}(1^k, \mathsf{ai}(\alpha,\beta_0), \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_0})) = 1] - \nu(k)| \\
&\quad + |\nu(k) - \Pr[\mathcal{A}(1^k, \mathsf{ai}(\alpha,\beta_0), \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_1})) = 1]| \\
&\leq \frac{1}{4q(k)} + \frac{1}{4q(k)} < \frac{1}{q(k)}, \qquad (13)
\end{aligned}$$

where the probabilities are over the random choice of α, β_0, β_1 uniformly at random and also over the choice of randomness consumed by MBPO, ai, and $\mathcal{A}$. In the above, the first inequality is due to the triangle inequality, and in the second inequality we used the inequality (11) (in which we regard β as β_0) and the inequality (12) (in which we regard β and β' as β_1 and β_0, respectively).

Recall that the choice of the positive polynomial q was arbitrarily, and thus for all positive polynomials q we can show the inequation (13), which implies that $\mathsf{Adv}^{\texttt{AIND-AI}}_{\mathsf{MBPO},\mathsf{ai},\mathcal{A}}(k)$ is negligible. Recall also that the choice of $\mathcal{A}$ and ai was also arbitrarily, and thus the above works for any PPTA $\mathcal{A}$ and any δ-$\texttt{cPUAI}$ function ai. This completes the proof of Lemma 6. □

D.6 Proof of Theorem 3

Theorem 3 is shown by using the equivalence of t-composability and t-distributional indistinguishability for coordinate-wise well-spread (CWS) distributions, established by Bitansky and Canetti [8].

First, we recall the notion of coordinate-wise well-spread distributions.

Definition 9. *([8]) Let $t = t(k) > 0$ be a polynomial. Let $\mathcal{X} = \{\mathcal{X}_k\}_{k\in\mathbb{N}}$ be a domain ensemble, and $\Phi = \{\Phi_k\}_{k\in\mathbb{N}}$ be an ensemble of distributions, where each Φ_k is a distribution over $(\mathcal{X}_k)^t$. We say that Φ is* t-coordinate-wise well-spread (CWS)*, if*

$$\max_{a\in\mathcal{X}_k} \Pr_{(\alpha_1,\ldots,\alpha_t)\leftarrow\Phi_k}[\exists i \in [t] : \alpha_i = a]$$

is negligible in k, where the probability is over the choice of $(\alpha_1, \ldots, \alpha_t) \in (\mathcal{X}_k)^t$ according the distribution Φ_k.

Then, we recall the definition of t-distributional indistinguishability for point obfuscators.

Definition 10. *([8]) Let $t = t(k) > 0$ be a polynomial. Let $\mathcal{X} = \{\mathcal{X}_k\}_{k\in\mathbb{N}}$ be an ensemble of domains, and let PO be a point obfuscator for $\mathsf{PF}(\mathcal{X})$. We say that PO satisfies* t-distributional indistinguishability *if for any t-CWS distribution ensemble $\Phi = \{\Phi_k\}_{k\in\mathbb{N}}$ (over $\mathcal{X}$) and for all PPTAs $\mathcal{A}$, $\mathsf{Adv}^{t\text{-}\texttt{DI}}_{\mathsf{PO},\Phi,\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{t\text{-}\texttt{DI}}_{\mathsf{PO},\Phi,\mathcal{A}}(k) = 1] - 1/2|$ is negligible, where the experiment $\mathsf{Expt}^{t\text{-}\texttt{DI}}_{\mathsf{PO},\mathcal{A}}(k)$ is defined as follows:*

$$\mathsf{Expt}^{t\text{-}\texttt{DI}}_{\mathsf{PO},\Phi,\mathcal{A}}(k) : [\ (\alpha_1^{(1)}, \ldots, \alpha_t^{(1)}) \leftarrow \Phi_k;\ (\alpha_1^{(0)}, \ldots, \alpha_t^{(0)}) \leftarrow (\mathcal{X}_k)^t;\ b \leftarrow \{0,1\};$$
$$\mathtt{P}_i^* \leftarrow \mathsf{PO}(\mathcal{I}_{\alpha_i^{(b)}}) \text{ for } i \in [t];\ b' \leftarrow \mathcal{A}(1^k, \{\mathtt{P}_i^*\}_{i\in[t]});\ \text{Return } (b' \stackrel{?}{=} b)\ ].$$

Bitansky and Canetti [8] showed the following:

Lemma 9. *([8]) Let $t = t(k) > 0$ be a polynomial, let $\mathcal{X} = \{\mathcal{X}_k\}_{k\in\mathbb{N}}$ be a domain ensemble, and let PO be a point obfuscator for $\mathsf{PF}(\mathcal{X})$. If PO is t-composable, then PO satisfies t-distributional indistinguishability.*

Now, we are ready to prove Theorem 3.

Proof of Theorem 3. Let PO be the point obfuscator for $\mathsf{PF}(\mathcal{X})$ used as a building block of Π. We first show that an ordinary public key output by $\mathsf{PKG}(1^k)$ and a lossy public key output by $\mathsf{LKG}(1^k)$ are indistinguishable.

Let $\mathcal{A}$ be any PPTA distinguisher that tries to distinguish an ordinary public key and a lossy public key of the scheme Π. Define the following three distribution ensembles $\Phi^{(1)} = \{\Phi_k^{(1)}\}_{k\in\mathbb{N}}$, $\Phi^{(2)} = \{\Phi_k^{(2)}\}_{k\in\mathbb{N}}$, and $\Phi^{(3)} = \{\Phi_k^{(3)}\}_{k\in\mathbb{N}}$:

$$\begin{aligned}
\Phi_k^{(1)} &= \{\alpha_0 \leftarrow \mathcal{X}_k;\ \alpha_1 \leftarrow \mathcal{X}_k\backslash\{\alpha_0\}\ :\ (\alpha_0,\alpha_1)\ \}\\
\Phi_k^{(2)} &= \{\alpha_0 \leftarrow \mathcal{X}_k;\ \alpha_1 \leftarrow \mathcal{X}_k :\ (\alpha_0,\alpha_1)\ \}\\
\Phi_k^{(3)} &= \{\alpha \leftarrow \mathcal{X}_k :\ (\alpha,\alpha)\ \}
\end{aligned}$$

Note that $\Phi^{(3)}$ is a 2-CWS distribution ensemble.

Using the above notation, we can estimate the advantage of $\mathcal{A}$ as follows:

$$\begin{aligned}
&\mathsf{Adv}^{\mathtt{KEY}}_{\Pi,\mathcal{A}}(k) = 2\cdot|\Pr[\mathsf{Expt}^{\mathtt{KEY}}_{\Pi,\mathcal{A}}(k)=1] - \frac{1}{2}|\\
&= |\Pr[\mathsf{Expt}^{\mathtt{KEY}}_{\Pi,\mathcal{A}}(k)=1|b=0] - \Pr[\mathsf{Expt}^{\mathtt{KEY}}_{\Pi,\mathcal{A}}(k)=1|b=1]|\\
&= |\Pr[(pk,sk)\leftarrow\mathsf{PKG}(1^k) : \mathcal{A}(pk)=1] - \Pr[pk\leftarrow\mathsf{LKG}(1^k):\mathcal{A}(pk)=1]|\\
&= |\Pr[(\alpha_0,\alpha_1)\leftarrow\Phi_k^{(1)} : \mathcal{A}(\mathsf{PO}(\mathcal{I}_{\alpha_0}),\mathsf{PO}(\mathcal{I}_{\alpha_1}))=1] - \Pr[(\alpha_0,\alpha_1)\leftarrow\Phi_k^{(3)} : \mathcal{A}(\mathsf{PO}(\mathcal{I}_{\alpha_0}),\mathsf{PO}(\mathcal{I}_{\alpha_1}))=1]|\\
&\le |\Pr[(\alpha_0,\alpha_1)\leftarrow\Phi_k^{(1)} : \mathcal{A}(\mathsf{PO}(\mathcal{I}_{\alpha_0}),\mathsf{PO}(\mathcal{I}_{\alpha_1}))=1] - \Pr[(\alpha_0,\alpha_1)\leftarrow\Phi_k^{(2)} : \mathcal{A}(\mathsf{PO}(\mathcal{I}_{\alpha_0}),\mathsf{PO}(\mathcal{I}_{\alpha_1}))=1]|\\
&\quad + |\Pr[(\alpha_0,\alpha_1)\leftarrow\Phi_k^{(2)} : \mathcal{A}(\mathsf{PO}(\mathcal{I}_{\alpha_0}),\mathsf{PO}(\mathcal{I}_{\alpha_1}))=1] - \Pr[(\alpha_0,\alpha_1)\leftarrow\Phi_k^{(3)} : \mathcal{A}(\mathsf{PO}(\mathcal{I}_{\alpha_0}),\mathsf{PO}(\mathcal{I}_{\alpha_1}))=1]|
\end{aligned} \tag{14}$$

(where the probabilities are also over the randomness consumed by PO and $\mathcal{A}$.)

Note that the statistical distance between $\Phi_k^{(1)}$ and $\Phi_k^{(2)}$ is negligible, and thus the first term in the inequality (14) is negligible. Furthermore, our assumption that PO is 2-composable, combined with Lemma 9, implies that PO satisfies 2-distributional indistinguishability. This, combined with fact that $\Phi^{(3)}$ is 2-CWS, in turn implies that the second term in the inequality (14) is negligible. (Otherwise, we can construct a PPTA distinguisher that violates 2-distributional indistinguishability of PO with regard to $\Phi^{(3)}$.) In summary, Π satisfies the indistinguishability of ordinary/lossy public keys.

The fact that Π satisfies the statistical lossiness is straightforward to see. A lossy public key output by $\mathsf{LKG}(1^k)$ is of the form $pk = (\widehat{\mathsf{P}}_0, \widehat{\mathsf{P}}_1) = (\mathsf{PO}(\mathcal{I}_\alpha), \mathsf{PO}(\mathcal{I}_\alpha))$ for a randomly chosen $\alpha \in \mathcal{X}_k$, and thus for any plaintext $m \in \{0,1\}^t$, its encryption is of the form $C = (c_1,\ldots,c_t)$ where every c_i is computed as $c_i \leftarrow \mathsf{ReRand}(\mathsf{PO}(\mathcal{I}_\alpha))$, and thus the distribution of a ciphertext is identical for all plaintexts. (This remains true even when the distribution of public keys is also taken into account.) This implies that even a computationally unbounded adversary $\mathcal{A}$ has advantage zero in the experiment $\mathsf{Expt}^{\mathtt{LOS\text{-}CPA}}_{\Pi,\mathcal{A}}(k)$. This completes the proof of Theorem 3. □

E Non-adaptive Chosen Ciphertext Security via MBPF Obfuscation

In this section, we show a simpler `CCA1` secure variant of our proposed KEMs that we showed in Section 4. Interestingly, the relation between the KEM Γ' shown in this section and our first

$\mathsf{KKG}'(1^k)$:	$\mathsf{Encap}'(PK)$:	$\mathsf{Decap}'(SK, C)$:
$(pk_i, sk_i) \leftarrow \mathsf{PKG}(1^k)$ for $i \in [2]$	Parse PK as (pk_1, pk_2)	Parse SK as (sk_1, sk_2)
$PK \leftarrow (pk_1, pk_2)$	$\alpha \leftarrow \mathcal{X}_k$	Parse C as $(c_1, c_2, \mathtt{DL})$
$SK \leftarrow (sk_1, sk_2)$	$\beta \leftarrow \{0,1\}^t$	$\alpha \leftarrow \mathsf{Dec}(sk_1, c_1)$
Return (PK, SK)	$\mathtt{DL} \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta})$	If $\alpha = \perp$ then reutrn $\perp$
	Parse β as (r_1, r_2, K)	$\beta \leftarrow \mathtt{DL}(\alpha)$
	$\in (\{0,1\}^{\ell_{\mathtt{R}}})^2 \times \{0,1\}^k$	If $\beta = \perp$ then return $\perp$
	$c_i \leftarrow \mathsf{Enc}(pk_i, \alpha; r_i)$ for $i \in [2]$	Parse β as (r_1, r_2, K)
	$C \leftarrow (c_1, c_2, \mathtt{DL})$	$\in (\{0,1\}^{\ell_{\mathtt{R}}})^2 \times \{0,1\}^k$
	Return (C, K)	If $\forall i \in [2] : \mathsf{Enc}(pk_i, \alpha; r_i) = c_i$
		then return K else return $\perp$

Fig. 6. The proposed CCA1 secure KEM Γ'.

construction Γ in Section 4 is similar to the relation between the Naor-Yung PKE construction [63] and the Dolev-Dwork-Naor PKE construction [35].

Let $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with the plaintext space $\{0,1\}^k$ and the randomness length $\ell_{\mathtt{R}}(k)$. We define $t(k) = 2\ell_{\mathtt{R}}(k) + k$. Let $\mathcal{X} = \{\mathcal{X}_k\}_{k\in\mathbb{N}}$ be a domain ensemble such that each element in $\mathcal{X}_k$ is of length k, and let MBPO be an MBPF obfuscator for $\mathsf{MBPF}(\mathcal{X}, t)$. Then we construct a KEM $\Gamma' = (\mathsf{KKG}', \mathsf{Encap}', \mathsf{Decap}')$ as in Fig. 6.

As in the case of the CCA2 secure KEM Γ given in Section 4, the following "alternative" decapsulation algorithm $\mathsf{AltDecap}'$ is useful for showing the security of Γ': For a key pair $(PK, SK = (sk_1, sk_2))$ output by $\mathsf{KKG}'(1^k)$, we define the "alternative" secret key $\widehat{SK}$ by $\widehat{SK} = (PK, sk_2)$. $\mathsf{AltDecap}'$ takes an alternative key $\widehat{SK}$ and a ciphertext $C = (c_1, c_2, \mathtt{DL})$ as input, and runs as follows:

$\mathsf{AltDecap}'(\widehat{SK}, C)$**:** On input an alternative secret key $\widehat{SK} = (PK, sk_2)$ and a ciphertext $C = (c_1, c_2, \mathtt{DL})$, $\mathsf{AltDecap}'$ runs in exactly the same way as $\mathsf{Decap}'(SK, C)$, except that it executes $\mathsf{Dec}(sk_2, c_2)$ in the third step, instead of $\mathsf{Dec}(sk_1, c_1)$.

Regarding $\mathsf{AltDecap}'$, the following is easy to see due to the symmetric role of sk_1 and sk_2, and the validity check of c_1 and c_2 performed at the last step. (The proof is essentially the same as that of Lemma 3, and thus omitted.)

Lemma 10. *Let $(PK, SK = (sk_1, sk_2))$ be a key pair output by $\mathsf{KKG}'(1^k)$, and let $\widehat{SK} = (PK, sk_2)$ be the alternative secret key as defined above. Then, for any ciphertext C (which could be outside the range of $\mathsf{Encap}'(PK)$), it holds that $\mathsf{Decap}'(SK, C) = \mathsf{AltDecap}'(\widehat{SK}, C)$.*

Now, we show the CCA1 security of Γ' as follows.

Theorem 5. *Assume that Π is ϵ-CPA secure with negligible ϵ and MBPO is AIND-δ-cPUAI secure with $\delta(k) \geq 2\epsilon(k)$. Then, the KEM Γ' constructed as in Fig. 6 is CCA1 secure.*

Proof of Theorem 5. We will show that for any PPTA adverasry $\mathcal{A}$ attacking the CCA1 security of the KEM Γ', there exist PPTA $\mathcal{B}_o$ and a $(k\epsilon)$-cPUAI function $\mathsf{ai}_{\Gamma'} : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^*$ such that

$$\mathsf{Adv}^{\mathtt{CCA1}}_{\Gamma',\mathcal{A}}(k) \leq 2 \cdot \mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_{\Gamma'},\mathcal{B}_o}(k). \tag{15}$$

This inequality, combined with our assumptions on the building blocks, implies that $\mathsf{Adv}^{\mathtt{CCA1}}_{\Gamma',\mathcal{A}}(k)$ is negligible, and proves the theorem.

Fix arbitrarily a CCA1 adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against Γ'. 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 $\mathsf{Expt}^{\mathtt{CCA1}}_{\Gamma',\mathcal{A}}(k)$ itself. Without loss of generality, we generate/choose the challenge ciphertext $C^* = (c_1^*, c_2^*, \mathtt{DL}^*)$ and the challenge session-key K_b^*, where b is the challenge bit for $\mathcal{A}$, before running $\mathcal{A}_1$. (Note that this does not affect $\mathcal{A}$'s behavior.)

Game 2: Same as Game 1, except that $\mathtt{DL}^*$ is replaced with an obfuscation of the MBPF $\mathcal{I}_{\alpha^*\to\beta'}$ with an independently chosen random value $\beta' \in \{0,1\}^t$. That is, the step "$\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha^*\to\beta^*})$" is replaced with the steps "$\beta' \leftarrow \{0,1\}^t$; $\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha^*\to\beta'})$." (Note that r_1^*, r_2^*, and K_1^* are still generated from β^*.)

For $i \in [2]$, let Succ_i be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game i. Using the notation, $\mathcal{A}$'s $\mathtt{CCA1}$ advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{CCA1}}_{\Gamma',\mathcal{A}}(k) = 2 \cdot |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \le 2 \cdot |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| + 2 \cdot |\Pr[\mathsf{Succ}_2] - \frac{1}{2}|. \tag{16}$$

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

Firstly, we would like to consider the upperbound of $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$. To this end, we need to use the $\mathtt{AIND}$-δ-$\mathtt{cPUAI}$ security of MBPO. We therefore specify the auxiliary input function that we are going to consider. Define the probabilistic function $\mathsf{ai}_{\Gamma'} : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^*$ that takes $(\alpha, \beta) \in \mathcal{X}_k \times \{0,1\}^t$ as input, and computes $z = (\mathsf{st}, c_1^*, c_2^*, K^*) \in \{0,1\}^*$ in the following way:

$$\mathsf{ai}_{\Gamma'}(\alpha,\beta) : [\ (PK = (pk_1, pk_2), SK) \leftarrow \mathsf{KKG}'(1^k);\ \text{Parse } \beta \text{ as } (r_1^*, r_2^*, K^*) \in (\{0,1\}^{\ell_R})^2 \times \{0,1\}^k;$$
$$c_i^* \leftarrow \mathsf{Enc}(pk_i, \alpha; r_i^*) \text{ for } i \in [2];\ \mathsf{st} \leftarrow \mathcal{A}_1^{\mathsf{Decap}'(SK,\cdot)}(PK);\ \text{Return } z \leftarrow (\mathsf{st}, c_1^*, c_2^*, K^*)\ ].$$

where the randomness used by $\mathsf{ai}_{\Gamma'}$ is the randomness for executing KKG' and that for executing $\mathcal{A}_1$. Note that $\mathsf{ai}_{\Gamma'}$ uses $\mathcal{A}_1$ as a subroutine and the output of $\mathsf{ai}_{\Gamma'}$ contains the state information st that is supposed to be passed to $\mathcal{A}_2$, and thus the output length of $\mathsf{ai}_{\Gamma'}$ cannot be a-priori bounded, as opposed to the case to our proposed construction Γ. Note also that $\mathsf{ai}_{\Gamma'}$ is efficiently computable. In particular, although $\mathcal{A}_1$ needs to be given access to the decapsulation oracle $\mathsf{Decap}'(SK,\cdot)$, the secret key SK is generated during the process of computing $\mathsf{ai}_{\Gamma'}$, and once the oracle $\mathsf{Decap}'(SK,\cdot)$ is given, $\mathcal{A}_1$ can be computed efficiently. In order to use $\mathsf{ai}_{\Gamma'}$ as an auxiliary input function corresponding to an $\mathtt{AIND}$-$\mathtt{AI}$ adversary against the MBPF obfuscator MBPO, we state the following claim, which will be proven in the end of the proof of this theorem.

Claim 15 *$\mathsf{ai}_{\Gamma'}$ is a (2ϵ)-$\mathtt{cPUAI}$ function.*

We proceed to showing the upperbound of $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$.

Claim 16 *There exists a PPTA $\mathcal{B}_o$ such that $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_{\Gamma'},\mathcal{B}_o}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$.*

Proof of Claim 16. We show how to construct a PPTA $\mathcal{B}_o$ with the claimed advantage. $\mathcal{B}_o$ is given as input 1^k, $z = (\mathsf{st}, c_1^*, c_2^*, K^*) \leftarrow \mathsf{ai}_{\Gamma'}(\alpha, \beta_0)$, and $\mathtt{DL}^*$ which is output from either $\mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_0})$ or $\mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_1})$, where $\alpha \in \mathcal{X}_k$ and $\beta_0, \beta_1 \in \{0,1\}^t$ are chosen randomly, and runs as follows:

$\mathcal{B}_o(1^k, z, \mathtt{DL}^*)$: $\mathcal{B}_o$ first parses z as $(\mathsf{st}, c_1^*, c_2^*, K^*)$. Then, $\mathcal{B}_o$ picks $\gamma \in \{0,1\}$ and $K_0^* \in \{0,1\}^k$ uniformly at random, sets $C^* \leftarrow (c_1^*, c_2^*, \mathtt{DL}^*)$ and $K_1^* \leftarrow K^*$, runs $\gamma' \leftarrow \mathcal{A}_2(\mathsf{st}, C^*, K_\gamma^*)$, and terminates with output $b' \leftarrow (\gamma' \stackrel{?}{=} \gamma)$.

The above completes the description of $\mathcal{B}_o$. Let b be the challenge bit for $\mathcal{B}_o$. $\mathcal{B}_o$'s $\mathtt{AIND}$-$\mathtt{AI}$ advantage can be estimated as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_{\Gamma'},\mathcal{B}_o}(k) &= 2 \cdot |\Pr[b' = b] - \frac{1}{2}| = |\Pr[b' = 1 | b = 0] - \Pr[b' = 1 | b = 1]| \\ &= |\Pr[\gamma' = \gamma | b = 0] - \Pr[\gamma' = \gamma | b = 1]|.\end{aligned}$$

Consider the case when $b = 0$, i.e. $\mathtt{DL}^*$ is computed as $\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_0})$. Note that by the definition of the AIND-AI experiment and the function $\mathsf{ai}_{\Gamma'}$, if we regard α and β_0 in $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_{\Gamma'},\mathcal{B}_o}(k)$ as α^* and β^* in Game 1, respectively, then the values in z (i.e. $\mathcal{A}$'s state information st, the ciphertexts c_1^* and c_2^*, and the value K^* that is used as K_1^*), are generated/chosen in exactly the same way as those in Game 1. In particular, $\mathcal{A}_1$ is run during the calculation of $\mathsf{ai}_{\Gamma'}$, but it is given a correctly generated public key $PK = (pk_1, pk_2)$ as input, and $\mathcal{A}_1$'s decapsulation queries C are answered with $\mathsf{Decap}'(SK, C)$, as is done in Game 1, and thus the state information st is generated in exactly the same way as that in Game 1. Furthermore, since γ is chosen randomly by $\mathcal{B}_o$, the challenge ciphertext $C^* = (c_1^*, c_2^*, \mathtt{DL}^*)$ and the challenge session-key K_γ^* that are input into $\mathcal{A}_2$ are also generated in exactly the same way as those in Game 1 in which the challenge bit is γ. Since $\mathcal{B}_o$ inputs $(\mathsf{st}, C^*, K_\gamma^*)$ to $\mathcal{A}_2$, $\mathcal{B}_o$ simulates Game 1 perfectly for $\mathcal{A}_2$ in which the challenge bit for $\mathcal{A}$ is γ. Under this situation, the probability that $\gamma' = \gamma$ occurs is exactly the same as the probability that $\mathcal{A}$ succeeds in guessing its challenge bit in Game 1, i.e. $\Pr[\gamma' = \gamma | b = 0] = \Pr[\mathsf{Succ}_1]$.

Next, consider the case when $b = 1$, i.e. $\mathtt{DL}^*$ is computed as $\mathtt{DL}^* \leftarrow \mathsf{MBPO}(\mathcal{I}_{\alpha\to\beta_1})$, where β_1 is also chosen uniformly at random, independently of β_0. Under this situation, if we regard α, β_0, and β_1 in $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_{\Gamma'},\mathcal{B}_o}(k)$ as α^*, β^*, and β' in Game 2, respectively, then the challenge ciphertext/session-key pair (C^*, K_γ^*) is generated in such a way that it is distributed identically to that in Game 2, and thus $\mathcal{B}_o$ simulates Game 2 perfectly for $\mathcal{A}$ in which the challenge bit for $\mathcal{A}$ is γ. Therefore, with a similar argument to the above, we have $\Pr[\gamma' = \gamma | b = 1] = \Pr[\mathsf{Succ}_2]$.

In summary, we have $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{\mathsf{MBPO},\mathsf{ai}_{\Gamma'},\mathcal{B}_o}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$. This completes the proof of Claim 16. □

Claim 17 $\Pr[\mathsf{Succ}_2] = 1/2$.

Proof of Claim 17. This is obvious because K_1^*, which is contained in β^*, is independent of the challenge ciphertext C^*, and the distribution of K_1^* and that of K_0^* are exactly the same in Game 2. Therefore, the distribution of the challenge ciphertext/session-key pair (C^*, K_b^*) as well as all other values (public key PK and the responses to decapsulation queries) are identically distributed in both cases $b = 0$ and $b = 1$, which implies $\Pr[\mathsf{Succ}_2] = 1/2$. This completes the proof of Claim 17. □

Claims 16 and 17, and the inequality (16) guarantee that there exist a PPTA $\mathcal{B}_o$ and a (2ϵ)-cPUAI function $\mathsf{ai}_{\Gamma'}$ satisfying the inequality (15), as required. Recall that the choice of the PPTA CCA1 adversary $\mathcal{A}$ was arbitrarily, and thus this for any PPTA CCA1 adversary we can show its negligible advantage. Hence, Γ' is CCA1 secure.

It remains to prove Claim 15.

Proof of Claim 15. As we have mentioned, $\mathsf{ai}_{\Gamma'}$ is efficiently computable. Thus, in the following we show that $\mathsf{ai}_{\Gamma'}$ is (2ϵ)-computationally partially uninvertible.

Let $\mathcal{F}$ be an arbitrary PPTA adversary that runs in $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_{\Gamma'},\mathcal{F}}(k)$. For $i \in [3]$, let $\mathsf{ai}_i : \mathcal{X}_k \times \{0,1\}^t \to \{0,1\}^*$ be the probabilistic function defined as follows:

ai_1**:** This is $\mathsf{ai}_{\Gamma'}$ itself. (This is introduced for notational convenience.)

ai_2**:** Same as ai_1, except that c_2^* is generated as $c_2^* \leftarrow \mathsf{Enc}(pk_2, 0^k; r_2^*)$, instead of $c_2^* \leftarrow \mathsf{Enc}(pk_2, \alpha; r_2^*)$.

ai_3**:** Same as ai_2, except that all decryption queries C from $\mathcal{A}_1$ are answered with $\mathsf{AltDecap}'(\widehat{SK}, C)$, instead of $\mathsf{Decap}'(SK, C)$, where $\widehat{SK}$ is the alternative secret key corresponding to the key pair (PK, SK).

For notational convenience, for $i \in [3]$, let $p_i = \Pr[\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_i,\mathcal{F}}(k) = 1]$. By the triangle inequality, we have

$$\mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_{\Gamma'},\mathcal{F}}(k) = p_1 - \frac{1}{|\mathcal{X}_k|} \leq \sum_{i\in[2]} |p_i - p_{i+1}| + p_3 - \frac{1}{|\mathcal{X}_k|}. \tag{17}$$

In the following we show the upperbound of the terms that appear in the right hand side of the above inequality.

Subclaim 1 *There exists a PPTA $\mathcal{B}_p$ such that $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}_p}(k) = |p_1 - p_2|$.*

Proof of Subclaim 1. We show how to construct a PPTA $\mathtt{CPA}$ adversary $\mathcal{B}_p$ that attacks the PKE scheme Π with the claimed advantage. The description of $\mathcal{B}_p = (\mathcal{B}_{p1}, \mathcal{B}_{p2})$ is as follows:

$\mathcal{B}_{p1}(pk)$**:** $\mathcal{B}_{p1}$ picks $\alpha \in \mathcal{X}_k$ uniformly at random, and sets $M_0 \leftarrow \alpha$ and $M_1 \leftarrow 0^k$. Then $\mathcal{B}_{p1}$ prepares the state information $\mathsf{st}_{\mathcal{B}}$ consisting of all information known to $\mathcal{B}_{p1}$, and terminates with output $(M_0, M_1, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_{p2}(\mathsf{st}_{\mathcal{B}}, c^*)$**:** $\mathcal{B}_{p,2}$ first runs $(pk_1, sk_1) \leftarrow \mathsf{PKG}(1^k)$, sets $pk_2 \leftarrow pk$, $PK \leftarrow (pk_1, pk_2)$, and $SK \leftarrow (sk_1, \perp)$. $\mathcal{B}_{p2}$ also picks $K^* \in \{0,1\}^k$ uniformly at random, executes $c_1^* \leftarrow \mathsf{Enc}(pk_1, \alpha)$, and sets $c_2^* \leftarrow c^*$. $\mathcal{B}_{p2}$ then runs $\mathsf{st} \leftarrow \mathcal{A}_1^{\mathsf{Decap}'(SK,\cdot)}(PK)$. (Note that the knowledge of sk_2 is not needed to run $\mathsf{Decap}'(SK, C)$, and thus $\mathcal{B}_{p2}$ can perform this step.) Then, $\mathcal{B}_{p2}$ sets $z \leftarrow (\mathsf{st}, c_1^*, c_2^*, K^*)$, and runs $\alpha' \leftarrow \mathcal{F}(1^k, z)$. Finally, $\mathcal{B}_{p2}$ terminates with output $b' \leftarrow (\alpha' \stackrel{?}{=} \alpha)$.

The above completes the description of $\mathcal{B}_p$. Let b be the challenge bit for $\mathcal{B}_p$. $\mathcal{B}_p$'s $\mathtt{CPA}$ advantage can be calculated as follows:

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

Consider the case when $b = 0$, i.e. c^* is an encryption of $M_0 = \alpha$. It is easy to see that in this case, $\mathcal{B}_p$ simulates $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_1,\mathcal{F}}(k)$ perfectly for $\mathcal{F}$. In particular, in the real experiment $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_1,\mathcal{F}}(k)$, c_1^* and c_2^* are computed using randomness r_1^* and r_2^* that are contained in $\beta \in \{0,1\}^t$, which are chosen uniformly at random, and c_1^* and c_2^* generated in $\mathcal{B}_p$'s $\mathtt{CPA}$ experiment are also generated by using the randomness which are identically distributed (the randomness for c_1^* is chosen by $\mathcal{B}_p$ and the randomness for c_2^* is chosen by $\mathcal{B}_p$'s $\mathtt{CPA}$ experiment). Under this situation, the probability that $\alpha' = \alpha$ occurs is identical to the probability that $\mathcal{F}$ succeeds in outputting α in $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_1,\mathcal{F}}(k)$, i.e. $\Pr[\alpha' = \alpha|b = 0] = p_1$.

On the other hand, it is also easy to see that when $b = 1$ (i.e. c^* is an encryption of $M_1 = 0^k$), $\mathcal{B}_p$ simulates $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_2,\mathcal{F}}(k)$ perfectly for $\mathcal{F}$. With a similar argument to the above, we have $\Pr[\alpha' = \alpha|b = 1] = p_2$.

In summary, we have $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}_p}(k) = |p_1 - p_2|$. This completes the proof of Subclaim 1. □

Subclaim 2 $p_2 = p_3$.

Proof of Subclaim 2. Note that the difference between the experiments $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_2,\mathcal{F}}(k)$ and $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_3,\mathcal{F}}(k)$ is whether we use $\mathsf{Decap}'(SK,\cdot)$ or $\mathsf{AltDecap}'(\widehat{SK},\cdot)$ for answering the decryption queries from $\mathcal{A}_1$, where $\widehat{SK}$ is the alternative secret key. However, Lemma 10 tells us that for any (possibly invalid) ciphertext C, we have $\mathsf{Decap}'(SK, C) = \mathsf{AltDecap}'(\widehat{SK}, C)$. Therefore, from the viewpoint of $\mathcal{A}_1$,

these two oracles behave identically, and hence the view of $\mathcal{A}_1$ is distributed identically in both experiments. This means that the value z in both experiments is distributed identically, and thus $\mathcal{F}$ outputs α with exactly the same probability. This completes the proof of Subclaim 2. □

Subclaim 3 *There exists a PPTA $\mathcal{B}'_p$ such that $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}'_p}(k) = p_3 - 1/|\mathcal{X}_k|$.*

Proof of Subclaim 3. We show how to construct a PPTA $\mathtt{CPA}$ adversary $\mathcal{B}'_p$ that attacks the PKE scheme Π with the claimed advantage. The description of $\mathcal{B}'_p = (\mathcal{B}'_{p1}, \mathcal{B}'_{p2})$ is as follows:

$\mathcal{B}'_{p1}(pk)$**:** $\mathcal{B}'_{p1}$ picks $\alpha \in \mathcal{X}_k$ uniformly at random, and sets $M_0 \leftarrow \alpha$ and $M_1 \leftarrow 0^k$. Then $\mathcal{B}'_{p1}$ prepares the state information $\mathsf{st}_{\mathcal{B}}$ consisting of all information known to $\mathcal{B}'_{p1}$, and terminates with output $(M_0, M_1, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}'_{p2}(\mathsf{st}_{\mathcal{B}}, c^*)$**:** $\mathcal{B}'_{p2}$ first runs $(pk_2, sk_2) \leftarrow \mathsf{PKG}(1^k)$, sets $pk_1 \leftarrow pk$, $PK \leftarrow (pk_1, pk_2)$, and $\widehat{SK} \leftarrow (PK, sk_2)$. $\mathcal{B}'_{p2}$ also picks $K^* \in \{0,1\}^k$ uniformly at random, executes $c_2^* \leftarrow \mathsf{Enc}(pk_2, 0^k)$, and sets $c_1^* \leftarrow c^*$. $\mathcal{B}'_{p2}$ then runs $\mathsf{st} \leftarrow \mathcal{A}_1^{\mathsf{AltDecap}'(\widehat{SK},\cdot)}(PK)$, sets $z \leftarrow (\mathsf{st}, c_1^*, c_2^*, K^*)$, and runs $\alpha' \leftarrow \mathcal{F}(1^k, z)$. Finally, $\mathcal{B}'_{p2}$ terminates with output $b' \leftarrow (\alpha' \stackrel{?}{=} \alpha)$.

The above completes the description of $\mathcal{B}'_p$. Let b be the challenge bit for $\mathcal{B}'_p$. $\mathcal{B}'_p$'s $\mathtt{CPA}$ advantage can be calculated as follows:

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

Consider the case when $b = 0$, i.e. c^* is an encryption of $M_0 = \alpha$. It is easy to see that in this case, $\mathcal{B}'_p$ simulates $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_3,\mathcal{F}}(k)$ perfectly for $\mathcal{F}$. Under this situation, the probability that $\alpha' = \alpha$ occurs is identical to the probability that $\mathcal{F}$ succeeds in outputting α in $\mathsf{Expt}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_3,\mathcal{F}}(k)$, i.e. $\Pr[\alpha' = \alpha|b = 0] = p_3$.

On the other hand, note that when $b = 1$ (i.e. c^* is an encryption of $M_1 = 0^k$), the value z does not contain any information on α. (In particular, in this case both c_1^* and c_2^* are an encryption of 0^k.) Therefore, the view of $\mathcal{F}$ is independent of α. This must mean that the probability that $\mathcal{F}$ succeeds in outputting α is exactly $1/|\mathcal{X}_k|$. That is, we have $\Pr[\alpha' = \alpha|b = 1] = 1/|\mathcal{X}_k|$.

In summary, we have $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}'_p}(k) = p_3 - 1/|\mathcal{X}_k|$. (Here, without loss of generality we use $p_3 \geq 1/|\mathcal{X}_k|$.) This completes the proof of Subclaim 3. □

Subclaims 1 to 3 and the inequality (17) guarantee that there exist PPTAs $\mathcal{B}_p$ and $\mathcal{B}'_p$ such that

$$\mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_{\Gamma'},\mathcal{F}}(k) \leq \mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}_p}(k) + \mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}'_p}(k).$$

Here, since the PKE scheme Π is ϵ-$\mathtt{CPA}$ secure, for all sufficiently large $k \in \mathbb{N}$, we have $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}_p}(k) \leq \epsilon(k)$ and $\mathsf{Adv}^{\mathtt{CPA}}_{\Pi,\mathcal{B}'_p}(k) \leq \epsilon(k)$.

In summary, for all sufficiently large $k \in \mathbb{N}$ we have $\mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_{\Gamma'},\mathcal{F}}(k) \leq 2\epsilon(k)$. Recall that the choice of $\mathcal{F}$ was arbitrarily, and thus for all PPTAs $\mathcal{F}$ we can show that $\mathsf{Adv}^{\mathtt{P\text{-}Inv}}_{\mathsf{ai}_{\Gamma'},\mathcal{F}}(k) \leq \epsilon(k)$ holds for all sufficiently large $k \in \mathbb{N}$. Therefore, $\mathsf{ai}_{\Gamma'}$ is (2ϵ)-computationally partially uninvertible. This completes the proof of Claim 15. □

This completes the proof of Theorem 5. □

F On Replacing MBPF Obfuscators with SKE

In this section, we discuss several issues on replacing an MBPF obfuscator with a SKE scheme in our proposed constructions of KEMs.

Motivation. As has been clarified in several previous works [23, 33, 44, 25], there is a strong connection between MBPF obfuscators and SKE schemes. More specifically, an obfuscation of an MBPF $\mathcal{I}_{K\to m}$ can be considered as an encryption of a plaintext m using a key K. In order for the converse direction to hold, i.e. viewing an encryption of a plaintext m under a key K (together with a decryption algorithm) as an obfuscation of the MBPF $\mathcal{I}_{K\to m}$, the SKE scheme needs to satisfy a property that the decryption algorithm using a key K outputs $\perp$ when it is input a ciphertext generated by using a different key $K' \neq K$. This property is referred to as the *unique-key property*[5] [33, 44, 25] (we recall the definition in Appendix A).

Given a SKE scheme that has the unique-key property and satisfies the security that we call AIND-δ-cPUAI security and AIND-δ-sPUAI security, which are defined in essentially the same way as those for MBPF obfuscators, then we can replace the MBPF obfuscator in our proposed KEM Γ in Section 4 with the SKE scheme. (Interestingly, the unique-key property is not needed for our CCA1 secure construction given in Appendix E, and thus we can replace the MBPF obfuscator with a SKE without the unique-key property.)

Although it was shown in [25] how to convert any SKE scheme into one that supports the unique-key property by using a family of pairwise-independent hash functions, we could not figure out whether this conversion preserves AIND-δ-cPUAI security and AIND-δ-sPUAI security in general. (We conjecture that it does not.)

Therefore, we think that in general AIND-δ-cPUAI secure (and AIND-δ-sPUAI secure) SKE schemes without the unique-key property are potentially easier to construct than MBPF obfuscators with the same security. This is the main reason why we focus on replacing an MBPF obfuscator with a SKE scheme in our proposed constructions. In the following we will show that the unique-key property is not a necessary property for constructing CCA2 secure KEMs by using a SKE scheme, via a tag-based KEM (TBKEM for short, whose formal definition can be found in Appendix A) together with the TBKEM-to-PKE/KEM transformation due to [54].

Organization of This Section. The rest of this section is organized as follows: In Appendix F.1, we formalize the AIND-δ-cPUAI security and the AIND-δ-sPUAI security for SKE, and we also recall the transformation of an MBPF obfuscator into a SKE scheme. In Appendix F.2, we show that the proposed KEM using a SKE scheme can be modified to a wCCA secure TBKEM, and show that the security of this TBKEM can be proved without relying on the unique-key property of the used SKE scheme. Finally, in Appendix F.3, we show an evidence that achieving AIND-δ-cPUAI and AIND-δ-sPUAI secure SKE schemes is not so trivial, by clarifying an implication of these security definitions to a kind of leakage-resilience security that takes into account leakage only from a key (as opposed to leakage simultaneously from a key and a random plaintext being encrypted).

F.1 Average-Case Indistinguishability with Auxiliary Input for SKE

We formalize the security notion that we need for a SKE scheme here.

Definition 11. *We say that a SKE scheme $E = (\mathsf{SEnc}, \mathsf{SDec})$ (whose plaintext space is $\{0,1\}^t$ for some polynomial $t = t(k) > 0$) satisfies average-case indistinguishability w.r.t. δ-computationally*

[5] This property was called *wrong-key detection* in [44, 25].

*(resp. δ-statistically) partially uninvertible auxiliary input (*AIND-δ-cPUAI *(resp.* AIND-δ-sPUAI*) security, for short) if for all PPTAs $\mathcal{A}$ and all δ-*cPUAI *(resp. δ-*sPUAI*) functions* ai*, the advantage function* $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{A}}(k) = 1] - 1/2|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{A}}(k)$ *is defined as follows:*

$$\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{A}}(k) : [K \leftarrow \{0,1\}^k;\ m_0, m_1 \leftarrow \{0,1\}^t;\ z \leftarrow \mathsf{ai}(K, m_0);\ b \leftarrow \{0,1\}$$
$$c^* \leftarrow \mathsf{SEnc}(K, m_b);\ b' \leftarrow \mathcal{A}(1^k, z, c^*);\ \text{Return } (b' \stackrel{?}{=} b)].$$

From MBPF Obfuscator to SKE. Here, we recall the transformation of an MBPF obfuscator into a SKE scheme [23, 25].

Let MBPO be an MBPF obfuscator for $\mathsf{MBPF}(\{0,1\}^*, t)$. From MBPO, we construct a SKE scheme $E = (\mathsf{SEnc}, \mathsf{SDec})$ as follows:

$\mathsf{SEnc}(K, m)$**:** On input a key $K \in \{0,1\}^k$ and a plaintext $m \in \{0,1\}^t$, return a ciphertext $c \leftarrow \mathsf{MBPO}(\mathcal{I}_{K \to m})$.

$\mathsf{SDec}(K, c)$**:** On input a key $K \in \{0,1\}^k$ and a ciphertext c, interpret c as a circuit and return $m \leftarrow c(K)$.

The following is obvious from the definition.

Lemma 11. *If the MBPF obfuscator* MBPO *is* AIND-δ-cPUAI *(resp.* AIND-δ-sPUAI*) secure, then the SKE scheme E constructed as above is* AIND-δ-cPUAI *(resp.* AIND-δ-sPUAI*) secure and furthermore has the unique-key property.*

F.2 wCCA Secure TBKEM via SKE

As we explained in Section 7 and at the beginning of Appendix F, if we replace the MBPF obfuscator in our proposed KEMs in Section 4 with a SKE scheme, then the SKE scheme needs to satisfy unique-key property. (However, for our CCA1 secure KEM in Appendix E, we do not need it.)

Here, we show that we can construct a CCA2 secure PKE scheme/KEM without relying on the unique-key property. This can be accomplished by first constructing a wCCA secure TBKEM and then converting it into a CCA2 secure PKE/KEM using the existing methods [30, 24, 54] using a one-time signature (or a combination of a commitment and a message authentication code [13]).

Our construction of a TBKEM is a simple modification of the KEM Γ given in Section 4. Other than using a SKE scheme instead of an MBPF obfuscator, the difference between the proposed TBKEM and the KEM Γ is that we use a tag that is input to the encapsulation algorithm of a TBKEM as a "selector" of public keys $\{pk_i^{(j)}\}_{i \in [k], j \in \{0,1\}}$, instead of using a hash value $h = \mathsf{H}_\kappa(\widetilde{c})$.

Formally, the construction of the TBKEM is as follows: Let $\Pi = (\mathsf{PKG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with the plaintext space $\{0,1\}^k$ and the randomness length $\ell_{\mathtt{R}}(k)$. We define $t(k) = k \cdot \ell_{\mathtt{R}}(k) + k$. Let $E = (\mathsf{SEnc}, \mathsf{SDec})$ be a SKE schemes with the plaintext space $\{0,1\}^t$. Then we construct a TBKEM $\mathcal{T} = (\mathsf{TKG}, \mathsf{TEncap}, \mathsf{TDecap})$ as in Fig. 7. (We assume that the tag space of $\mathcal{T}$ is $\{0,1\}^k$ when used with the security parameter 1^k.)

The security of $\mathcal{T}$ is guaranteed by the following theorems. As in the proposed KEM Γ in Section 4, wCCA security of $\mathcal{T}$ can be shown in two ways.

Theorem 6. *Assume that Π is ϵ-*CPA *secure with negligible ϵ, and E is* AIND-δ-cPUAI *secure with $\delta(k) \geq k\epsilon(k)$. Then, the TBKEM $\mathcal{T}$ constructed as in Fig. 7 is* wCCA *secure.*

Theorem 7. *Assume that Π is an ϵ-lossy encryption scheme with negligible ϵ, and E is* AIND-δ-sPUAI *secure with $\delta(k) \geq k\epsilon(k)$. Then, the TBKEM $\mathcal{T}$ constructed as in Fig. 7 is* wCCA *secure.*

$\mathsf{TKG}(1^k)$:	$\mathsf{TEncap}(PK, \mathsf{tag})$:	$\mathsf{TDecap}(SK, \mathsf{tag})$:
$(pk_i^{(j)}, sk_i^{(j)}) \leftarrow \mathsf{PKG}(1^k)$ for $i \in [k]$ and $j \in \{0,1\}$	Parse PK as $\{pk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}$	Parse SK as $\{sk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}$.
$PK \leftarrow \{pk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}$	$\alpha \leftarrow \{0,1\}^k$	Parse C as $(c_1, \ldots, c_k, \widetilde{c})$.
$SK \leftarrow \{sk_i^{(j)}\}_{i\in[k],j\in\{0,1\}}$	$\beta \leftarrow \{0,1\}^t$	View tag as $(t_1 \| \ldots \| t_k) \in \{0,1\}^k$.
Return (PK, SK)	$\widetilde{c} \leftarrow \mathsf{SEnc}(\alpha, \beta)$	$\alpha \leftarrow \mathsf{Dec}(sk_1^{(t_1)}, c_1)$
	View tag as $(t_1 \| \ldots \| t_k) \in \{0,1\}^k$.	If $\alpha = \perp$ then return $\perp$.
	Parse β as $(r_1, \ldots, r_k, K) \in (\{0,1\}^{\ell_R})^k \times \{0,1\}^k$	$\beta \leftarrow \mathsf{SDec}(\alpha, \widetilde{c})$
	$c_i \leftarrow \mathsf{Enc}(pk_i^{(t_i)}, \alpha; r_i)$ for $i \in [k]$	If $\beta = \perp$ then return $\perp$.
	$C \leftarrow (c_1, \ldots, c_k, \widetilde{c})$	Parse β as $(r_1, \ldots, r_k, K) \in (\{0,1\}^{\ell_R})^k \times \{0,1\}^k$
	Return (C, K)	If $\forall i \in [k] : \mathsf{Enc}(pk_i^{(t_i)}, \alpha; r_i) = c_i$ then return K else return $\perp$

Fig. 7. The proposed wCCA secure TBKEM $\mathcal{T}$ using SKE (possibly without the unique-key property).

The proofs of Theorems 6 and 7 are almost identical to those of Theorems 1 and 2, and thus we only provide a proof sketch of the former theorem.

Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ be any PPTA wCCA adversary. Consider the following sequence of games.

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

Game 2: Same as Game 1, except that all queries (tag, C) are answered with $\mathsf{AltTDecap}(\widehat{SK}_{\mathsf{tag}^*}, C)$, where $\mathsf{AltTDecap}$ is the alternative decapsulation algorithm that is defined similarly to $\mathsf{AltDecap}$ for the KEM Γ given in Section 4 and $\widehat{SK}_{\mathsf{tag}^*}$ is the alternative secret key, which is defined similarly to that for $\mathsf{AltDecap}$, corresponding to (PK, SK) and the k-bit string tag^* which is the challenge tag submitted by $\mathcal{A}_0$.

Game 3: Same as Game 2, except that $\widetilde{c}^*$ is replaced with an encryption of an independently chosen random value $\beta' \in \{0,1\}^t$. That is, the step "$\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, \beta^*)$" is replaced with the steps "$\beta' \leftarrow \{0,1\}^t$; $\widetilde{c}^* \leftarrow \mathsf{SEnc}(\alpha^*, \beta')$."

For $i \in [3]$, let Succ_i be the event that $\mathcal{A}$ succeeds in guessing the challenge bit in Game i.

We can show that $\Pr[\mathsf{Succ}_1] = \Pr[\mathsf{Succ}_2]$, because $\mathsf{TDecap}(SK, \mathsf{tag}, C) = \mathsf{AltTDecap}(\widehat{SK}_{\mathsf{tag}^*}, \mathsf{tag}, C)$ holds for all tag/ciphertext pairs (tag, C) satisfying $\mathsf{tag} \neq \mathsf{tag}^*$ and thus the oracle given to $\mathcal{A}$ in Game 1 and that in Game 2 behave identically for all queries from $\mathcal{A}$.

We can show $|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$ to be negligible using the AIND-δ-cPUAI security of E and the ϵ-CPA security of Π. In showing this, we consider an auxiliary input function that is exactly the same as ai_Γ we used in the proof of Theorem 1.

Finally, $\Pr[\mathsf{Succ}_3] = 1/2$ because $\mathcal{A}$'s view is independent of the challenge bit. (More specifically, K_1^* and K_0^* are both chosen uniformly from $\{0,1\}^k$ and independent of the challenge ciphertext C^* (and any other values that are available for $\mathcal{A}$ in Game 3), and thus the challenge session-key is distributed identically regardless of the challenge bit.)

F.3 On the Non-triviality for Achieving AIND-δ-cPUAI and AIND-δ-sPUAI Security

In this subsection, we show an evidence that constructing an AIND-δ-cPUAI (and AIND-δ-sPUAI) secure SKE scheme is at least as difficult as constructing a SKE scheme which satisfies one-time security in the presence of hard-to-invert auxiliary input that captures the *leakage only from a key* (which has been considered in several papers [33, 44, 25]). This is done by showing that it is possible to construct a SKE scheme satisfying the latter security from a SKE scheme satisfying the former security.

We first define the latter security notion for a SKE scheme. To this end, we need to define an appropriate auxiliary input function that captures a leakage from a key, which we simply call an *uninvertible auxiliary input function*.

Definition 12. *Let* $\widehat{\mathsf{ai}} : \{0,1\}^k \to \{0,1\}^*$ *be a (possibly probabilistic) function. We say that* $\widehat{\mathsf{ai}}$ *is a* δ-computationally (resp. δ-statistically) uninvertible auxiliary input function (δ-`cUAI` (resp. δ-`sUAI`) function, for short)*, if (1) it is efficiently computable, and (2) for every PPTA (resp. computationally unbounded algorithm)* $\mathcal{F}$ *and for all sufficiently large* $k \in \mathbb{N}$*, it holds that* $\mathsf{Adv}^{\mathtt{Inv}}_{\widehat{\mathsf{ai}},\mathcal{F}}(k) := \Pr[\mathsf{Expt}^{\mathtt{Inv}}_{\widehat{\mathsf{ai}},\mathcal{F}}(k) = 1] - 2^{-k} \leq \delta(k)$*, where the experiment* $\mathsf{Expt}^{\mathtt{Inv}}_{\widehat{\mathsf{ai}},\mathcal{F}}(k)$ *is defined as follows:*

$$\mathsf{Expt}^{\mathtt{Inv}}_{\widehat{\mathsf{ai}},\mathcal{F}}(k) : [\ K \leftarrow \{0,1\}^k;\ z \leftarrow \widehat{\mathsf{ai}}(K);\ K' \leftarrow \mathcal{F}(1^k, z);\ \text{Return } (K' \stackrel{?}{=} K)\ ].$$

We then define the one-time security in the presence of uninvertible auxiliary input.

Definition 13. *We say that a SKE scheme* $E = (\mathsf{SEnc}, \mathsf{SDec})$ *satisfies* indistinguishability under one-time encryption in the presence of δ-computationally (resp. δ-statistically) uninvertible auxiliary input ($\mathtt{OT}$-δ-`cUAI` (resp. $\mathtt{OT}$-δ-`sUAI`) secure, for short) *if for all PPTAs* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *and all* δ-`cUAI` *(resp.* δ-`sUAI`*) functions* $\widehat{\mathsf{ai}}$*, the advantage function* $\mathsf{Adv}^{\mathtt{OT\text{-}AI}}_{E,\widehat{\mathsf{ai}},\mathcal{A}}(k) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathtt{OT\text{-}AI}}_{E,\widehat{\mathsf{ai}},\mathcal{A}}(k) = 1] - 1/2|$ *is negligible, where the experiment* $\mathsf{Expt}^{\mathtt{OT\text{-}AI}}_{E,\widehat{\mathsf{ai}},\mathcal{A}}(k)$ *is defined as follows:*

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

Relations to the Existing Definitions. Our definition of $\mathtt{OT}$-δ-`cUAI` security is the "one-time encryption" version of [33, Definition 4.1]. (Actually, [33] only considers exponentially computationally uninvertible functions as auxiliary input functions).

Furthermore, our definition of $\mathtt{OT}$-δ-`cUAI`/$\mathtt{OT}$-δ-`sUAI` security is similar to the "*semantic security with weak keys and auxiliary inputs*" in [26, Def. 4.1], but is weaker in two aspects. First, our definition only considers keys that are chosen uniformly at random, while the definition in [26] treats "weak keys" that are only guaranteed to have some sufficiently high min-entropy. Second, our definition only requires ordinary indistinguishability (of an encryption of two challenge plaintexts), while the definition in [26] requires the existence of a universal simulator that can generate a ciphertext that is indistinguishable from an honestly generated ciphertext for any adversarially chosen plaintext (the simulator needs to be universal in the sense that it needs to work for all PPTA adversaries).

Constructing $\mathtt{OT}$-δ-`cUAI` *and* $\mathtt{OT}$-δ-`sUAI` *Secure SKE Schemes.* Let $E = (\mathsf{SEnc}, \mathsf{SDec})$ and $E' = (\mathsf{SEnc}', \mathsf{SDec}')$ be SKE schemes, where we assume that the plaintext space of E and that of E' are $\{0,1\}^k$ and $\{0,1\}^t$ (where $t = t(k) > 0$ is a polynomial), respectively. Then we construct another SKE scheme $\widetilde{E} = (\widetilde{\mathsf{SEnc}}, \widetilde{\mathsf{SDec}})$ with the plaintext space $\{0,1\}^t$ as in Fig. 8.

The following shows that constructing an $\mathtt{AIND}$-δ-`cPUAI` (resp. $\mathtt{AIND}$-δ-`sPUAI`) secure SKE scheme is at least as difficult as constructing an $\mathtt{OT}$-δ-`cUAI` (resp. $\mathtt{OT}$-δ-`sUAI`) secure one.

Theorem 8. *If the SKE scheme* E *is* $\mathtt{AIND}$-δ-`cPUAI` *(resp.* $\mathtt{AIND}$-δ-`sPUAI`*) secure and the SKE scheme* E' *is* $\mathtt{OT}$ *secure, then the SKE scheme* $\widetilde{E}$ *constructed as in Fig. 8 is* $\mathtt{OT}$-δ-`cUAI` *(resp.* $\mathtt{OT}$-δ-`sUAI`*) secure.*

$\widetilde{\mathsf{SEnc}}(K, m)$:	$\widetilde{\mathsf{SDec}}(K, C)$:
$R \leftarrow \{0,1\}^k$	Parse C as (c_1, c_2)
$c_1 \leftarrow \mathsf{SEnc}(K, R)$	$R \leftarrow \mathsf{SDec}(K, c_1)$
$c_2 \leftarrow \mathsf{SEnc}'(R, m)$	If $R = \perp$ then return $\perp$
Return $C \leftarrow (c_1, c_2)$	Return $m \leftarrow \mathsf{SDec}'(R, c_2)$

Fig. 8. The construction of an OT-δ-cUAI (resp. OT-δ-sUAI) secure SKE scheme $\widetilde{E}$ from an AIND-δ-cPUAI (resp. AIND-δ-sPUAI) secure SKE scheme E and an OT secure SKE scheme E'.

Proof of Theorem 8. The security proofs for the both cUAI and sUAI cases are essentially the same, and thus we only show the former case.

We will show that for any PPTA adversary $\mathcal{A}$ attacking the OT-δ-cUAI security of the SKE scheme $\widetilde{E}$ and any δ-cUAI function $\widehat{\mathsf{ai}}$, there exist PPTAs $\mathcal{B}$ and $\mathcal{B}'$ and a δ-cPUAI function ai such that

$$\mathsf{Adv}^{\mathtt{OT\text{-}AI}}_{\widetilde{E},\widehat{\mathsf{ai}},\mathcal{A}}(k) \leq 2 \cdot \mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{B}}(k) + \mathsf{Adv}^{\mathtt{OT}}_{E',\mathcal{B}'}(k), \tag{18}$$

which, combined with the assumptions on E and E', implies that $\mathsf{Adv}^{\mathtt{OT\text{-}AI}}_{\widetilde{E},\widehat{\mathsf{ai}},\mathcal{A}}(k)$ is negligible, and proves the theorem.

To this end, fix arbitrarily a PPTA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ attacking the OT-δ-cPUAI security of $\widetilde{E}$ and a δ-cPUAI function $\widehat{\mathsf{ai}}$. Consider the following games: (The values with asterisk represent those related to the challenge ciphertext for $\mathcal{A}$.)

Game 1: This is the experiment $\mathsf{Expt}^{\mathtt{OT\text{-}AI}}_{\widetilde{E},\widehat{\mathsf{ai}},\mathcal{A}}(k)$ itself.

Game 2: Same as Game 1, except that c_2^* is generated by using an independently chosen random key $R' \in \{0,1\}^k$. That is, the step "$c_2^* \leftarrow \mathsf{SEnc}'(R^*, m_b)$" is replaced with the steps "$R' \leftarrow \{0,1\}^k$; $c_2^* \leftarrow \mathsf{SEnc}'(R', m_b)$," where b is the challenge bit for $\mathcal{A}$. (Note that c_1^* still encrypts R^*.)

For $i \in [2]$, let Succ_i be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game i. Using the above notation, $\mathcal{A}$'s advantage can be calculated as follows:

$$\mathsf{Adv}^{\mathtt{OT\text{-}AI}}_{\widetilde{E},\widehat{\mathsf{ai}},\mathcal{A}}(k) = 2 \cdot |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \leq 2 \cdot |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| + 2 \cdot |\Pr[\mathsf{Succ}_2] - \frac{1}{2}|. \tag{19}$$

Therefore, it remains to upperbound the right hand side of the above inequality.

We first would like to show the upperbound of $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$, using the AIND-$\delta$-cPUAI security of E. To this end, we first specify the δ-cPUAI function $\mathsf{ai} : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}^*$ as follows:

$\mathsf{ai}(K, R)$**:** On input $K, R \in \{0,1\}^k$, compute $z' \leftarrow \widehat{\mathsf{ai}}(K)$ and output $z \leftarrow (z', R)$.

It is straightforward to see that the above function is a δ-cPUAI function, because $\widehat{\mathsf{ai}}$ is a δ-cUAI function and R is independent of z'. In fact, it is easy to construct an inverter for $\widehat{\mathsf{ai}}$, given an inverter for ai. Furthermore, here, it is also easy to see that if $\widehat{\mathsf{ai}}$ is a δ-sUAI function, then ai is a δ-sPUAI function.

Now, we proceed to showing the upperbound of the right hand side of the inequality (19).

Claim 18 *There exists a PPTA $\mathcal{B}$ such that* $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{B}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$.

Proof of Claim 18. Consider a PPTA adversary $\mathcal{B}$ that takes an auxiliary input $z = (z', R_0) \leftarrow \mathsf{ai}(K, R_0)$ and a challenge ciphertext $c_1^* \leftarrow \mathsf{SEnc}(K, R_b)$ as input (where $K, R_0, R_1 \in \{0,1\}^k$ are chosen uniformly at random, and b is the challenge bit for $\mathcal{B}$), and runs as follows:

$\mathcal{B}(1^k, z = (z', R_0), c^*)$**:** $\mathcal{B}$ runs $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(1^k, z')$. Then $\mathcal{B}$ picks a fair coin $\gamma \in \{0,1\}$, computes $c_2^* \leftarrow \mathsf{SEnc}'(R_0, m_\gamma)$ sets $c_1^* \leftarrow c^*$ and $C^* \leftarrow (c_1^*, c_2^*)$, and then runs $\gamma' \leftarrow \mathcal{A}_2(\mathsf{st}, C^*)$. Finally, $\mathcal{B}$ terminates with output $b' \leftarrow (\gamma' \stackrel{?}{=} \gamma)$.

The above completes the description of $\mathcal{B}$. $\mathcal{B}$'s AIND-AI advantage can be calculated as follows:

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

Consider the case when $b = 0$. Note that by definition, if we regard K in the experiment $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{B}}(k)$ as the key K^* under which $\mathcal{A}$'s challenge ciphertext is generated in Game 1, then $\mathcal{B}$ always inputs a correct auxiliary input z' output from $\widehat{\mathsf{ai}}(K)$. If we furthermore regard R_0 in the experiment $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{B}}(k)$ as R^* in Game 1, then the distribution of the challenge ciphertext $C^* = (c_1^*, c_2^*)$ is identical to that generated in Game 1 in which the challenge bit for $\mathcal{A}$ is γ (in particular, c_1^* is computed as $\mathsf{SEnc}(K, R_0)$ and c_2^* is computed as $\mathsf{SEnc}'(R_0, m_\gamma)$). Under the situation, the probability that $\gamma' = \gamma$ occurs is exactly the same as the probability that $\mathcal{A}$ succeeds in guessing the challenge bit in Game 1, i.e. $\Pr[\gamma' = \gamma | b = 0] = \Pr[\mathsf{Succ}_1]$.

When $b = 1$, on the other hand, $c^* = c_1^*$ is now an encryption of R_1 chosen independently of R_0, while still c_2^* is computed as $\mathsf{SEnc}'(R_0, m_\gamma)$. Under the situation, we can regard R_0 and R_1 in $\mathsf{Expt}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{B}}(k)$ as R^* and R' in Game 2, respectively, and thus $\mathcal{B}$ simulates Game 2 perfectly for $\mathcal{A}$ in which the challenge bit for $\mathcal{A}$ is γ. With a similar argument to the above, we have $\Pr[\gamma' = \gamma | b = 1] = \Pr[\mathsf{Succ}_2]$.

In summary, we have $\mathsf{Adv}^{\mathtt{AIND\text{-}AI}}_{E,\mathsf{ai},\mathcal{B}}(k) = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$. This completes the proof of Claim 18. □

Claim 19 *There exists a PPTA $\mathcal{B}'$ such that* $\mathsf{Adv}^{\mathtt{OT}}_{E',\mathcal{B}'}(k) = 2 \cdot |\Pr[\mathsf{Succ}_2] - 1/2|$.

Proof of Claim 19. We show how to construct a PPTA adversary $\mathcal{B}'$ that attacks the SKE scheme E' with the claimed advantage. The description of $\mathcal{B}' = (\mathcal{B}'_1, \mathcal{B}'_2)$ is as follows:

$\mathcal{B}'_1(1^k)$**:** $\mathcal{B}'_1$ first picks a random value $K^* \in \{0,1\}^k$, and computes $z' \leftarrow \widehat{\mathsf{ai}}(K^*)$. Then $\mathcal{B}'_1$ runs $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(1^k, z')$. $\mathcal{B}'_1$ then prepares the state information $\mathsf{st}_\mathcal{B}$ consisting of all information known to $\mathcal{B}'_1$, and terminates with output $(m_0, m_1, \mathsf{st}_\mathcal{B})$.

$\mathcal{B}'_2(\mathsf{st}_\mathcal{B}, c^*)$**:** $\mathcal{B}'_2$ picks $R^* \in \{0,1\}^k$ uniformly at random, computes $c_1^* \leftarrow \mathsf{SEnc}(K^*, R^*)$, sets $c_2^* \leftarrow c^*$ and $C^* \leftarrow (c_1^*, c_2^*)$, and runs $b' \leftarrow \mathcal{A}_2(\mathsf{st}, C^*)$. $\mathcal{B}'_2$ finally outputs this b' and terminates.

The above completes the description of $\mathcal{B}'$. Let b be the challenge bit for $\mathcal{B}'$. It is easy to see that $\mathcal{B}'$ perfectly simulates Game 2 for $\mathcal{A}$ in which the challenge bit for $\mathcal{A}$ is that of $\mathcal{B}$. Therefore, we have $\Pr[b' = b] = \Pr[\mathsf{Succ}_2]$, and thus $\mathsf{Adv}^{\mathtt{OT}}_{E',\mathcal{B}'}(k) = 2 \cdot |\Pr[b' = b] - 1/2| = 2 \cdot |\Pr[\mathsf{Succ}_2] - 1/2|$. This completes the proof of Claim 19 □

We have seen that there exist PPTAs $\mathcal{B}$ and $\mathcal{B}'$ and a δ-cPUAI function ai satisfying the inequality (18), as required. Since the choice of $\mathcal{A}$ and its corresponding δ-cUAI function $\widehat{\mathsf{ai}}$ was arbitrarily, the above proof works for any choice of PPTA $\mathcal{A}$ and a δ-cUAI function $\widehat{\mathsf{ai}}$. This means that $\widetilde{E}$ is OT-δ-cUAI secure. This completes the proof of Theorem 8. □

$\mathsf{MBPO}^{\mathcal{R}}(\mathcal{I}_{\alpha\to\beta})$:	$\mathcal{C}^{\mathcal{R}}_{r,X,Y}(x)$:
$r \leftarrow \{0,1\}^k$	$(X'\|K) \leftarrow \mathcal{R}(r\|x)$ where $\|X'\| = 2k$ and $\|K\| = t$
$(X\|K) \leftarrow \mathcal{R}(r\|\alpha)$ where $\|X\| = 2k$ and $\|K\| = t$	If $X' \neq X$ then return $\perp$.
$Y \leftarrow K \oplus \beta$	Return $\beta \leftarrow K \oplus Y$.
Return $\mathtt{DL}^{\mathcal{R}} \leftarrow \mathcal{C}^{\mathcal{R}}_{r,X,Y}(\cdot)$.	

Fig. 9. The MBPF obfuscator $\mathsf{MBPO}^{\mathcal{R}}$ in the random oracle model in [57]. $\mathsf{MBPO}^{\mathcal{R}}$ takes an MBPF $\mathcal{I}_{\alpha\to\beta}$ as input, and returns a circuit $\mathtt{DL}^{\mathcal{R}} = \mathcal{C}^{\mathcal{R}}_{r,X,Y}(\cdot)$ that is described in the right column.

G AIND-δ-cPUAI Secure MBPF Obfuscator in the Random Oracle Model

Here, we recall the MBPF obfuscator by Lynn, Prabhakaran, and Sahai [57] that uses a random oracle, and show that it can be shown to be AIND-δ-cPUAI secure (for any negligible function δ) in the random oracle model. (We note that in the random oracle model, not only an MBPF obfuscator, an obfuscated circuit, and an adversary, but also the corresponding δ-cPUAI function is allowed to access to the random oracle.)

Let $t = t(k) > 0$ be a polynomial and let $\mathcal{R} : \{0,1\}^* \to \{0,1\}^{2k+t}$ be a random oracle. Then, the MBPF obfuscator $\mathsf{MBPO}^{\mathcal{R}}$ in [57] for $\mathsf{MBPF}(\{0,1\}^*, t)$ is as in Fig. 9.

We note that this construction only satisfies the approximate functionality. That is, with a negligible probability (over the choice of the random oracle $\mathcal{R}$), $\mathsf{MBPO}^{\mathcal{R}}(\mathcal{I}_{\alpha\to\beta})$ can output an obfuscated circuit DL for which there exists a value $\alpha' \neq \alpha$ such that $\mathtt{DL}(\alpha') \neq \perp$. However, such approximate functionality is sufficient for our purpose in this paper. (Additionally, when viewed as a SKE scheme, it satisfies the unique-key property.)

The security can be shown as follows.

Theorem 9. *The MBPF obfuscator $\mathsf{MBPO}^{\mathcal{R}}$ constructed as in Fig. 9 is AIND-δ-cPUAI secure for any negligible function δ in the random oracle model where $\mathcal{R}$ is modeled as a random oracle.*

Proof of Theorem 9. Let $\mathcal{A}$ be any PPTA adversary and let $\mathsf{ai} : \{0,1\}^k \times \{0,1\}^t \to \{0,1\}^*$ be any δ-cPUAI function where δ is a negligible function, and let $\mathcal{R}$ be a random oracle. Suppose $\mathcal{A}$ and ai make at most $q_{\mathcal{A}}$ and q_{ai} queries, respectively. (Note that since $\mathcal{A}$ and ai are PPTAs, $q_{\mathcal{A}}$ and q_{ai} are polynomials.) Recall that in the experiment, $\mathcal{A}$ is given an auxiliary input $z \leftarrow \mathsf{ai}^{\mathcal{R}}(\alpha, \beta_0)$ and an obfuscated circuit $\mathtt{DL}^{\mathcal{R}} = \mathcal{C}^{\mathcal{R}}_{r,X,Y}(\cdot) \leftarrow \mathsf{MBPO}^{\mathcal{R}}(\mathcal{I}_{\alpha\to\beta_b})$ where $\mathcal{R}(r\|\alpha) = (X\|K)$ and $Y = (K \oplus \beta_b)$, is given oracle access to the random oracle $\mathcal{R}$, and has to guess the challenge bit b.

Firstly, note that the probability that $\mathsf{ai}^{\mathcal{R}}(\alpha, \beta_0)$ makes the query $(r\|\alpha)$ is negligible (actually it is at most $q_{\mathsf{ai}}/2^k$), because r is independent of the view of ai and r is chosen uniformly at random from $\{0,1\}^k$.

Secondly, conditioned on the event that ai has not made the query $(r\|\alpha)$, it is easy to see that the probability that $\mathcal{A}(z, \mathtt{DL})$ submits the query $(r\|\alpha)$ is negligible due to the partial uninvertibility of ai. More specifically, the probability can be shown to be at most $q_{\mathcal{A}} \cdot (\delta + 1/2^k) + q_{\mathsf{ai}}/2^k$. (If this does not hold, then one can construct an inverter for ai that is given $z \leftarrow \mathsf{ai}^{\mathcal{R}}(\alpha, \beta)$, uses $\mathcal{A}$ as a building block and succeeds in outputting α with the advantage greater than δ.)

Finally, conditioned on the event that neither ai nor $\mathcal{A}$ made the query $(r\|\alpha)$, K looks like a uniformly random string over $\{0,1\}^t$, and the information on the challenge bit b is information-theoretically hidden from $\mathcal{A}$, because $\mathcal{R}$ is a random oracle.

Therefore, by the union bound over the undesirable events that ai or $\mathcal{A}$ makes the query $(r\|\alpha)$, we can conclude that $\mathcal{A}$ has at most negligible advantage. This completes the proof of Theorem 9. □