Contention in Cryptoland: Obfuscation, Leakage and UCE

Mihir Bellare[1] Igors Stepanovs[2] Stefano Tessaro[3]

October 2014

Abstract

We study the achievability of different forms of obfuscation and related primitives $\mathbf{A}, \mathbf{B}$ through relations of the form $\mathbf{A} \Rightarrow \neg\mathbf{B}$ —this says that $\mathbf{A}, \mathbf{B}$ cannot both exist— or $\mathbf{A} \Rightarrow \mathbf{B}$ —this says that if $\mathbf{A}$ exists so does $\mathbf{B}$ or if $\mathbf{B}$ does not exist then neither does $\mathbf{A}$. Specifically: (1) We show that VGBO (Virtual Grey Box Obfuscation) for all circuits, which has been conjectured to be achieved by candidate constructions, would imply the failure of Canetti's 1997 AI-DHI1 (auxiliary input DH inversion) assumption and corresponding AIPO (Auxiliary-Input Point-function Obfuscation) scheme (2) We recover AIPO via a variant AI-DHI2 assumption, certain forms of UCE (Universal Computational Extractors), and a construction from any auxiliary-input OWF (3) We show that iO (indistinguishability Obfuscation) for all circuits implies the impossibility of certain forms of leakage-resilient encryption and other forms of UCE.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grants CNS-1116800 and CNS-1228890.

[2] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: istepano@eng.ucsd.edu. Supported in part by NSF grants CNS-1116800 and CNS-1228890.

[3] Department of Computer Science, University of California Santa Barbara, Santa Barbara, California 93106, USA. Email: tessaro@cs.ucsb.edu. URL: http://www.cs.ucsb.edu/~tessaro/. Supported in part by NSF grant CNS-1423566.

Contents

1 Introduction **3**

2 Preliminaries **7**

3 VGBO and the AI-DHI assumptions **8**

4 Achieving AIPO **12**

5 Impossibility results from iO **15**

A Standard definitions **20**

B Proof of Theorem 4.1 **22**

C Proof of Theorem 4.2 **23**

D PRIV1-AI-DPKE $\Rightarrow$ AIPO **24**

E Impossibility of KM-leakage-resilient encryption **25**

F Proof of Theorem 5.2 **28**

1 Introduction

Cryptographic theory is being increasing bold with regard to assumptions and conjectures. This is particularly true in the area of obfuscation, where candidate constructions have been provided whose claim to achieve a certain form of obfuscation is either itself an assumption [30] or is justified under other, new and strong assumptions [41, 9, 33]. This is attractive and exciting because we gain new capabilities and applications. But it behoves us also to be cautious and try to ascertain, not just whether the assumptions are true, but whether the goals are even achievable.

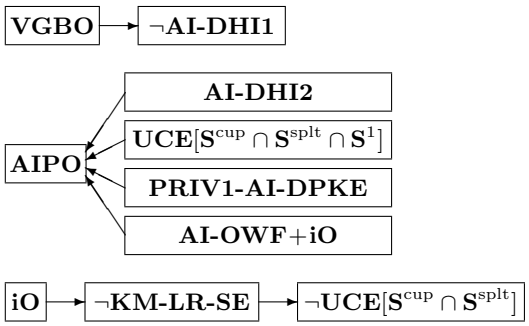
But how are we to determine this? The direct route is cryptanalysis, and we have indeed seen some success [38, 27, 32, 28]. But cryptanalysis can be difficult and runs into major open complexity-theoretic questions. There is another rewarding route, that we pursue here. This is to seek and establish relations, or connections, between different primitives and assumptions $\mathbf{A}, \mathbf{B}$ in the area. We can divide such relations into two types. First are "negative" relations, of the form $\mathbf{A} \Rightarrow \neg\mathbf{B}$. This shows that $\mathbf{A}, \mathbf{B}$ are not *both* achievable. We may not know which of the two fails, but we cannot have both, which is valuable and sometimes surprising information. Second are "positive" relations $\mathbf{A} \Rightarrow \mathbf{B}$, showing that if $\mathbf{A}$ is achievable then so is $\mathbf{B}$. This has double utility: If we believe $\mathbf{A}$ exists then we have increased confidence in the existence of $\mathbf{B}$, but if we believe $\mathbf{B}$ does not exist then we can conclude that $\mathbf{A}$ does not either.

Several prior works have used **iO** as a starting point to obtain intriguing negative relations $\mathbf{iO} \Rightarrow \neg\mathbf{B}$, for different choices of $\mathbf{B}$ [11, 18, 20, 19, 4, 35]. GGHW [31] use "special-purpose obfuscation" as a starting point to conclude $\neg\mathbf{diO}$. In this paper, we ask what is possible given a different starting point, namely **VGBO** (Virtual Grey Box Obfuscation) [8]. We show that VGBO rules out the AI-DHI1 assumption, and thus the primary existing construction of another form of obfuscation, AIPO. This leads us to seek and provide new constructions for AIPO. Finally we provide some new and improved negative relations with starting point iO. The results we obtain shed light on whether or not VGBO for all circuits is possible and further clarify which primitives or assumptions can co-exist. Our results are summarized in Fig. 1 and we now discuss them further.

VGBO vs AI-DHI1. We show that $\mathbf{VGBO} \Rightarrow \neg\mathbf{AI\text{-}DHI1}$. That is, if Virtual Grey Box Obfuscation (VGBO) of all circuits is possible then Canetti's 1997 AI-DHI1 (Auxiliary-Input DH Inversion) assumption [22, 12] fails. Equivalently, if the AI-DHI1 assumption is true, then VGBO obfuscation of all circuits is not achievable. We do not know which of the two fails, but at least one must. (Of course it could be that both fail.) Let us now back up to provide more information on the objects involved and the proof.

The study of obfuscation began with VBBO (Virtual Black Box Obfuscation) [36, 3], which asks that for any PT adversary $\mathcal{A}$ given the obfuscated circuit, there is a PT simulator $\mathcal{S}$ given an oracle for the original circuit, such that the two have about the same probability of returning 1. The impossibility of VBBO [3, 34, 13] has lead to efforts to define and achieve weaker forms of obfuscation. VGBO [8] is a natural relaxation of VBBO allowing the simulator $\mathcal{S}$ to be computationally unbounded but restricted to polynomially-many oracle queries. This bypasses known VBBO impossibility results while still allowing interesting applications. Furthermore BCKP [9, 10] show that VGBO for NC^1 is achievable (under a novel assumption). They then say "existing candidate indistinguishability obfuscators for all circuits [30, 16, 2] may also be considered as candidates for VGB obfuscation, for all circuits" [10, Section 1.1]. This would mean, in particular, that VGBO for all circuits is achievable. In this paper we ask if this *VGB conjecture* is true. Towards obtaining answers, we study the implications of the conjecture through relations.

The AI-DHI1 assumption [22, 12] says that there is an ensemble $\mathcal{G} = \{\mathbb{G}_\lambda : \lambda \in \mathbb{N}\}$ of prime-order groups such that, for r, s chosen at random from $\mathbb{G}_\lambda$, no polynomial-time adversary can distinguish



VGBO: Virtual Grey Box Obfuscation; **AIPO:** Auxiliary-Input Point-function Obfuscation; **iO:** Indistinguishability Obfuscation; **AI-DHI1**, **AI-DHI2:** Auxiliary-Input DH Inversion Assumptions; **PRIV1-AI-DPKE:** Auxiliary-Input, single-message Deterministic PKE; **KM-LR-SE:** Key-Message Leakage-resilient Symmetric Encryption; **AI-OWF:** Auxiliary-Input One-Way Function; **UCE[S]:** Universal Computational Extractor for class of sources **S**; $\mathbf{S}^{\text{splt}}$**:** Class of split sources; $\mathbf{S}^{\text{cup}}$**:** Class of computationally unpredictable sources; $\mathbf{S}^{1}$**:** Class of sources making only one oracle query.

Figure 1: **Relations established in this paper.** $\mathbf{A} \Rightarrow \neg\mathbf{B}$ means that if **A** exists or is true then **B** is not; $\mathbf{A} \Rightarrow \mathbf{B}$ means that if **A** exists or is true then so is **B**; $\neg\mathbf{A} \Rightarrow \neg\mathbf{B}$ means that if **A** does not exist then neither does **B**, and is equivalent to $\mathbf{B} \Rightarrow \mathbf{A}$.

---

between (r, r^x) and (r, s), even when given auxiliary information a about x, as long as this information a is "x-prediction-precluding," meaning does not allow one to just compute x in polynomial time. The assumption has been used for oracle hashing [22], point-function obfuscation [12] and zero-knowledge proofs [12].

Recall that our result is that $\mathbf{VGBO} \Rightarrow \neg\mathbf{AI\text{-}DHI1}$: If VGBO for all circuits is possible then the AI-DHI1 assumption is false. To prove this, we take any ensemble $\mathcal{G} = \{\mathbb{G}_\lambda : \lambda \in \mathbb{N}\}$ of prime-order groups. For random x, we define a way of picking the auxiliary information a such that (1) a is x-prediction-precluding, but (2) there is a polynomial-time adversary that, given a, can distinguish between (r, r^x) and (r, s) for random r, s. Consider the circuit C_x that on input u, v returns 1 if $v = u^x$ and 0 otherwise. The auxiliary information a will be a VGB obfuscation $\overline{\mathrm{C}}$ of C_x. Now (2) is easy to see: the adversary, given challenge (u, v), can win by returning $\overline{\mathrm{C}}(u, v)$. But why is (1) true? We first use the assumed VGB security of the obfuscator to reduce (1) to showing that *no, even unbounded*, simulator, given an oracle for C_x, can extract x in a polynomial number of queries. The latter is then shown through an information-theoretic argument that exploits the group structure.

AI-DHI2. If the VGB conjecture is true, then our result above says we lose AI-DHI1 and all results and constructions based on it. To recover these, we suggest a modification (weakening) of AI-DHI1 that we call AI-DHI2. The modification is simple. Recall that in AI-DHI1, the ensemble contains a single group $\mathbb{G}_\lambda$ for every value of the security parameter λ. We allow a finite family of groups for each value of λ, the description $\langle\mathbb{G}\rangle$ of a particular group $\mathbb{G}$ to be picked by a polynomial-time generator algorithm GG on input 1^λ. We consider a parameterized assumption AI-DHI2[GG] that says that, for $\langle\mathbb{G}\rangle$ generated by GG, and for r, s then chosen at random from $\mathbb{G}$, no polynomial-time adversary can distinguish between (r, r^x) and (r, s), even when given $\langle\mathbb{G}\rangle$ and auxiliary information a about x, as long as this information a is x-prediction-precluding. A key element of this formulation is that the auxiliary information a does not depend on $\mathbb{G}$. Finally the AI-DHI2 assumption is that there exists GG such that the AI-DHI2[GG] assumption holds.

The two criteria for a good assumption are *plausibility* and *utility*. In our present context the first corresponds to asking whether our methods can be extended to show that $\mathbf{VGBO} \Rightarrow \neg\mathbf{AI\text{-}DHI2}$. We find that whether or not our attack works depends on the generator. For some GG satisfying a property we call "verifiability," we can show that $\mathbf{VGBO} \Rightarrow \neg\mathbf{AI\text{-}DHI2}[\mathsf{GG}]$. However, there are GG that do not appear to be verifiable, leaving **AI-DHI2** a viable assumption even if VGBO for all circuits is possible. This is a step forward but not by itself enough; the utility we want is that applications that used AI-DHI1 can use AI-DHI2. Below we will show this and more.

One might say it is premature to introduce and use AI-DHI2, since we do not know that VGBO is possible, and AI-DHI1 may be true. But notice that AI-DHI2 is a *weaker* assumption than AI-DHI1. That is, **AI-DHI1** $\Rightarrow$ **AI-DHI2**. (GG could always output the single group $\mathbb{G}_\lambda$ on input 1^λ.) So using AI-DHI2 in place of AI-DHI1 is a win-win situation. If VGBO is not possible and AI-DHI1 is true, then so is AI-DHI2, but if VGBO is possible and AI-DHI1 is thus false, AI-DHI2 and applications based on it may still be standing.

AIPO results. A point function with target k is the circuit $\mathbf{I}_k$ that on input k' returns 1 if $k' = k$ and 0 otherwise. When, faced with the impossibility of VBBO for all circuits, researchers asked whether one could obfuscate particular classes of circuits, point functions emerged as the canonical target, due both to their being so basic and to their obfuscation having many applications [22, 39, 43, 34, 24, 25, 8, 12, 40, 20].

AIPO is a strong formalization of security for point-function obfuscation from [34, 12]. It asks that the obfuscations of $\mathbf{I}_{k_0}$ and $\mathbf{I}_{k_1}$ be indistinguishable even given k_1-prediction-precluding auxiliary information a where k_0, k_1 are drawn from some common distribution. This formulation has been used for 3-round ZK [12] and certain forms of UCE [21]. Achieving it has however been (surprisingly) difficult. The primary construction is from AI-DHI1 [22, 12]: to obfuscate $\mathbf{I}_k$, pick a random r from the group $\mathbb{G}_\lambda$, let $s = r^k$ and return the circuit $\overline{\mathrm{C}}_{r,s}$ that on input k' returns 1 if $r^{k'} = s$ and 0 otherwise.

If the VGB conjecture is true, then our result above says we lose the AI-DHI1-based construction of AIPO. In fact, it is not just that the assumption fails; our attack directly shows that the construction itself fails. This calls AIPO into question and in particular leads us to ask whether VGBO and AIPO can co-exist. We suggest that they can by providing several alternative constructions of AIPO under different assumptions that as far as we know can co-exist with VGBO.

We give four constructions that are summarized in the second set of relations in Fig. 1 and that we now discuss. (1) **AI-DHI2** $\Rightarrow$ **AIPO**: We show that the weakening AI-DHI2 of AI-DHI1 we introduced above, and that as far as we know can co-exist with VGBO, is enough to guarantee AIPO. The construction is as from AI-DHI1 except that the group is chosen dynamically by the obfuscator, anew for each invocation. (2) **UCE**$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1] \Rightarrow$ **AIPO**: We provide a construction of AIPO based on function families satisfying an assumption in the UCE (Universal Computational Extractor) class of assumptions of [5], namely UCE for the class $\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1$ of computationally unpredictable split sources making only one oracle query. The definitions are recalled in Appendix A, but what is most relevant here is that the strength of a UCE assumption is very sensitive to the choice of class of sources that parameterizes the assumption, and UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1]$ is a conservative choice that evades the BFM iO-based attacks against UCE$[\mathbf{S}^{\mathrm{cup}}]$ [18]. (3) **PRIV1-AI-DPKE** $\Rightarrow$ **AIPO**: We construct AIPO from deterministic public-key encryption (DPKE) schemes that are secure even given message-prediction-precluding auxiliary information (AI), as defined by Brakerski and Segev [17]. We only require security for a single message (PRIV1) so the negative result of [45] does not apply. (4) **AI-OWF**$+$**iO** $\Rightarrow$ **AIPO**: We construct AIPO from the weakest primitive in the auxiliary-information domain, namely a function that is one-way even given input-prediction-precluding auxiliary information. The construction additionally uses iO.

Negative relations from iO. Indistinguishability Obfuscation (iO) [3, 30] for all circuits has already been the starting point for negative relations **iO** $\Rightarrow \neg$**B**, for different choices of **B** [11, 18, 20, 19, 4, 35]. This last segment of our paper provides further iO-based negative relations, depicted at the bottom of Fig. 1, that broaden and improve prior ones.

DKL [29] and CKVW [25] provide key leakage resilient symmetric encryption (K-LR-SE) schemes: they retain security even when the adversary has any key-prediction-precluding auxil-

iary information about the key. We consider a generalization: In key-message leakage resilient symmetric encryption (KM-LR-SE), the auxiliary information is allowed to depend not just on the key but also on the message, the requirement however still being that it is key-prediction-precluding. The enhancement would appear to be innocuous, because the strong semantic-security style formalizations of encryption that we employ in any case allow the adversary to have a priori information about the message. However, we show that this goal is impossible to achieve if iO for all circuits is possible. That is, $\mathbf{iO} \Rightarrow \neg\mathbf{KM\text{-}LR\text{-}SE}$. We use the technique introduced by Brzuska and Mittelback (BM1) [20] to show that $\mathbf{iO} \Rightarrow \neg\mathbf{MB\text{-}AIPO}$ and our result is essentially a reformulation of the latter, but, besides being of interest from the perspective of leakage-resilience, this reformulation will be the basis of a further negative result as we discuss below.

BFM [18] showed that $\mathbf{iO} \Rightarrow \neg\mathbf{UCE}[\mathbf{S}^{\mathrm{cup}}]$. That is, if iO for all circuits is possible, then no family of functions can achieve UCE security relative to the class $\mathbf{S}^{\mathrm{cup}}$ of all computationally unpredictable sources. This lead BHK [5] to propose further restricting attention to "split" sources. Such sources can leak information about an oracle query and its answer separately, but not together, which circumvents the BFM attack. Indeed, UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$ appeared plausible. However here we show that $\mathbf{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}] \Rightarrow \mathbf{KM\text{-}LR\text{-}SE}$, meaning we can build a key-message leakage resilient symmetric encryption scheme given any UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$-secure function family. But we saw above that $\mathbf{iO} \Rightarrow \neg\mathbf{KM\text{-}LR\text{-}SE}$ and can thus conclude that $\mathbf{iO} \Rightarrow \neg\mathbf{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$. That is, if iO for all circuits is possible, then UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$-security is not achievable.

In UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$, the source is allowed a polynomial number of oracle queries. Our attacks do not threaten UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1}]$, the assumption we used above to get AIPO, because here the source is only allowed one oracle query. In fact we are not aware of any applications assuming UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$; prior applications [5, 4] have also either used UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1}]$ or quite different classes like UCE$[\mathbf{S}^{\mathrm{sup}}]$, and neither of these is at risk. However our $\mathbf{iO} \Rightarrow \neg\mathbf{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$ result is of interest towards understanding the achievability of UCE assumptions and the effectiveness of different kinds of restrictions (in this case, splitting) on sources.

Discussion and further related work. Our above-mentioned $\mathbf{PRIV1\text{-}AI\text{-}DPKE} \Rightarrow \mathbf{AIPO}$ result motivates achieving PRIV1-AI-DPKE. BS [17] target this goal and provide schemes based on standard assumptions such as DLIN and subgroup indistinguishability, but they only achieve security if the message is subexponentially unpredictable from the auxiliary information while we need security even when it is only polynomially unpredictable from the auxiliary information.

Wee [43] provides a point-function obfuscator based on a fixed permutation about which a novel assumption is made. This construction does not target AIPO, but one can ask whether or not it achieves it. Goldwasser and Kalai [34] show that it does not. BP [12] consider the construction with a family of permutations rather than a fixed one, and show, under a novel assumption, that in this case it does achieve AIPO. BP [12] also note that this construction cannot be proven AIPO based only on the assumption that the underlying permutation is a AI-OWF. In this light it is interesting that we obtain AIPO from an arbitrary AI-OWF under the additional assumption of iO.

Brzuska and Mittelbach (BM1) [20] show that $\mathbf{iO} \Rightarrow \neg\mathbf{MB\text{-}AIPO}$. The latter, multi-bit auxiliary-input point-function obfuscation [40], seems to be quite a bit stronger than AIPO itself and in particular this result does not rule out AIPO.

Brzuska and Mittelbach (BM2) [21] show that $\mathbf{iO} + \mathbf{AIPO} \Rightarrow \mathbf{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1}]$. From this and our constructions of AIPO we obtain $\mathbf{iO} + \mathbf{AI\text{-}DHI2} \Rightarrow \mathbf{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1}]$, a construction of $\mathbf{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1}]$ function families from iO and an algebraic assumption. We also obtain $\mathbf{iO} + \mathbf{AI\text{-}OWF} \Rightarrow \mathbf{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1}]$. These positive results further validate the UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1}]$ assumption.

BHK [5] use UCE to construct point obfuscators providing security when the target is statistically unpredictable given the auxiliary input. AIPO requires more, namely security even when the target is only computationally unpredictable given the auxiliary input.

2 Preliminaries

Notation. We denote by $\lambda \in \mathbb{N}$ the security parameter and by 1^λ its unary representation. We denote the size of a finite set X by $|X|$. If $x \in \{0,1\}^*$ is a string then $|x|$ denotes its length, $x[i]$ denotes its i-th bit, and $x[i..j] = x[i]\ldots x[j]$ for $1 \le i \le j \le |x|$. We let ε denote the empty string. If s is an integer then $\mathsf{Pad}_s(\mathrm{C})$ denotes circuit C padded to have size s. We say that circuits $\mathrm{C}_0, \mathrm{C}_1$ are equivalent, written $\mathrm{C}_0 \equiv \mathrm{C}_1$, if they agree on all inputs. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes the number of its coordinates and $\mathbf{x}[i]$ denotes its i-th coordinate. If X is a finite set, we let $x \leftarrow\!\!\$\, X$ denote picking an element of X uniformly at random and assigning it to x. Algorithms may be randomized unless otherwise indicated. Running time is worst case. "PT" stands for "polynomial-time," whether for randomized algorithms or deterministic ones. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with random coins r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \leftarrow\!\!\$\, A(x_1, \ldots)$ be the result of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. We let $[A(x_1, \ldots)]$ denote the set of all possible outputs of A when invoked with inputs $x_1, \ldots$. We say that $f\colon \mathbb{N} \to \mathbb{R}$ is negligible if for every positive polynomial p, there exists $\lambda_p \in \mathbb{N}$ such that $f(\lambda) < 1/p(\lambda)$ for all $\lambda > \lambda_p$. We use the code based game playing framework of [6]. (See Fig. 2 for an example.) By $\mathrm{G}^{\mathcal{A}}(\lambda)$ we denote the event that the execution of game G with adversary $\mathcal{A}$ and security parameter λ results in the game returning true.

Function families. A family of functions F specifies the following. PT key generation algorithm $\mathsf{F.Kg}$ takes 1^λ to return a key $fk \in \{0,1\}^{\mathsf{F.kl}(\lambda)}$, where $\mathsf{F.kl}\colon \mathbb{N} \to \mathbb{N}$ is the key length function associated to F. Deterministic, PT evaluation algorithm $\mathsf{F.Ev}$ takes 1^λ, key $fk \in [\mathsf{F.Kg}(1^\lambda)]$ and an input $x \in \{0,1\}^{\mathsf{F.il}(\lambda)}$ to return an output $\mathsf{F.Ev}(1^\lambda, fk, x) \in \{0,1\}^{\mathsf{F.ol}(\lambda)}$, where $\mathsf{F.il}, \mathsf{F.ol}\colon \mathbb{N} \to \mathbb{N}$ are the input and output length functions associated to F, respectively. We say that F is *injective* if the function $\mathsf{F.Ev}(1^\lambda, fk, \cdot)\colon \{0,1\}^{\mathsf{F.il}(\lambda)} \to \{0,1\}^{\mathsf{F.ol}(\lambda)}$ is injective for every $\lambda \in \mathbb{N}$ and every $fk \in [\mathsf{F.Kg}(1^\lambda)]$. Notions of security for function families that we will use are UCE and AI-OWF, defined in Appendix A and Section 4 respectively.

Auxiliary information generators. Auxiliary information leaving some quantity computationally unpredictable is common to many of the notions we consider and it is convenient to abstract this out. An *auxiliary information generator* X specifies a PT algorithm $\mathsf{X.Ev}$ that takes 1^λ to return a *target* $k \in \{0,1\}^{\mathsf{X.tl}(\lambda)}$, a *payload* $m \in \{0,1\}^{\mathsf{X.pl}(\lambda)}$ and *auxiliary information* a, where $\mathsf{X.tl}, \mathsf{X.pl}\colon \mathbb{N} \to \mathbb{N}$ are the target and payload lengths, respectively. Consider game PRED of Fig. 2 associated to X and a predictor adversary $\mathcal{Q}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda) = \Pr[\mathrm{PRED}^{\mathcal{Q}}_{\mathsf{X}}(\lambda)]$. We say that X is *unpredictable* if $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\cdot)$ is negligible for every PT adversary $\mathcal{Q}$. We say that X is *uniform* if $\mathsf{X.Ev}(1^\lambda)$ picks the target $k \in \{0,1\}^{\mathsf{X.tl}(\lambda)}$ and the payload $m \in \{0,1\}^{\mathsf{X.pl}(\lambda)}$ uniformly and independently. Note that the auxiliary information a may depend on both the target k and the payload m, but unpredictability refers to recovery of the target k alone.

Obfuscators. An *obfuscator* is a PT algorithm Obf that on input 1^λ and a circuit C returns a circuit $\overline{\mathrm{C}}$ such that $\overline{\mathrm{C}} \equiv \mathrm{C}$. (That is, $\overline{\mathrm{C}}(x) = \mathrm{C}(x)$ for all x.) We refer to the latter as the *correctness condition*. In the sequel we consider various notions of security for obfuscators, namely VGBO, AIPO and iO.

Game $\mathrm{PRED}^{\mathcal{Q}}_{\mathsf{X}}(\lambda)$	Game $\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}}(\lambda)$	Game $\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp},q}(\lambda)$
$(k, m, a) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$\mathrm{C} \leftarrow\!\!\$\ \mathsf{Smp}(1^\lambda)$	$\mathrm{C} \leftarrow\!\!\$\ \mathsf{Smp}(1^\lambda)\ ;\ i \leftarrow 0$
$k' \leftarrow\!\!\$\ \mathcal{Q}(1^\lambda, a)$	$\overline{\mathrm{C}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathrm{C})$	$b' \leftarrow\!\!\$\ \mathcal{S}^{\mathrm{CIRC}}(1^\lambda)\ ;$ Return b'
Return $(k = k')$	$b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathrm{C}})$	$\underline{\mathrm{CIRC}(x)}$
	Return b'	$i \leftarrow i + 1$
		If $i > q(\lambda)$ then return $\perp$
		$y \leftarrow \mathrm{C}(x)\ ;$ Return y

Figure 2: Game defining unpredictabilty of auxiliary information generator X and games defining VGB security of obfuscator Obf.

3 VGBO and the AI-DHI assumptions

BCKP [9] conjecture that candidate constructions of iO also achieve VGBO and thus in particular that VGB obfuscation for all circuits is possible. Here we explore the plausibility of this *VGB conjecture*. We show that it implies the failure of Canetti's AI-DHI1 assumption. Either this assumption is false or VGBO for all circuits is not possible. (In fact, our result refers to an even weaker VGBO assumption.) We then suggest a weakening of AI-DHI1 that we call AI-DHI2 that is parameterized by a group generator. We show that our attack on AI-DHI1 extends to rule out AI-DHI2 for group generators satisfying a property we call verifiability. However there are group generators that do not appear to be verifiable, making AI-DHI2 a viable alternative to AI-DHI1. In later sections we will show how to recover, under AI-DHI2, the most important application of AI-DHI1, namely AIPO.

Definitions. Let Obf be an obfuscator as defined in Section 2. We define what it means for it to be a VGB obfuscator. We will use a weak variant of the notion used in some of the literature [8, 9], which strengthens our results since they are negative relations with starting point VGBO.

A *sampler* Smp in this context is an algorithm that takes 1^λ to return a circuit C. Let q be a polynomial, $\mathcal{A}$ an adversary and $\mathcal{S}$ a (not necessarily PT) algorithm called a simulator. For $\lambda \in \mathbb{N}$ let

$$\mathsf{Adv}^{\mathsf{vgb}}_{\mathsf{Obf},\mathsf{Smp},q,\mathcal{A},\mathcal{S}}(\lambda) = \left|\Pr\left[\,\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}}(\lambda)\,\right] - \Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp},q}(\lambda)\,\right]\right|$$

where the games are in Fig. 2. Let SAMP be a set of samplers. We say that Obf is a VGB obfuscator for SAMP if for every PT adversary $\mathcal{A}$ there exists a (not necessarily PT) simulator $\mathcal{S}$ and a polynomial q such that $\mathsf{Adv}^{\mathsf{vgb}}_{\mathsf{Obf},\mathsf{Smp},q,\mathcal{A},\mathcal{S}}(\cdot)$ is negligible for all $\mathsf{Smp} \in \mathsf{SAMP}$.

We note that [9] use a VGB variant stronger than the above where the advantage measures the difference in probabilities of $\mathcal{A}$ and $\mathcal{S}$ guessing a predicate $\pi(\mathrm{C})$, rather than just the probabilities of outputting one, which is all we need here. Also note that our VGB definition is trivially achievable whenever $|\mathsf{SAMP}| = 1$, since $\mathcal{S}$ can simulate game $\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}}(\lambda)$ for any fixed choice of $\mathcal{A}$ and Smp. Our applications however use a SAMP of size 2.

The AI-DHI1 assumption. Let $\mathcal{G} = \{\mathbb{G}_\lambda : \lambda \in \mathbb{N}\}$ be an ensemble of groups where for every $\lambda \in \mathbb{N}$ the order $p(\lambda)$ of group $\mathbb{G}_\lambda$ is a prime in the range $2^{\lambda-1} < p(\lambda) < 2^\lambda$. We assume that relevant operations are computable in time polynomial in λ, including computing $p(\cdot)$, testing membership in $\mathbb{G}_\lambda$ and performing operations in $\mathbb{G}_\lambda$. By $\mathbb{G}^*_\lambda$ we denote the non-identity members of the group, which is the set of generators since the group has prime order. An auxiliary information generator X for $\mathcal{G}$ is an auxiliary information generator as per Section 2 with the additional property that

Game $\text{AIDHI1}^{\mathcal{A}}_{\mathcal{G},\mathsf{X}}(\lambda)$	Game $\text{AIDHI2}^{\mathcal{A}}_{\mathsf{GG},\mathsf{X}}(\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}$	$b \leftarrow\!\!\$\ \{0,1\}\ ;\ \langle\mathbb{G}\rangle \leftarrow\!\!\$\ \mathsf{GG}(1^\lambda)$
$(k,\varepsilon,a) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$(k,\varepsilon,a) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$
$g \leftarrow\!\!\$\ \mathbb{G}^*_\lambda\ ;\ K_1 \leftarrow g^k\ ;\ K_0 \leftarrow\!\!\$\ \mathbb{G}_\lambda$	$g \leftarrow\!\!\$\ \mathrm{Gen}(\mathbb{G})\ ;\ K_1 \leftarrow g^k\ ;\ K_0 \leftarrow\!\!\$\ \mathbb{G}$
$b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, g, K_b, a)$	$b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \langle\mathbb{G}\rangle, g, K_b, a)$
Return $(b = b')$	Return $(b = b')$

Figure 3: Games defining the AI-DHI1 and AI-DHI2 assumptions.

---

the target k returned by $\mathsf{X.Ev}(1^\lambda)$ is in $\mathbb{Z}_{p(\lambda)}$ (i.e. is an exponent) and the payload m is ε (i.e. is effectively absent).

Now consider game AIDHI1 of Fig. 3 associated to $\mathcal{G}, \mathsf{X}$ and an adversary $\mathcal{A}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{aidhi1}}_{\mathcal{G},\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\text{AIDHI1}^{\mathcal{A}}_{\mathcal{G},\mathsf{X}}(\lambda)] - 1$. We say that $\mathcal{G}$ is AI-DHI1-secure if $\mathsf{Adv}^{\mathsf{aidhi1}}_{\mathcal{G},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for every unpredictable X for $\mathcal{G}$ and every PT adversary $\mathcal{A}$. The AI-DHI1 assumption [22, 12] is that there exists a family of groups $\mathcal{G}$ which is AI-DHI1 secure.

VGBO $\Rightarrow \neg$AI-DHI1. The following says if VGB obfuscation is possible then the AI-DHI1 assumption is false: there exists *no* family of groups $\mathcal{G}$ that is AI-DHI1 secure. Our theorem only assumes a very weak form of VGB obfuscation for a class with two samplers (given in the proof).

Theorem 3.1 *Let $\mathcal{G}$ be a family of groups. Then there is a pair $\mathsf{Smp}, \mathsf{Smp}_0$ of PT samplers (defined in the proof) such that if there exists a VGB-secure obfuscator for the class $\mathsf{SAMP} = \{\mathsf{Smp}, \mathsf{Smp}_0\}$, then $\mathcal{G}$ is not AI-DHI1-secure.*

Proof of Theorem 3.1: Let Obf be the assumed obfuscator. Let X be the auxiliary information generator for $\mathcal{G}$ defined as follows:

Algorithm $\mathsf{X.Ev}(1^\lambda)$	Circuit $\mathrm{C}_{1^\lambda,k}(g,K)$
$k \leftarrow\!\!\$\ \mathbb{Z}_{p(\lambda)}$	If $(g \notin \mathbb{G}^*_\lambda$ or $K \notin \mathbb{G}_\lambda)$ then return 0
$\overline{\mathrm{C}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathrm{C}_{1^\lambda,k})$	If $(g^k = K)$ then return 1
Return $(k, \varepsilon, \overline{\mathrm{C}})$	Else return 0

The auxiliary information $a = \overline{\mathrm{C}}$ produced by X is an obfuscation of the circuit $\mathrm{C}_{1^\lambda,a}$ shown on the right above. The circuit has 1^λ and the target value k embedded inside. The circuit takes inputs g, K and checks that the first is a group element different from the identity —and thus a generator— and the second is a group element. It then returns 1 if g^k equals K, and 0 otherwise.

We first construct a PT adversary $\mathcal{A}^*$ such that $\mathsf{Adv}^{\mathsf{aidhi1}}_{\mathcal{G},\mathsf{X},\mathcal{A}^*}(\cdot)$ is non-negligible. On input $1^\lambda, g, K_b, \overline{\mathrm{C}}$, it simply returns $\overline{\mathrm{C}}(g, K_b)$. That is, it runs the obfuscated circuit $\overline{\mathrm{C}}$ on g and K_b to return the outcome. If the challenge bit b in game $\text{AIDHI1}^{\mathcal{A}^*}_{\mathcal{G},\mathsf{X}}(\lambda)$ is 1 then the adversary always outputs $b' = 1$. Otherwise, the adversary outputs $b' = 1$ with probability $1/p(\lambda)$. We have $\mathsf{Adv}^{\mathsf{aidhi1}}_{\mathcal{G},\mathsf{X},\mathcal{A}^*}(\lambda) = 1 - 1/p(\lambda) \geq 1 - 2^{1-\lambda}$, which is not negligible.

We now show that the constructed auxiliary information generator X is unpredictable. In particular, for any PT adversary $\mathcal{Q}$ we construct a PT adversary $\mathcal{A}$ and samplers Smp, Smp_0 such that for all simulators $\mathcal{S}$ and all polynomials q,

$$\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda) \leq \mathsf{Adv}^{\mathsf{vgb}}_{\mathsf{Obf},\mathsf{Smp},q,\mathcal{A},\mathcal{S}}(\lambda) + \mathsf{Adv}^{\mathsf{vgb}}_{\mathsf{Obf},\mathsf{Smp}_0,q,\mathcal{A},\mathcal{S}}(\lambda) + \frac{q(\lambda)}{2^{\lambda-1}}\ . \tag{1}$$

Concretely, the adversary $\mathcal{A}$ and the samplers $\mathsf{Smp}, \mathsf{Smp}_0$ operate as follows:

Algorithm $\mathsf{Smp}(1^\lambda)$	Algorithm $\mathsf{Smp}_0(1^\lambda)$	Adversary $\mathcal{A}(1^\lambda, \overline{\mathrm{C}})$
$k \leftarrow\!\!\$\ \mathbb{Z}_{p(\lambda)}$	Return C_0	$k' \leftarrow\!\!\$\ \mathcal{Q}(1^\lambda, \overline{\mathrm{C}})$
Return $\mathrm{C}_{1^\lambda,k}$		$\bar{g} \leftarrow\!\!\$\ \mathbb{G}_\lambda^*$
		Return $\overline{\mathrm{C}}(\bar{g}, \bar{g}^{k'})$

Here, in Smp_0, the circuit C_0 takes as input a pair group elements g, g' from $\mathbb{G}_\lambda$ and always returns 0.

To show Equation (1), we first note that by construction

$$\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda) = \Pr\left[\,\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}}(\lambda)\,\right], \tag{2}$$

because an execution of $\mathrm{PRED}^{\mathcal{Q}}_{\mathsf{X}}(\lambda)$ results in the same output distribution as in $\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}}(\lambda)$. The only difference is that in the latter, the check of whether the guess is correct is done via the obfuscated circuit $\overline{\mathrm{C}}$. Therefore, for all simulators $\mathcal{S}$ and polynomials q, we can rewrite Equation (2) as

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda) = {} & \Pr\left[\,\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}}(\lambda)\,\right] - \Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp},q}(\lambda)\,\right] \\
& + \Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp},q}(\lambda)\,\right] - \Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp}_0,q}(\lambda)\,\right] \\
& + \Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp}_0,q}(\lambda)\,\right] - \Pr\left[\,\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}_0}(\lambda)\,\right] \\
& + \Pr\left[\,\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}_0}(\lambda)\,\right].
\end{aligned}$$

To upper bound $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda)$, we first note that

$$\Pr\left[\,\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}}(\lambda)\,\right] - \Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp},q}(\lambda)\,\right] \leq \mathsf{Adv}^{\mathsf{vgb}}_{\mathsf{Obf},\mathsf{Smp},q,\mathcal{A},\mathcal{S}}(\lambda)$$

and

$$\Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp}_0,q}(\lambda)\,\right] - \Pr\left[\,\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}_0}(\lambda)\,\right] \leq \mathsf{Adv}^{\mathsf{vgb}}_{\mathsf{Obf},\mathsf{Smp}_0,q,\mathcal{A},\mathcal{S}}(\lambda)\,.$$

Moreover, $\Pr\left[\,\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}_0}(\lambda)\,\right] = 0$, since by construction $\mathcal{A}$ never outputs 1 in $\mathrm{VGB1}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{Smp}_0}(\lambda)$, as it is given an obfuscation of the constant circuit C_0.

We are left with upper bounding the difference between $\Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp},q}(\lambda)\,\right]$ and $\Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp}_0,q}(\lambda)\,\right]$. Note that $\mathcal{S}$ is allowed to issue at most $q(\lambda)$ queries to the given circuit, which is either $\mathrm{C}_{1^\lambda,k}$ for a random $k \leftarrow\!\!\$\ \mathbb{Z}_{p(\lambda)}$ or C_0. Denote by Hit the event that $\mathcal{S}$ makes a query (g, K) in $\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp},q}(\lambda)$ such that $g^k = K$. Then, by a standard argument,

$$\Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp},q}(\lambda)\,\right] - \Pr\left[\,\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp}_0,q}(\lambda)\,\right] \leq \Pr\left[\,\mathsf{Hit}\,\right].$$

Namely, to compute $\Pr\left[\,\mathsf{Hit}\,\right]$, we move from $\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp},q}(\lambda)$ to the simpler $\mathrm{VGB0}^{\mathcal{S}}_{\mathsf{Smp}_0,q}(\lambda)$, where all of $\mathcal{S}$'s queries are answered with 0. We extend the game to sample a random key $k \leftarrow\!\!\$\ \mathbb{Z}_{p(\lambda)}$, and we define Hit' as the event that for one of $\mathcal{S}$'s queries (g, K) we have $g^k = K$. It is not hard to see that $\Pr\left[\,\mathsf{Hit}'\,\right]$ and $\Pr\left[\,\mathsf{Hit}\,\right]$ are identical, as both games are identical as long as the corresponding events do not occur. Since there are at most $q(\lambda)$ queries, and exactly one k can produce the answer

1 for each query, the union bound yields

$$\Pr[\,\mathsf{Hit}\,] = \Pr\left[\,\mathsf{Hit}'\,\right] \leq \frac{q(\lambda)}{p(\lambda)} \leq \frac{q(\lambda)}{2^{\lambda-1}}\ ,$$

which concludes the proof. ∎

The AI-DHI2 assumption. We now suggest a relaxation AI-DHI2 of the AI-DHI1 assumption given above. The idea is that for each value of λ there is not one, but many possible groups. Formally, a *group generator* is a PT algorithm GG that on input 1^λ returns a description $\langle\mathbb{G}\rangle$ of a cyclic group $\mathbb{G}$ whose order $|\mathbb{G}|$ is in the range $2^{\lambda-1} < |\mathbb{G}| < 2^\lambda$. We assume that given $1^\lambda, \langle\mathbb{G}\rangle$, relevant operations are computable in time polynomial in λ, including performing group operations in $\mathbb{G}$ and picking at random from $\mathbb{G}$ and the set $\mathrm{Gen}(\mathbb{G})$ of generators of $\mathbb{G}$. An auxiliary information generator X for GG is an auxiliary information generator as per Section 2 with the additional property that the target k returned by $\mathsf{X.Ev}(1^\lambda)$ is in $\mathbb{Z}_{2^{\lambda-1}}$ —this makes it a valid exponent for *any* group $\mathbb{G}$ with $\langle\mathbb{G}\rangle \in [\mathsf{GG}(1^\lambda)]$— and the payload m is ε (i.e. is effectively absent).

Now consider game AIDHI2 of Fig. 3 associated to GG, X and an adversary $\mathcal{A}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{aidhi2}}_{\mathsf{GG},\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{AIDHI2}^{\mathcal{A}}_{\mathsf{GG},\mathsf{X}}(\lambda)] - 1$. We say that GG is AI-DHI2-secure if $\mathsf{Adv}^{\mathsf{aidhi2}}_{\mathsf{GG},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for every unpredictable X for GG and every PT adversary $\mathcal{A}$. The (new) AI-DHI2 assumption is that there exists a group generator GG which is AI-DHI2 secure.

A *verifier* for group generator GG is a deterministic, PT algorithm $\mathsf{GG.Vf}$ that can check whether a given string d is a valid description of a group generated by the generator GG. Formally, $\mathsf{GG.Vf}$ on input $1^\lambda, d$ returns true if $d \in [\mathsf{GG}(1^\lambda)]$ and false otherwise, for all $d \in \{0,1\}^*$. We say that GG is *verifiable* if it has a verifier and additionally, in time polynomial in $1^\lambda, \langle\mathbb{G}\rangle$, where $\langle\mathbb{G}\rangle \in [\mathsf{GG}(1^\lambda)]$, one can test membership in $\mathbb{G}$ and in the set $\mathrm{Gen}(\mathbb{G})$ of generators of $\mathbb{G}$. The following extends Theorem 3.1 to say that if VGBO is possible then no verifiable group generator is AI-DHI2 secure.

Theorem 3.2 *Let* GG *be verifiable group generator. Then there is a pair* $\mathsf{Smp}, \mathsf{Smp}_0$ *of PT samplers such that if there exists a VGB-secure obfuscator for the class* $\mathsf{SAMP} = \{\mathsf{Smp}, \mathsf{Smp}_0\}$*, then* GG *is not AI-DHI2-secure.*

We omit a full proof, as it is very similar to the one of Theorem 3.1. We only note that to adapt the proof, we require $\mathsf{X.Ev}(1^\lambda)$ to output a random k in $\mathbb{Z}_{2^{\lambda-1}}$ together with the obfuscation of the following circuit $\mathrm{C}_{1^\lambda,k}$. The circuit $C_{1^\lambda,k}$ takes as input a string d expected to be a group description, together with two strings g and K. It first runs $\mathsf{GG.Vf}$ on input $1^\lambda, d$ to check whether $d \in [\mathsf{GG}(1^\lambda)]$, returning 0 if the check fails. If the check succeeds, so that we can write $d = \langle\mathbb{G}\rangle$, it further checks that $g \in \mathrm{Gen}(\mathbb{G})$ and $K \in \mathbb{G}$, returning 0 if this fails. Finally the circuit returns 1 if and only if $g^k = K$ in the group $\mathbb{G}$. The crucial point is that for every valid input (d, g, K), there is at most one $k \in \mathbb{Z}_{2^{\lambda-1}}$ which satisfies $g^k = K$ in the group described by d. This uses the assumption that $\mathbb{G}$ is cyclic.

Many group generators are cyclic and verifiable. For example, consider a generator GG that on input 1^λ returns a description of $\mathbb{G} = \mathbb{Z}_p^*$ for a safe prime $p = 2q - 1$. (That is, q is also a prime.) The verifier can extract p, q from $\langle\mathbb{G}\rangle$ and check their primality in PT. For such generators, we may prefer not to assume AI-DHI2-security, due to Theorem 3.2. However there are group generators that do not appear to be verifiable and where Theorem 3.2 thus does not apply. One must be careful to note that this does *not* mean that VGBO would not rule out AI-DHI2 security for these group generators. It just means that our current proof method may not work. Still at this point, the AI-DHI2 assumption, which only says there is *some* group generator that is AI-DHI2-secure, seems plausible. We will see in Section 4 that it allows us to recover AIPO, for which the primary existing construction was from AI-DHI1, and we will also give several other constructions of AIPO.

Game $\mathrm{AIPO}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$	Game $\mathrm{AIOWF}^{\mathcal{F}}_{\mathsf{F}}(\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}$	$fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$
$(k_0, m_0, a_0) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$(k, m, a) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$
$(k_1, m_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, k)$
$\overline{\mathrm{P}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathbf{I}_{k_b})$; $b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathrm{P}}, a_1)$	$k' \leftarrow\!\!\$\ \mathcal{F}(1^\lambda, fk, y, a)$
Return $(b = b')$	Return $(y = \mathsf{F.Ev}(1^\lambda, fk, k'))$

Figure 4: Games defining AIPO security of obfuscator Obf and auxiliary-input one-wayness of function family F.

4 Achieving AIPO

Auxiliary-input point-function obfuscation (AIPO) [34, 12] has been difficult to achieve. In fact, only two constructions have been proved AIPO secure. One is Canetti's [22], based on AI-DHI1. Also, BP [12] show AIPO security of Wee's construction [43] under an ad-hoc assumption much stronger than that of an AI-OWF, and it is not clear how it can be weakened. There are other constructions of point obfuscators [26, 5] but they do not achieve AIPO or are in the ROM [39].

In Section 3 we showed that if VGBO is possible then AI-DHI1 fails. In light of this and the above we are motivated to look for alternative AIPO constructions, particularly ones that can co-exist with VGBO. In this section, we provide four of them, starting with one based on the AI-DHI2 assumption we introduced in Section 3, and then going on to constructions from UCE, DPKE and AI-OWFs (cf. Fig. 1). We view the last as the most interesting.

Definitions. If k is a bit-string then $\mathbf{I}_k$: $\{0,1\}^{|k|} \to \{0,1\}$ denotes a canonical representation of the circuit that on input $k' \in \{0,1\}^{|k|}$ returns 1 if $k = k'$ and 0 otherwise. It is assumed that given $\mathbf{I}_k$, one can compute k in time linear in $|k|$. A circuit C is called a *point circuit* if there is a k, called the circuit target, such that $\mathrm{C} \equiv \mathbf{I}_k$. Let Obf be an obfuscator. The correctness condition we gave in Section 2 guarantees that on input $1^\lambda, \mathbf{I}_k$, it returns a point circuit with target k, which is the condition for calling it a point-function obfuscator. We say that Obf has target length $\mathsf{Obf.tl}$: $\mathbb{N} \to \mathbb{N}$ if the correctness condition is only required on inputs $\mathbf{I}_k$ with $k \in \{0,1\}^{\mathsf{Obf.tl}(\lambda)}$.

Let X be an auxiliary input generator. Let Obf be a point function obfuscator with $\mathsf{Obf.tl} = \mathsf{X.tl}$. We say that Obf is AIPO[X]-secure if $\mathsf{Adv}^{\mathsf{aipo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for every PT $\mathcal{A}$, where for $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{aipo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{AIPO}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)] - 1$ where game $\mathrm{AIPO}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$ is defined in Fig. 4. We say that Obf is AIPO-secure if it is AIPO[X]-secure for all unpredictable X with $\mathsf{X.tl} = \mathsf{Obf.tl}$.

AI-DHI2 ⇒ AIPO. Let GG be a group generator. We construct a point-function obfuscator Obf as follows. Let $\mathsf{Obf.tl}(\lambda) = \lambda - 1$ for all $\lambda \in \mathbb{N}$, and

Algorithm $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$	Circuit $\mathrm{C}_{\langle\mathbb{G}\rangle,g,K}(k)$
$\langle\mathbb{G}\rangle \leftarrow\!\!\$\ \mathsf{GG}(1^\lambda)$; $g \leftarrow\!\!\$\ \mathrm{Gen}(\mathbb{G})$	If $(g^k = K)$ then return 1
$K \leftarrow g^k$; Return $\mathrm{C}_{\langle\mathbb{G}\rangle,g,K}$	Else return 0

The following says this achieves AIPO under AI-DHI2. The proof is straightforward and is given in Appendix B.

Theorem 4.1 *Let* GG *be an AI-DHI2-secure group generator. Let* Obf *be as defined above. Then* Obf *is an AIPO-secure point-function obfuscator.*

UCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1$] ⇒ AIPO. We now construct an AIPO-secure point-function obfuscator from $\mathrm{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1]$, which is a relatively weak assumption in the UCE [5] class recalled in

Appendix A. Note that in an earlier work, BM [21] showed that UCE[$\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt} \cap \mathbf{S}^{1}$]-security is achievable assuming iO and AIPO. It follows from our result that UCE[$\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt} \cap \mathbf{S}^{1}$] and AIPO are equivalent, assuming iO.

Let H be a family of functions as per Section 2. Associate to it the point obfuscator Obf defined as follows. Let $\mathsf{Obf.tl} = \mathsf{H.il}$, and

Algorithm $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$	Circuit $\mathrm{C}_{1^\lambda, hk, y}(k')$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$; $y \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k)$	$y' \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k')$
Return $\mathrm{C}_{1^\lambda, hk, y}$	If $(y = y')$ then return 1 else return 0

The following theorem shows that UCE[$\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt} \cap \mathbf{S}^{1}$] implies AIPO-security of the above construction.

Theorem 4.2 *Let* H *be an injective family of functions that is* UCE[$\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt} \cap \mathbf{S}^{1}$]*-secure. Assume that* $2^{-\mathsf{H.ol}(\cdot)}$ *is negligible. Let* Obf *be as defined above. Then* Obf *is an* AIPO*-secure point function obfuscator.*

The injectivity of H is assumed in order to meet the correctness condition of a point-function obfuscator. It is not important for security. We note that the perfect correctness we have required for point-function obfuscators can be relaxed to a computational correctness requirement, namely that given an obfuscation $\overline{\mathrm{P}}$ of a point function $\mathbf{I}_k$, no PT adversary can find $k' \neq k$ such that $\overline{\mathrm{P}}(k') = 1$ holds with better than a negligible probability. This relaxed form of correctness can be achieved assuming nothing but UCE, meaning the injectivity requirement can be dropped.

Due to lack of space in the body, the proof of Theorem 4.2 is in Appendix C. Briefly, one first bounds the aipo advantage of an adversary against an auxiliary information generator X by the advantage of a source-distinguisher pair where the source is split and makes only one query. One then proves that the source is computationally unpredictable based on the unpredictability of X.

PRIV1-AI-DPKE ⇒ AIPO. We show that PRIV1-secure auxiliary-input deterministic public-key encryption [17] implies AIPO-secure point-function obfuscation. Note that [17, 44, 46] provide constructions achieving this security, but only for subexponentially hard to invert auxiliary inputs.

In Appendix D we recall the definitions following [17]. Now let $\mathsf{D\text{-}PKE}$ be an auxiliary-input deterministic public-key encryption scheme. We construct an obfuscator Obf with $\mathsf{Obf.tl} = \mathsf{D\text{-}PKE.ml}$ as follows:

Algorithm $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$	Circuit $\mathrm{C}_{1^\lambda, pk, c}(k)$
$(pk, sk) \leftarrow\!\!\$\ \mathsf{D\text{-}PKE.Kg}(1^\lambda)$	If $(\mathsf{D\text{-}PKE.Enc}(1^\lambda, pk, k) = c)$
$c \leftarrow \mathsf{D\text{-}PKE.Enc}(1^\lambda, pk, k)$; Return $\mathrm{C}_{1^\lambda, pk, c}$	Then return 1 else return 0

Note that the decryption correctness of $\mathsf{D\text{-}PKE}$ implies the correctness of the obfuscator, but this remains true even if decryption is not PT. The proof of the following theorem is quite straightforward and is given in Appendix D.

Theorem 4.3 *Let* $\mathsf{D\text{-}PKE}$ *be a PRIV1-secure auxiliary-input deterministic public-key encryption scheme. Let* Obf *be as defined above. Then* Obf *is an AIPO-secure point-function obfuscator.*

AI-OWF+iO ⇒ AIPO. We view this as our most interesting construction. In the class of primitives providing security in the presence of auxiliary inputs, the simplest and most basic is an auxiliary-input one-way function (AI-OWF). This simply means a family of functions which is one-way even in the presence of auxiliary information that leaves the input computationally hard to compute.

Games G_0, G_1

$b \leftarrow\!\!\$\ \{0,1\}$; $(k_0, m_0, a_0) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$; $(k_1, m_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$

$\quad fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$; $y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, k_b)$; $\overline{\mathrm{P}} \leftarrow\!\!\$\ \mathsf{Obf}^*(\mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{1^\lambda, fk, y}))$ // G_0

$\quad \overline{\mathrm{P}} \leftarrow\!\!\$\ \mathsf{Obf}^*(\mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2))$ // G_1

$b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathrm{P}}, a_1)$; Return $(b = b')$

Circuit $\mathrm{C}^1_{1^\lambda, fk, y}(k)$

If $(y = \mathsf{F.Ev}(1^\lambda, fk, k))$ then return 1 else return 0

Circuit $\mathrm{C}^2(k)$

Return 0

Figure 5: **Games for proof of Theorem 4.4.**

Can we obtain AIPO from an arbitrary AI-OWF? We show that one can if we additionally assume iO [3, 30, 42, 15, 1].

Let F be a family of functions and let X be an auxiliary information generator with $\mathsf{X.tl} = \mathsf{F.il}$ (cf. Section 2). We say that F is AIOWF[X]-secure if $\mathsf{Adv}^{\mathsf{ai\text{-}owf}}_{\mathsf{F},\mathsf{X},\mathcal{F}}(\cdot)$ is negligible for all PT adversaries $\mathcal{F}$, where $\mathsf{Adv}^{\mathsf{ai\text{-}owf}}_{\mathsf{F},\mathsf{X},\mathcal{F}}(\lambda) = \Pr[\mathrm{AIOWF}^{\mathcal{F}}_{\mathsf{F},\mathsf{X}}(\lambda)]$ and game $\mathrm{AIOWF}^{\mathcal{F}}_{\mathsf{F},\mathsf{X}}(\lambda)$ is defined in Fig. 4. We say that F is AIOWF-secure if it is AIOWF[X]-secure for all unpredictable X.

Previously a similar condition was required as a part of the definition of an auxiliary-input extractable one-way function [23]. We stress that we require only one-wayness; we do *not* require extractability.

BP [12] remark that the point-function obfuscator of [43] would not achieve AIPO security assuming only AIOWF-security of the underlying function. Our result is in some sense an answer, showing that even an arbitrary AIOWF function family yields AIPO if used in our construction with iO.

Let F be a family of functions. Let Obf^* be an indistinguishability obfuscator and let $s\colon \mathbb{N} \to \mathbb{N}$. We construct a point-function obfuscator Obf with $\mathsf{Obf.tl} = \mathsf{F.il}$ as follows:

Algorithm $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$

$fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$; $y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, k)$

$\overline{\mathrm{P}} \leftarrow\!\!\$\ \mathsf{Obf}^*(\mathsf{Pad}_{s(\lambda)}(\mathrm{C}_{1^\lambda, fk, y}))$; Return $\overline{\mathrm{P}}$

Circuit $\mathrm{C}_{1^\lambda, fk, y}(k')$

If $(y = \mathsf{F.Ev}(1^\lambda, fk, k'))$ then return 1

Else return 0

The following says this construction works if Obf^* is iO-secure.

Theorem 4.4 *Let F be an injective family of functions that is* AIOWF*-secure. Then there is a polynomial s such that the following is true. Let Obf^* be an indistinguishability obfuscator. Then Obf constructed above from $\mathsf{F}, s, \mathsf{Obf}^*$ is an* AIPO*-secure point-function obfuscator.*

As with our construction from UCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}$], here we only assume that F is injective in order to satisfy the (perfect) correctness requirement of an obfuscator. This assumption is required neither for "computational" correctness nor for security. We will exploit the fact that iO security of an obfuscator implies its diO security if the circuits differ on only one input [15]. Relevant definitions are recalled in Appendix A. **Proof of Theorem 4.4:** The injectivity of F implies that Obf satisfies the correctness condition of a point-function obfuscator. We now prove security. We define s as follows: for any λ let $s(\lambda)$ be a polynomial upper bound on $\max(|\mathrm{C}^1_{1^\lambda, fk, y}|, |\mathrm{C}^2|)$, where the circuits are defined in Fig. 5 and the maximum is over all $fk \in [\mathsf{F.Kg}(1^\lambda)]$ and $y \in \{0,1\}^{\mathsf{F.ol}(\lambda)}$. Let X be an auxiliary information generator. Let $\mathcal{A}$ be a PT adversary. Consider the games and the associated circuits of Fig. 5. Lines not annotated with comments are common to all games. Game G_0 is

equivalent to $\mathrm{AIPO}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$, and $\Pr[\mathrm{G}_1] = 1/2$ because the inputs to $\mathcal{A}$ in game G_1 do not depend on the challenge bit b. It follows that

$$\mathsf{Adv}^{\mathsf{aipo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda) = 2(\Pr[\mathrm{G}_1] + \Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]) - 1 = 2(\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]).$$

We now show that $\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]$ is negligible, which implies that $\mathsf{Adv}^{\mathsf{aipo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible and hence proves the theorem. We construct a circuit sampler S^* and an iO-adversary $\mathcal{O}$ as follows:

Circuit Sampler $\mathsf{S}^*(1^\lambda)$	Adversary $\mathcal{O}(1^\lambda, \overline{\mathrm{C}}, aux)$
$d \leftarrow\!\!\$\ \{0,1\}$; $fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$; $(k_0, m_0, a_0) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$(d, a_1) \leftarrow aux$
$(k_1, m_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$; $y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, k_d)$	$d' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathrm{C}}, a_1)$
$\mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{1^\lambda, fk, y})$; $\mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2)$	If $(d = d')$ then return 1
$aux \leftarrow (d, a_1)$; Return $(\mathrm{C}_0, \mathrm{C}_1, aux)$	Else return 0

It follows that $\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1] = \mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S}^*,\mathcal{O}}(\lambda)$. Next, we show that $\mathsf{S}^* \in \boldsymbol{S}_{\mathrm{diff}}(1)$. (See Appendix A for notation.) By applying the result of BCP [15] saying that any indistinguishability obfuscator is also a $\boldsymbol{S}_{\mathrm{diff}}(1)$-secure obfuscator, we get that $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf}^*,\mathsf{S}^*,\mathcal{O}}(\cdot)$ is negligible by the iO-security of Obf^*.

Given any PT difference adversary $\mathcal{D}$ against S^*, we build a PT adversary $\mathcal{F}$ against AIOWF[X]-security as follows:

$$\begin{array}{l}\underline{\text{Adversary } \mathcal{F}(1^\lambda, fk, y, a)} \\ d \leftarrow\!\!\$\ \{0,1\}\ ;\ \mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{1^\lambda, fk, y})\ ;\ \mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2) \\ aux \leftarrow (d, a)\ ;\ k' \leftarrow\!\!\$\ \mathcal{D}(\mathrm{C}_1, \mathrm{C}_0, aux)\ ;\ \text{return } k'\end{array}$$

Note that the constructed adversary has to guess the challenge bit d that is sampled in the circuit sampler S^*. The distinguisher is only guaranteed to behave in the same way when the guess is correct, hence a factor of $1/2$ is introduced. We have $\mathsf{Adv}^{\mathsf{ai\text{-}owf}}_{\mathsf{F},\mathsf{X},\mathcal{F}}(\lambda) \geq \mathsf{Adv}^{\mathsf{diff}}_{\mathsf{S}^*,\mathcal{D}}(\lambda)/2$, and hence the difference-security of S^* follows from the assumption that F is AIOWF[X]-secure. ∎

Does VGBO break AIPO? A natural question is whether our VGBO-based attacks on the AI-DHI assumptions from Theorems 3.1 and 3.2 can be extended to rule out AIPO itself, calling all our AIPO constructions above into question. At the moment we do not know how to provide such attacks. But we note that for each of these constructions as well as the construction of Wee [43] we can give VGBO attacks that rule out security if the underlying primitives satisfy certain verifiability conditions, akin to Theorem 3.2. Since there exist candidate choices of the primitives that do not satisfy these conditions we do not universally break the constructions, but this shows that one must be careful in instantiation and many natural choices will not co-exist with VGBO.

5 Impossibility results from iO

We refer to a symmetric encryption scheme as K-leakage-resilient if it retains security in the presence of any leakage about the key that leaves the key computationally unpredictable [29]. Such schemes have been designed in [29, 25]. Here, we extend the model by allowing the leakage to depend not just on the key but also on the message, still leaving the key computationally unpredictable. The extension seems innocuous, since the indistinguishability style formalizations used here already capture the adversary having some information about the message. But we show that the resulting KM-leakage-resilience is not achievable if iO for all circuits exists.

BFM [18] showed that UCE[$\mathbf{S}^{\mathrm{cup}}$]-security is not possible if iO exists. We improve this to show that UCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}$]-security is not possible if iO exists. We obtain this by giving a construction

of a KM-leakage-resilient symmetric encryption scheme from UCE[$\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt}$] and then invoking our above-mentioned result.

This shows that our result about KM-leakage-resilience has double utility. First it is of direct interest with regard to understanding what is and is not achievable in leakage-resilient cryptography. Second, it is a tool to obtain other negative relations with starting point iO: to rule out X we can show that X implies KM-leakage-resilient symmetric encryption.

We use standard definitions of UCE-secure function families [5], indistinguishability obfuscation [3, 30, 42, 15, 1] and pseudorandom generators [14, 47], as provided in Appendix A.

iO $\Rightarrow \neg$KM-LR-SE. A symmetric encryption scheme SE specifies the following. PT encryption algorithm $\mathsf{SE.Enc}$ takes 1^λ, a key $k \in \{0,1\}^{\mathsf{SE.kl}(\lambda)}$ and a message $m \in \{0,1\}^{\mathsf{SE.ml}(\lambda)}$ to return a ciphertext c, where $\mathsf{SE.kl}, \mathsf{SE.ml}\colon \mathbb{N} \to \mathbb{N}$ are the key length and message length functions of SE, respectively. Deterministic PT decryption algorithm $\mathsf{SE.Dec}$ takes $1^\lambda, k, c$ to return a plaintext $m \in \{0,1\}^{\mathsf{SE.ml}(\lambda)}$. Note that there is a key length but no prescribed key-generation algorithm. In Appendix E we define what it means for SE to be X-KM-leakage resilient where X is an auxiliary information generator. We say that SE is KM-leakage-resilient if it is X-KM-leakage resilient for all unpredictable, *uniform* X (cf. Section 2 for definition of latter). We also specify a weaker-than-normal correctness condition. Making the correctness and security requirements as weak as possible strengthens our result since it is negative. Also the weaker the notion, the easier to show that other notions imply it, making the result a more powerful tool towards obtaining further negative results. The following says that KM-leakage-resilient symmetric encryption is not achievable if iO and one-way functions exist.

Theorem 5.1 *Let SE be a symmetric encryption scheme. Let Obf be an indistinguishability obfuscator. Let R be a PR-secure PRG with $\mathsf{R.sl} = \mathsf{SE.ml}$. Assume that $2^{-\mathsf{SE.kl}(\lambda)}$ and $2^{-\mathsf{R.sl}(\lambda)}$ are negligible. Then there exists a uniform auxiliary information generator X such that the following holds: (1) X is unpredictable, but (2) SE is not X-KM-leakage resilient.*

The proof, in Appendix E, mainly follows [20] with some new elements. The idea is that the auxiliary information generator X picks a key k and message m uniformly and independently at random and lets C be the circuit that embeds k and the result y of the PRG on m. On input a ciphertext c, circuit C decrypts it under k and then checks that the PRG applied to the result equals y. The auxiliary information is an obfuscation $\overline{\rm C}$ of C. The attack showing claim (2) of Theorem 5.1 is straightforward but its analysis is more work and exploits the security of the PRG. The bulk of the proof is to show that iO-security of the obfuscator coupled with security of the PRG implies claim (1), namely the unpredictability of X. See Appendix E for details.

UCE[$\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt}$] $\Rightarrow$ **KM-LR-SE**. We give a construction of a KM-leakage resilient symmetric encryption scheme from a $\mathbf{UCE}[\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt}]$ family H, which will allow us to rule out such families under iO. Assume for simplicity that $\mathsf{H.il}$ is odd, and let $\ell = (\mathsf{H.il}-1)/2$. We call the symmetric encryption scheme $\mathsf{SE} = \mathbf{H}\&\mathbf{C}[\mathsf{H}]$ that we associate to H the Hash-and-Check scheme. It is defined as follows. Let $\mathsf{SE.kl}(\lambda) = \mathsf{SE.ml}(\lambda) = \ell(\lambda)$ for all $\lambda \in \mathbb{N}$. Let the encryption and decryption algorithms be as follows:

Algorithm $\mathsf{SE.Enc}(1^\lambda, k, m)$	Algorithm $\mathsf{SE.Dec}(1^\lambda, k, (hk, \mathbf{y}))$
$hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	For $i = 1, \ldots, \lvert\mathbf{y}\rvert$ do
For $i = 1, \ldots, \lvert m\rvert$ do	If $(\mathsf{H.Ev}(1^\lambda, hk, k\Vert 1\Vert\langle i\rangle_{\ell(\lambda)}) = \mathbf{y}[i])$
$\quad \mathbf{y}[i] \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k\Vert m[i]\Vert\langle i\rangle_{\ell(\lambda)})$	Then $m[i] \leftarrow 1$ else $m[i] \leftarrow 0$
Return $(hk, \mathbf{y})$	Return m

Here $\langle i \rangle_{\ell(\lambda)} = 1^i 0^{\ell(\lambda)-i}$ denotes a particular, convenient encoding of integer $i \in \{1, \ldots, \ell(\lambda)\}$ as a string of $\ell(\lambda)$ bits, and $m[i]$ denotes the i-th bit of m. The ciphertext $(hk, \mathbf{y})$ consists of a key hk for H chosen randomly and anew at each encryption, together with the vector $\mathbf{y}$ whose i-th entry is the hash of the i-th message bit along with the key and index i. This scheme will have perfect correctness if H is injective, but we do not want to assume this. The following theorem says that the scheme is KM-leakage resilient and also has (somewhat better than) weak correctness under UCE-security of H. The definitions of KM-leakage resilience and $\mathsf{Adv}^{\mathsf{dec}}_{\mathsf{SE}}(\cdot)$ are in Appendix E. The proof of the following is in Appendix F.

Theorem 5.2 *Let* H *be a family of functions that is* UCE$[\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt}]$*-secure. Assume* $\mathsf{H.il}(\cdot) \in \Omega((\cdot)^{\epsilon})$ *for some constant* $\epsilon > 0$ *and* $2^{-\mathsf{H.ol}(\cdot)}$ *is negligible. Let* $\mathsf{SE} = \mathbf{H}\&\mathbf{C}[\mathsf{H}]$*. Then (1) symmetric encryption scheme* SE *is KM-leakage resilient, and (2)* $1 - \mathsf{Adv}^{\mathsf{dec}}_{\mathsf{SE}}(\cdot)$ *is negligible.*

$\underline{\mathbf{iO} \Rightarrow \neg\mathbf{UCE}[\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt}]}$. In the BFM [18] iO-based attack on UCE$[\mathbf{S}^{\rm cup}]$, the source builds a circuit which embeds an oracle query x and its answer y, and outputs an obfuscation of this circuit in the leakage. Splitting is a restriction on sources introduced in BHK [5] with the aim of preventing such attacks. A split source cannot build the BFM circuit because the split structure denies it the ability to leak information that depends both on a query and its answer. Thus, the BFM attack does not work for UCE$[\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt}]$. However, we show that in fact UCE$[\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt}]$-security is still not achievable assuming iO. This is now a simple corollary of Theorems 5.1 and 5.2 that in particular was the motivation for the latter:

Theorem 5.3 *Let* H *be a family of functions such that* $\mathsf{H.il}(\cdot) \in \Omega((\cdot)^{\epsilon})$ *for some constant* $\epsilon > 0$ *and* $2^{-\mathsf{H.ol}(\cdot)}$ *is negligible. Assume the existence of an indistinguishability obfuscator and a one-way function. Then* H *is not* UCE$[\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt}]$*-secure.*

We believe this result is interesting towards understanding the achievability of different forms of UCE and the effectiveness of different restrictions on sources (in this case, splitting) towards this end, but we note that we are not aware of any applications of UCE$[\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt}]$. Our results do not threaten UCE$[\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt} \cap \mathbf{S}^{1}]$, which we assume in Theorem 4.2 and is also assumed in [5], or UCE$[\mathbf{S}^{\rm sup}]$, which is assumed in [5, 4].

BM [21] show that UCE$[\mathbf{S}^{\rm cup} \cap \mathbf{S}^{\rm splt} \cap \mathbf{S}^{1}]$-security and UCE$[\mathbf{S}^{\rm sup} \cap \mathbf{S}^{\rm splt}]$-security are achievable assuming iO and AIPO. Our negative result of Theorem 5.3 does not contradict this, and in fact complements it to give a full picture of the achievability of security for split sources.

Acknowledgments

We thank Huijia Lin for discussions and insights.

References

[1] P. Ananth, D. Boneh, S. Garg, A. Sahai, and M. Zhandry. Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689, 2013. `http://eprint.iacr.org/2013/689`. 14, 16, 21

[2] B. Barak, S. Garg, Y. T. Kalai, O. Paneth, and A. Sahai. Protecting obfuscation against algebraic attacks. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 221–238. Springer, May 2014. 3

[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 J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Aug. 2001. 3, 5, 14, 16, 21

[4] M. Bellare and V. T. Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 627–656. Springer, Apr. 2015. 3, 5, 6, 17

[5] M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs. Cryptology ePrint Archive, Report 2013/424, 2013. Preliminary version in CRYPTO 2013. 5, 6, 7, 12, 16, 17, 20, 21

[6] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, May / June 2006. 7

[7] M. Bellare, I. Stepanovs, and S. Tessaro. Poly-many hardcore bits for any one-way function and a framework for differing-inputs obfuscation. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 102–121. Springer, Dec. 2014. 21

[8] N. Bitansky and R. Canetti. On strong simulation and composable point obfuscation. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 520–537. Springer, Aug. 2010. 3, 5, 8

[9] N. Bitansky, R. Canetti, Y. T. Kalai, and O. Paneth. On virtual grey box obfuscation for general circuits. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 108–125. Springer, Aug. 2014. 3, 8

[10] N. Bitansky, R. Canetti, Y. T. Kalai, and O. Paneth. On virtual grey box obfuscation for general circuits. Cryptology ePrint Archive, Report 2014/554, 2014. `http://eprint.iacr.org/2014/554`. 3

[11] N. Bitansky, R. Canetti, O. Paneth, and A. Rosen. On the existence of extractable one-way functions. In D. B. Shmoys, editor, *46th ACM STOC*, pages 505–514. ACM Press, May / June 2014. 3, 5

[12] N. Bitansky and O. Paneth. Point obfuscation and 3-round zero-knowledge. In R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 190–208. Springer, Mar. 2012. 3, 4, 5, 6, 9, 12, 14

[13] N. Bitansky and O. Paneth. On the impossibility of approximate obfuscation and applications to resettable cryptography. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th ACM STOC*, pages 241–250. ACM Press, June 2013. 3

[14] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. 16, 22

[15] E. Boyle, K.-M. Chung, and R. Pass. On extractability obfuscation. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 52–73. Springer, Feb. 2014. 14, 15, 16, 21, 22

[16] Z. Brakerski and G. N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 1–25. Springer, Feb. 2014. 3

[17] Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 543–560. Springer, Aug. 2011. 5, 6, 13, 24

[18] C. Brzuska, P. Farshim, and A. Mittelbach. Indistinguishability obfuscation and UCEs: The case of computationally unpredictable sources. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 188–205. Springer, Aug. 2014. 3, 5, 6, 15, 17, 20

[19] C. Brzuska, P. Farshim, and A. Mittelbach. Random-oracle uninstantiability from indistinguishability obfuscation. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 428–455. Springer, Mar. 2015. 3, 5

[20] C. Brzuska and A. Mittelbach. Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 142–161. Springer, Dec. 2014. 3, 5, 6, 16, 26

[21] C. Brzuska and A. Mittelbach. Using indistinguishability obfuscation via UCEs. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 122–141. Springer, Dec. 2014. 5, 6, 13, 17, 21

[22] R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In B. S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 455–469. Springer, Aug. 1997. 3, 4, 5, 9, 12

[23] R. Canetti and R. R. Dakdouk. Extractable perfectly one-way functions. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 449–460. Springer, July 2008. 14

[24] R. Canetti and R. R. Dakdouk. Obfuscating point functions with multibit output. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 489–508. Springer, Apr. 2008. 5

[25] R. Canetti, Y. T. Kalai, M. Varia, and D. Wichs. On symmetric encryption and point obfuscation. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 52–71. Springer, Feb. 2010. 5, 15

[26] R. Canetti, D. Micciancio, and O. Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *30th ACM STOC*, pages 131–140. ACM Press, May 1998. 12

[27] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé. Cryptanalysis of the multilinear map over the integers. Cryptology ePrint Archive, Report 2014/906, 2014. `http://eprint.iacr.org/2014/906`. 3

[28] J.-S. Coron, T. Lepoint, and M. Tibouchi. Cryptanalysis of two candidate fixes of multilinear maps over the integers. Cryptology ePrint Archive, Report 2014/975, 2014. `http://eprint.iacr.org/2014/975`. 3

[29] Y. Dodis, Y. T. Kalai, and S. Lovett. On cryptography with auxiliary input. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 621–630. ACM Press, May / June 2009. 5, 15

[30] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, Oct. 2013. 3, 5, 14, 16, 21

[31] S. Garg, C. Gentry, S. Halevi, and D. Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 518–535. Springer, Aug. 2014. 3

[32] C. Gentry, S. Halevi, H. K. Maji, and A. Sahai. Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero. Cryptology ePrint Archive, Report 2014/929, 2014. `http://eprint.iacr.org/2014/929`. 3

[33] C. Gentry, A. Lewko, A. Sahai, and B. Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. Cryptology ePrint Archive, Report 2014/309, 2014. `http://eprint.iacr.org/2014/309`. 3

[34] S. Goldwasser and Y. T. Kalai. On the impossibility of obfuscation with auxiliary input. In *46th FOCS*, pages 553–562. IEEE Computer Society Press, Oct. 2005. 3, 5, 6, 12

[35] M. D. Green, J. Katz, A. J. Malozemoff, and H.-S. Zhou. A unified approach to idealized model separations via indistinguishability obfuscation. Cryptology ePrint Archive, Report 2014/863, 2014. `http://eprint.iacr.org/2014/863`. 3, 5

[36] S. Hada. Zero-knowledge and code obfuscation. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 443–457. Springer, Dec. 2000. 3

[37] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. 28

[38] H. T. Lee and J. H. Seo. Security analysis of multilinear maps over the integers. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 224–240. Springer, Aug. 2014. 3

[39] B. Lynn, M. Prabhakaran, and A. Sahai. Positive results and techniques for obfuscation. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 20–39. Springer, May 2004. 5, 12

[40] T. Matsuda and G. Hanaoka. Chosen ciphertext security via point obfuscation. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 95–120. Springer, Feb. 2014. 5, 6

[41] R. Pass, K. Seth, and S. Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 500–517. Springer, Aug. 2014. 3

[42] A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In D. B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014. 14, 16, 21

[43] H. Wee. On obfuscating point functions. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 523–532. ACM Press, May 2005. 5, 6, 12, 14, 15

[44] H. Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 246–262. Springer, Apr. 2012. 13, 24

[45] D. Wichs. Barriers in cryptography with weak, correlated and leaky sources. In R. D. Kleinberg, editor, *ITCS 2013*, pages 111–126. ACM, Jan. 2013. 5

[46] X. Xie, R. Xue, and R. Zhang. Deterministic public key encryption and identity-based encryption from lattices in the auxiliary-input setting. In I. Visconti and R. D. Prisco, editors, *SCN 12*, volume 7485 of *LNCS*, pages 1–18. Springer, Sept. 2012. 13, 24

[47] A. C.-C. Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, Nov. 1982. 16, 22

A Standard definitions

UCE security. We recall the Universal Computational Extractor (UCE) framework of BHK [5]. Let H be a family of functions. Let $\mathcal{S}$ be an adversary called the *source* and $\mathcal{D}$ an adversary called the *distinguisher*. We associate to them and H the game $\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)$ in the left panel of Fig. 6. The source has access to an oracle HASH and we require that any query x made to this oracle have length $\mathsf{H.il}(\lambda)$. When the challenge bit b is 1 (the "real" case) the oracle responds via $\mathsf{H.Ev}$ under a key hk that is chosen by the game and *not* given to the source. When $b = 0$ (the "random" case) it responds as a random oracle. The source then leaks a string L to its accomplice distinguisher. The latter *does* get the key hk as input and must now return its guess $b' \in \{0,1\}$ for b. The game returns true iff $b' = b$, and the uce-advantage of $(\mathcal{S},\mathcal{D})$ is defined for $\lambda \in \mathbb{N}$ via $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\lambda) = 2\Pr[\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)] - 1$. If $\mathbf{S}$ is a class (set) of sources, we say that H is UCE[$\mathbf{S}$]-secure if $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\cdot)$ is negligible for all sources $\mathcal{S} \in \mathbf{S}$ and all PT distinguishers $\mathcal{D}$.

It is easy to see that UCE[$\mathbf{S}$]-security is not achievable if $\mathbf{S}$ is the class of all PT sources [5]. To obtain meaningful notions of security, BHK [5] impose restrictions on the source. A central restriction is unpredictability. A source is unpredictable if it is hard to guess the source's HASH queries even given the leakage, in the *random case* of the UCE game. Formally, let $\mathcal{S}$ be a source and $\mathcal{P}$ an adversary called a predictor and consider game $\mathrm{PRED}^{\mathcal{P}}_{\mathcal{S}}(\lambda)$ in the middle panel of Fig. 6. For $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{pred}}_{\mathcal{S},\mathcal{P}}(\lambda) = \Pr[\mathrm{PRED}^{\mathcal{P}}_{\mathcal{S}}(\lambda)]$. We say that $\mathcal{S}$ is computationally unpredictable if $\mathsf{Adv}^{\mathsf{pred}}_{\mathcal{S},\mathcal{P}}(\cdot)$ is negligible for all PT predictors $\mathcal{P}$, and let $\mathbf{S}^{\mathrm{cup}}$ be the class of all PT computationally unpredictable sources. We say that $\mathcal{S}$ is statistically unpredictable if $\mathsf{Adv}^{\mathsf{pred}}_{\mathcal{S},\mathcal{P}}(\cdot)$ is negligible for all (not necessarily PT) predictors $\mathcal{P}$, and let $\mathbf{S}^{\mathrm{sup}} \subseteq \mathbf{S}^{\mathrm{cup}}$ be the class of all PT statistically unpredictable sources.

BFM [18] show that UCE[$\mathbf{S}^{\mathrm{cup}}$]-security is not achievable assuming that indistinguishability obfuscation is possible. This has lead applications to either be based on UCE[$\mathbf{S}^{\mathrm{sup}}$] or on subsets of UCE[$\mathbf{S}^{\mathrm{cup}}$], meaning to impose further restrictions on the source. UCE[$\mathbf{S}^{\mathrm{sup}}$], introduced in [5, 18],

Game $\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)$	Game $\mathrm{PRED}^{\mathcal{P}}_{\mathcal{S}}(\lambda)$	Source $\mathcal{S}^{\mathrm{HASH}}(1^\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}$; $hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	$X \leftarrow \emptyset$	$(L_0, \mathbf{x}) \leftarrow\!\!\$\ \mathcal{S}_0(1^\lambda)$
$L \leftarrow\!\!\$\ \mathcal{S}^{\mathrm{HASH}}(1^\lambda)$	$L \leftarrow\!\!\$\ \mathcal{S}^{\mathrm{HASH}}(1^\lambda)$	For $i = 1, \ldots, \lvert\mathbf{x}\rvert$ do
$b' \leftarrow\!\!\$\ \mathcal{D}(1^\lambda, hk, L)$	$x' \leftarrow\!\!\$\ \mathcal{P}(1^\lambda, L)$	$\mathbf{y}[i] \leftarrow\!\!\$\ \mathrm{HASH}(\mathbf{x}[i])$
Return $(b' = b)$	Return $(x' \in X)$	$L_1 \leftarrow\!\!\$\ \mathcal{S}_1(1^\lambda, \mathbf{y})$
$\underline{\mathrm{HASH}(x)}$	$\underline{\mathrm{HASH}(x)}$	$L \leftarrow (L_0, L_1)$
If $T[x] = \bot$ then	If $T[x] = \bot$ then	Return L
If $b = 0$ then $T[x] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$	$T[x] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$	
Else $T[x] \leftarrow \mathsf{H.Ev}(1^\lambda, hk, x)$	$X \leftarrow X \cup \{x\}$	
Return $T[x]$	Return $T[x]$	

Figure 6: Games defining UCE security of function family H, unpredictability of source $\mathcal{S}$, and the split source $\mathcal{S} = \mathsf{Splt}[\mathcal{S}_0, \mathcal{S}_1]$ associated to $\mathcal{S}_0, \mathcal{S}_1$.

seems at this point to be a viable assumption. In order to restrict the computational case, one can consider split sources as defined in BHK [5]. Let $\mathcal{S}_0, \mathcal{S}_1$ be algorithms, neither of which have access to any oracles. The *split source* $\mathcal{S} = \mathsf{Splt}[\mathcal{S}_0, \mathcal{S}_1]$ associated to $\mathcal{S}_0, \mathcal{S}_1$ is defined in the right panel of Fig. 6. Algorithm $\mathcal{S}_0$ returns a pair $(L_0, \mathbf{x})$. Here $\mathbf{x}$ is a vector over $\{0,1\}^{\mathsf{H.il}(\lambda)}$ all of whose entries are required to be distinct. The first adversary creates the oracle queries for the source $\mathcal{S}$, the latter making these queries and passing the replies to the second adversary to get the leakage. In this way, neither $\mathcal{S}_0$ nor $\mathcal{S}_1$ have an input-output pair from the oracle, limiting their ability to create leakage useful to the distinguisher. A source $\mathcal{S}$ is said to belong to the class $\mathbf{S}^{\mathrm{splt}}$ if there exist PT $\mathcal{S}_0, \mathcal{S}_1$ such that $\mathcal{S} = \mathsf{Splt}[\mathcal{S}_0, \mathcal{S}_1]$, meaning is defined as above. The class of interest is now $\mathrm{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$, meaning UCE-security for computationally unpredictable, split sources.

Another way to restrict a UCE source is by limiting the number of queries it can make. Let $\mathbf{S}^q$ be the class of sources making $q(\cdot)$ oracle queries. This allows to consider $\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1$, a class of computationally unpredictable split sources that make a single query. BM [21] show that $\mathrm{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1]$-security is achievable assuming iO and AIPO.

Indistinguishability obfuscation. We use the definitional framework BST [7] which allows us to capture indistinguishabilty obfuscation, differing-inputs obfuscation and other variant notions via classes of circuit samplers. Let Obf be an obfuscator. A *sampler* in this context is a PT algorithm S that on input 1^λ returns a triple $(\mathrm{C}_0, \mathrm{C}_1, aux)$ where $\mathrm{C}_0, \mathrm{C}_1$ are circuits of the same size, number of inputs and number of outputs, and aux is a string. If $\mathcal{O}$ is an adversary and $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S},\mathcal{O}}(\lambda) = 2\Pr[\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathsf{S}}(\lambda)] - 1$ where game $\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathsf{S}}(\lambda)$ is defined in Fig. 7. Now let $\boldsymbol{S}$ be a class (set) of circuit samplers. We say that Obf is $\boldsymbol{S}$-secure if $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S},\mathcal{O}}(\cdot)$ is negligible for every PT adversary $\mathcal{O}$ and every circuit sampler $\mathsf{S} \in \boldsymbol{S}$. We say that circuit sampler S produces equivalent circuits if there exists a negligible function ν such that $\Pr\left[\ \mathrm{C}_0 \equiv \mathrm{C}_1\ :\ (\mathrm{C}_0, \mathrm{C}_1, aux) \leftarrow\!\!\$\ \mathsf{S}(1^\lambda)\ \right] \geq 1 - \nu(\lambda)$ for all $\lambda \in \mathbb{N}$. Let $\boldsymbol{S}_{\mathrm{eq}}$ be the class of all circuit samplers that produce equivalent circuits. We say that Obf is an indistinguishability obfuscator if it is $\boldsymbol{S}_{\mathrm{eq}}$-secure [3, 30, 42].

We say that a circuit sampler S is difference secure if $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{S},\mathcal{D}}(\cdot)$ is negligible for every PT adversary $\mathcal{D}$, where $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{S},\mathcal{D}}(\lambda) = \Pr[\mathrm{DIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda)]$ and game $\mathrm{DIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda)$ is defined in Fig. 7. Difference security of S means that given $\mathrm{C}_0, \mathrm{C}_1, aux$ it is hard to find an input on which the circuits differ [3, 15, 1]. Let $\boldsymbol{S}_{\mathrm{diff}}$ be the class of all difference-secure circuit samplers. We say that circuit sampler S produces d-differing circuits, where d: $\mathbb{N} \to \mathbb{N}$, if C_0 and C_1 differ on at most $d(\lambda)$ inputs with an overwhelming probability over $(\mathrm{C}_0, \mathrm{C}_1, aux) \leftarrow\!\!\$\ \mathsf{S}(1^\lambda)$ for all $\lambda \in \mathbb{N}$. Let $\boldsymbol{S}_{\mathrm{diff}}(d)$ be the class of all

Game $\mathrm{DIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda)$	Game $\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathsf{S}}(\lambda)$	Game $\mathrm{PRG}^{\mathcal{R}}_{\mathsf{R}}(\lambda)$
$(\mathrm{C}_0, \mathrm{C}_1, aux) \leftarrow\!\!\$\ \mathsf{S}(1^\lambda)$	$b \leftarrow\!\!\$\ \{0,1\}$	$b \leftarrow\!\!\$\ \{0,1\}$
$x \leftarrow\!\!\$\ \mathcal{D}(\mathrm{C}_0, \mathrm{C}_1, aux)$	$(\mathrm{C}_0, \mathrm{C}_1, aux) \leftarrow\!\!\$\ \mathsf{S}(1^\lambda)$	$m \leftarrow\!\!\$\ \{0,1\}^{\mathsf{R.sl}(\lambda)}$
Return $(\mathrm{C}_0(x) \neq \mathrm{C}_1(x))$	$\overline{\mathrm{C}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathrm{C}_b)$	$y_1 \leftarrow \mathsf{R.Ev}(1^\lambda, m)$
	$b' \leftarrow\!\!\$\ \mathcal{O}(1^\lambda, \overline{\mathrm{C}}, aux)$	$y_0 \leftarrow\!\!\$\ \{0,1\}^{2\cdot\mathsf{R.sl}(\lambda)}$
	Return $(b = b')$	$b' \leftarrow\!\!\$\ \mathcal{R}(1^\lambda, y_b)$
		Return $(b = b')$

Figure 7: Games defining difference-security of circuit sampler S, iO-security of obfuscator Obf and PR-security of pseudorandom generator R.

Games G_0–G_1

$b \leftarrow\!\!\$\ \{0,1\}$; $(k_0, m_0, a_0) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$; $(k_1, m_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$
$\langle\mathbb{G}\rangle \leftarrow\!\!\$\ \mathsf{GG}(1^\lambda)$; $g \leftarrow\!\!\$\ \mathbb{G}^*$
  $K \leftarrow g^{k_b}$ // G_0
  $K \leftarrow\!\!\$\ \mathbb{G}$ // G_1
$\overline{\mathrm{P}} \leftarrow \mathrm{C}_{\langle\mathbb{G}\rangle, g, K}$; $b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathrm{P}}, a_1)$; Return $(b = b')$

Circuit $\mathrm{C}_{\langle\mathbb{G}\rangle, g, K}(k)$
If $(g^k = K)$ then return 1
Else return 0

Figure 8: **Games for proof of Theorem 4.1.**

difference-secure circuit samplers that produce d-differing circuits, so that $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}_{\mathrm{diff}}(d) \subseteq \boldsymbol{S}_{\mathrm{diff}}$. The interest of this definition is the following result of BCP [15] that we use:

Proposition A.1 *If d is a polynomial then any $\boldsymbol{S}_{\mathrm{eq}}$-secure circuit obfuscator is also a $\boldsymbol{S}_{\mathrm{diff}}(d)$-secure circuit obfuscator.*

PRGs. A pseudorandom generator R [14, 47] specifies a deterministic PT algorithm $\mathsf{R.Ev}$ where $\mathsf{R.Ev}(1^\lambda, \cdot)$: $\{0,1\}^{\mathsf{R.sl}(\lambda)} \to \{0,1\}^{2\cdot\mathsf{R.sl}(\lambda)}$ for all $\lambda \in \mathbb{N}$, where $\mathsf{R.sl}$: $\mathbb{N} \to \mathbb{N}$ is the seed length function of R. We say that R is PR-secure if the function $\mathsf{Adv}^{\mathsf{pr}}_{\mathsf{R},\mathcal{R}}(\cdot)$ is negligible for every PT adversary $\mathcal{R}$, where for $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{pr}}_{\mathsf{R},\mathcal{R}}(\lambda) = 2\Pr[\mathrm{PRG}^{\mathcal{R}}_{\mathsf{R}}(\lambda)] - 1$ and game PRG is specified in Fig. 7.

B Proof of Theorem 4.1

First, we claim that Obf satisfies the correctness condition for all $\mathbf{I}_k$ with $k \in \{0,1\}^{\lambda-1}$. Note that GG on input 1^λ returns a description $\langle\mathbb{G}\rangle$ of a cyclic group $\mathbb{G}$ with an order greater than $2^{\lambda-1}$. So for any generator $g \in \mathrm{Gen}(\mathbb{G})$, the exponentiation g^k defined for $k \in \{0,1\}^{\lambda-1}$ is an injective function. The correctness follows.

We now prove the security of Obf. Let $\mathcal{A}$ be a PT adversary. Let X be any unpredictable auxiliary information generator for GG. Consider the games and associated circuit of Fig. 8. Lines not annotated with comments are common to both games. Game G_0 is equivalent to game $\mathrm{AIPO}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$, whereas $\Pr[\mathrm{G}_1] = \frac{1}{2}$ because no inputs to the adversary $\mathcal{A}$ depend on the challenge bit b in G_1. Hence,

$$\mathsf{Adv}^{\mathsf{aipo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda) = 2(\Pr[\mathrm{G}_1] + \Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]) - 1 = 2(\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]). \quad (3)$$

We now show that $\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]$ is negligible, hence proving the theorem.

Consider the following PT adversary $\mathcal{B}$ against the AIDHI2-security of GG with respect to X:

$$\begin{array}{l}\underline{\text{Adversary } \mathcal{B}(1^\lambda, \langle\mathbb{G}\rangle, g, K, a_1)} \\ d \leftarrow\!\!\$\ \{0,1\}\ ;\ (k_0, m_0, a_0) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda) \\ \overline{\mathrm{P}} \leftarrow \mathrm{C}_{\langle\mathbb{G}\rangle,g,K}\ ;\ d' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathrm{P}}, a_d) \\ \text{If } (d = d') \text{ then return 1 else return 0}\end{array}$$

Let b denote the challenge bit in game $\mathrm{AIDHI2}^{\mathcal{B}}_{\mathsf{GG},\mathsf{X}}(\lambda)$, and let b' denote the bit returned by $\mathcal{B}$ in the same game. We claim that

$$\Pr[\,b' = 1 \,|\, b = 1\,] = \Pr[\mathrm{G}_0] \qquad \text{and} \qquad \Pr[\,b' = 1 \,|\, b = 0\,] = \Pr[\mathrm{G}_1]\ .$$

We have $\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1] = \mathsf{Adv}^{\mathsf{aidhi2}}_{\mathsf{GG},\mathsf{X},\mathcal{B}}(\lambda)$, which is negligible by the assumed AI-DHI2 security of GG. The AIPO[X] security of Obf now follows from Equation (3).

C Proof of Theorem 4.2

Correctness of the obfuscator, meaning that the output of $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$ is a point circuit with target k, follows from the assumed injectivity of H. We now prove AIPO security under the UCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1$] assumption on H. Let $\mathbf{S} = \mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1$. Let X be any unpredictable auxiliary information generator with $\mathsf{X.tl} = \mathsf{H.il}$. Given a PT adversary $\mathcal{A}$ against the AIPO[X]-security of Obf, we construct a split source $\mathcal{S} = \mathsf{Splt}[\mathcal{S}_0, \mathcal{S}_1] \in \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^1$ and a PT distinguisher $\mathcal{D}$ against the UCE[$\mathbf{S}$]-security of H as follows:

$$\begin{array}{l|l|l}
\underline{\text{Algorithm } \mathcal{S}_0(1^\lambda)} & \underline{\text{Algorithm } \mathcal{S}_1(1^\lambda, y)} & \underline{\text{Distinguisher } \mathcal{D}(1^\lambda, hk, L)} \\
d \leftarrow\!\!\$\ \{0,1\} & \text{return } y & ((d, a_1), y) \leftarrow L \\
(k_0, m_0, a_0) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda) & \underline{\text{Circuit } \mathrm{C}_{1^\lambda,hk,y}(k')} & \overline{\mathrm{P}} \leftarrow \mathrm{C}_{1^\lambda,hk,y} \\
(k_1, m_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda) & y' \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k') & d' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathrm{P}}, a_1) \\
L_0 \leftarrow (d, a_1)\ ;\ \mathbf{x}[1] \leftarrow k_b & \text{If } (y = y') \text{ then return 1} & \text{If } (d = d') \text{ then return 1} \\
\text{return } (L_0, \mathbf{x}) & \text{Else return 0} & \text{Else return 0}
\end{array}$$

Let b denote the challenge bit in game $\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)$, and let b' denote the bit returned by $\mathcal{D}$ in the same game. We claim that

$$\Pr[\,b' = 1 \,|\, b = 1\,] = \Pr[\mathrm{AIPO}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)] \qquad \text{and} \qquad \Pr[\,b' = 1 \,|\, b = 0\,] = \frac{1}{2}\ .$$

The first equation holds by construction. The second equation is true because the value y is chosen uniformly at random, and hence $\mathcal{D}$ runs $\mathcal{A}$ with inputs $\overline{\mathrm{P}}$ and a_1 independent of the challenge bit d. It follows that $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\lambda) = \mathsf{Adv}^{\mathsf{aipo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda)/2$.

To conclude the proof, it remains to show that $\mathcal{S} \in \mathbf{S}^{\mathrm{cup}}$. Let $\mathcal{P}$ be a PT adversary against the computational unpredictability of $\mathcal{S}$. Then we build a PT adversary $\mathcal{Q}$ against the unpredictability of X as follows:

$$\begin{array}{l}\underline{\text{Adversary } \mathcal{Q}(1^\lambda, a)} \\ d \leftarrow\!\!\$\ \{0,1\}\ ;\ y \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)} \\ L \leftarrow ((d, a), y)\ ;\ k' \leftarrow\!\!\$\ \mathcal{P}(1^\lambda, L) \\ \text{Return } k'\end{array}$$

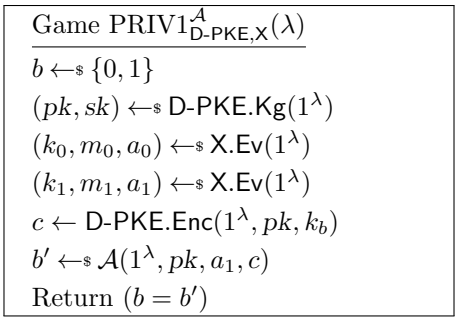


Figure 9: Game defining PRIV1-security of auxiliary-input deterministic public-key encryption scheme D-PKE.

We have $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda) = \mathsf{Adv}^{\mathsf{pred}}_{\mathcal{S},\mathcal{P}}(\lambda)$. Since X is unpredictable, it follows that $\mathcal{S}$ is also computationally unpredictable. And thus, by the above, if H is UCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1}$]-secure, then Obf is AIPO[X]-secure.

D PRIV1-AI-DPKE ⇒ AIPO

The study of deterministic public-key encryption schemes that are secure with respect to auxiliary inputs (AI-DPKE) was initiated in [17]. As a result, [17] presented AI-DPKE constructions based on d-linear assumption and subgroup indistinguishability assumptions. Subsequently, [44] provided a unifying framework for constructing AI-DPKE schemes from a number of standard assumptions, encompassing both constructions from [17] and also achieving an LWE-based construction. Another LWE-based construction was concurrently suggested in [46].

In Section 4, we construct an AIPO-secure point-function obfuscator from an AI-DPKE scheme that is secure for a single message. The security notion we require is slightly weaker than the original PRIV1 notion for AI-DPKE that was defined in [17]. Specifically, we require both challenge messages to be from the same efficiently samplable distribution of plaintext messages, whereas [17] also require the scheme to be secure with respect to different message distributions.

Definition. A deterministic public-key encryption scheme D-PKE specifies the following. PT key generation algorithm $\mathsf{D\text{-}PKE.Kg}$ takes 1^λ to return a public encryption key pk and a secret decryption key sk. Deterministic PT encryption algorithm $\mathsf{D\text{-}PKE.Enc}$ takes 1^λ, pk and a message $m \in \{0,1\}^{\mathsf{D\text{-}PKE.ml}(\lambda)}$ to return a ciphertext c, where $\mathsf{D\text{-}PKE.ml}\colon \mathbb{N} \to \mathbb{N}$ is the message length function associated to D-PKE. Deterministic PT decryption algorithm $\mathsf{D\text{-}PKE.Dec}$ takes 1^λ, sk, c to return a plaintext message $m \in \{0,1\}^{\mathsf{D\text{-}PKE.ml}(\lambda)}$. Correctness requires that for all $\lambda \in \mathbb{N}$, all $(pk, sk) \in [\mathsf{D\text{-}PKE.Kg}(1^\lambda)]$ and all $m \in \{0,1\}^{\mathsf{D\text{-}PKE.ml}(\lambda)}$ we have $\mathsf{D\text{-}PKE.Dec}(1^\lambda, sk, \mathsf{D\text{-}PKE.Enc}(1^\lambda, pk, m)) = m$.

Let D-PKE be a deterministic public-key encryption scheme, and let X be an auxiliary information generator with $\mathsf{X.tl} = \mathsf{D\text{-}PKE.ml}$. We say that D-PKE is PRIV1[X]-secure if $\mathsf{Adv}^{\mathsf{priv1}}_{\mathsf{D\text{-}PKE},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for all PT adversaries $\mathcal{A}$, where $\mathsf{Adv}^{\mathsf{priv1}}_{\mathsf{D\text{-}PKE},\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{PRIV1}^{\mathcal{A}}_{\mathsf{D\text{-}PKE},\mathsf{X}}(\lambda)] - 1$ and game $\mathrm{PRIV1}^{\mathcal{A}}_{\mathsf{D\text{-}PKE},\mathsf{X}}(\lambda)$ is defined in Fig. 9. We say that D-PKE is PRIV1-secure if it is PRIV1[X]-secure for all unpredictable X.

Proof of Theorem 4.3. The correctness of Obf follows from the decryption correctness of D-PKE, however it does not require the decryption algorithm $\mathsf{D\text{-}PKE.Dec}$ to be PT. We now prove that Obf is AIPO-secure. Let $\mathcal{A}$ be a PT adversary. Let X be any unpredictable auxiliary information

Game $\mathrm{IND}^{\mathcal{A}}_{\mathsf{SE},\mathsf{X}}(\lambda)$	Game $\mathrm{DEC}_{\mathsf{SE}}(\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}$	$k \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.kl}(\lambda)}$
$(k, m_1, a) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$m \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.ml}(\lambda)}$
$m_0 \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.ml}(\lambda)}$	$c \leftarrow\!\!\$\ \mathsf{SE.Enc}(1^\lambda, k, m)$
$c \leftarrow\!\!\$\ \mathsf{SE.Enc}(1^\lambda, k, m_b)$	$m' \leftarrow \mathsf{SE.Dec}(1^\lambda, k, c)$
$b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, a, c)$	Return $(m = m')$
Return $(b = b')$	

Figure 10: Games defining X-KM-leakage resilience of symmetric encryption scheme SE and decryption correctness of symmetric encryption scheme SE.

---

generator with $\mathsf{X.tl} = \mathsf{D\text{-}PKE.ml}$. Consider the following PT adversary $\mathcal{B}$ against the PRIV1[X] security of D-PKE:

$$\begin{array}{l}\underline{\text{Adversary } \mathcal{B}(1^\lambda, pk, a, c)} \\ \overline{\mathrm{P}} \leftarrow \mathrm{C}_{1^\lambda, pk, c}\ ;\ b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathrm{P}}, a) \\ \text{Return } b'\end{array}$$

By construction, we have

$$\mathsf{Adv}^{\mathsf{aipo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{priv1}}_{\mathsf{D\text{-}PKE},\mathsf{X},\mathcal{B}}(\lambda)\ .$$

The AIPO[X] security of Obf follows from the assumed PRIV1-security of D-PKE.

E Impossibility of KM-leakage-resilient encryption

We define KM-leakage-resilient symmetric encryption and prove Theorem 5.1.

Definitions. We define security of symmetric encryption in the presence of leakage about the key and the message. A symmetric encryption scheme SE specifies the following. PT encryption algorithm $\mathsf{SE.Enc}$ takes 1^λ, a key $k \in \{0,1\}^{\mathsf{SE.kl}(\lambda)}$ and a message $m \in \{0,1\}^{\mathsf{SE.ml}(\lambda)}$ to return a ciphertext c, where $\mathsf{SE.kl}, \mathsf{SE.ml}$: $\mathbb{N} \to \mathbb{N}$ are the key length and message length functions of SE, respectively. Deterministic PT decryption algorithm $\mathsf{SE.Dec}$ takes $1^\lambda, k, c$ to return a plaintext $m \in \{0,1\}^{\mathsf{SE.ml}(\lambda)}$. Both security and correctness will be non-standard and are discussed in turn. Note that there is a key length but no prescribed key-generation algorithm.

For security, let X be an auxiliary information generator with $\mathsf{X.tl} = \mathsf{SE.kl}$ and $\mathsf{X.pl} = \mathsf{SE.ml}$. Consider game $\mathrm{IND}^{\mathcal{A}}_{\mathsf{SE},\mathsf{X}}(\lambda)$ of Fig. 10 associated to SE, X and adversary $\mathcal{A}$. The message m_0 is picked uniformly at random. The adversary $\mathcal{A}$ must determine which message has been encrypted, given not just the ciphertext but auxiliary information a on the key and message m_1. For $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{SE},\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{IND}^{\mathcal{A}}_{\mathsf{SE},\mathsf{X}}(\lambda)] - 1$. We say that SE is X-KM-leakage resilient if the function $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{SE},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for all PT adversaries $\mathcal{A}$. This is of course not achievable if a allowed the adversary to compute k, so we restrict attention to unpredictable X. Furthermore, weakening the definition, we restrict attention to uniform X, meaning k and m_1 are uniformly and independently distributed. Thus we say that SE is KM-leakage-resilient if it is X-KM-leakage resilient for all unpredictable, uniform X.

The above requirement is strong in that security is required in the presence of (unpredictable) leakage on the key and first message. But beyond that, in other ways, it has been made weak, because this strengthens our negative results. Namely, we are only requiring security on random

messages, not chosen ones, with the key being uniformly distributed, and the key and the two messages all being independently distributed. Furthermore, in contrast to a typical indistinguishability definition, the adversary does not get the messages as input.

The standard correctness condition would ask that $\mathsf{SE.Dec}(1^\lambda, k, \mathsf{SE.Enc}(1^\lambda, k, m)) = m$ for all $k \in \{0,1\}^{\mathsf{SE.kl}(\lambda)}$, all $m \in \{0,1\}^{\mathsf{SE.ml}(\lambda)}$ and all $\lambda \in \mathbb{N}$. We call this perfect correctness. We formulate and use a weaker correctness condition because we can show un-achievability even under this and the weakening is crucial to our applications building KM-leakage-resilient encryption schemes to obtain further impossibility results. Specifically, we require correctness only for random messages and random keys with non-negligible probability. Formally, consider game $\mathrm{DEC}_{\mathsf{SE}}(\lambda)$ of Fig. 10 associated to SE, and for $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{dec}}_{\mathsf{SE}}(\lambda) = \Pr[\mathrm{DEC}_{\mathsf{SE}}(\lambda)]$ be the decryption correctness function of SE. We require that $\mathsf{Adv}^{\mathsf{dec}}_{\mathsf{SE}}(\cdot)$ be non-negligible.

We use standard definitions of UCE-secure function families, indistinguishability obfuscation and pseudorandom generators, as provided in Appendix A.

Proof of Theorem 5.1. The construction and proof extend ideas of [20]. Let a uniform auxiliary information generator X be specified as follows:

Algorithm $\mathsf{X.Ev}(1^\lambda)$	Circuit $\mathrm{C}_{1^\lambda,k,y}(c)$
$k \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.kl}(\lambda)}$	$m \leftarrow \mathsf{SE.Dec}(1^\lambda, k, c)$
$m \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.ml}(\lambda)}$; $y \leftarrow \mathsf{R.Ev}(1^\lambda, m)$	$y' \leftarrow \mathsf{R.Ev}(1^\lambda, m)$
$\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}_{1^\lambda,k,y})$; $\overline{\mathrm{C}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathrm{C})$	If $(y = y')$ then return 1
Return $(k, m, \overline{\mathrm{C}})$	Else return 0

The circuit $\mathrm{C}_{1^\lambda,k,y}$ takes as input a ciphertext c, decrypts it under the embedded key k to get back a $\mathsf{SE.ml}(\lambda)$-bit message m, applies the PRG to m to get a string y', and returns 1 iff y' equals the embedded string y. The auxiliary information generator creates this circuit as shown and outputs its obfuscation.

We define s as follows: For any $\lambda \in \mathbb{N}$ let $s(\lambda)$ be a polynomial upper bound on $\max(|\mathrm{C}^1_{1^\lambda,k,y}|, |\mathrm{C}^2|)$ where the circuits are defined in Fig. 11 and the maximum is over all $k \in \{0,1\}^{\mathsf{SE.kl}(\lambda)}$ and $y \in \{0,1\}^{2\cdot\mathsf{R.sl}(\lambda)}$.

Let us first present an attack proving (2). Below we define an adversary $\mathcal{A}$ against the X-KM-leakage resilience of SE and an adversary $\mathcal{R}$ against the PR-security of R:

Adversary $\mathcal{A}(1^\lambda, \overline{\mathrm{C}}, c)$	Adversary $\mathcal{R}(1^\lambda, y)$
$b' \leftarrow \overline{\mathrm{C}}(c)$	$k \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.kl}(\lambda)}$; $m_0 \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.ml}(\lambda)}$
Return b'	$c \leftarrow\!\!\$\ \mathsf{SE.Enc}(1^\lambda, k, m_0)$; $m \leftarrow \mathsf{SE.Dec}(1^\lambda, k, c)$
	$y' \leftarrow \mathsf{R.Ev}(1^\lambda, m)$
	If $(y' = y)$ then $g' \leftarrow 1$ else $g' \leftarrow 0$; Return g'

Adversary $\mathcal{A}$ has input 1^λ, the auxiliary information (leakage) which here is the obfuscated circuit $\overline{\mathrm{C}}$, and a ciphertext c. It simply computes and returns the bit $\overline{\mathrm{C}}(c) = \mathrm{C}_{1^\lambda,k,y}(c)$. For the analysis, consider game $\mathrm{IND}^{\mathcal{A}}_{\mathsf{SE},\mathsf{X}}(\lambda)$ of Fig. 10. If the challenge bit b is 1 and the decryption performed by $\overline{\mathrm{C}}$ is correct then $y' = y$, so

$$\Pr[\, b' = 1 \,|\, b = 1 \,] \geq \mathsf{Adv}^{\mathsf{dec}}_{\mathsf{SE}}(\lambda)\ . \tag{4}$$

In the case $b = 0$, the corresponding analysis in [20] for the insecurity of MB-AIPO relied on the fact that PRGs have low collision probability on random seeds. This will not suffice for us because of our weak correctness condition. The latter means that when $b = 0$, we do not know that $\mathsf{SE.Dec}(1^\lambda, k, c)$ equals m_0 and indeed have no guarantees on the distribution of decrypted plaintext

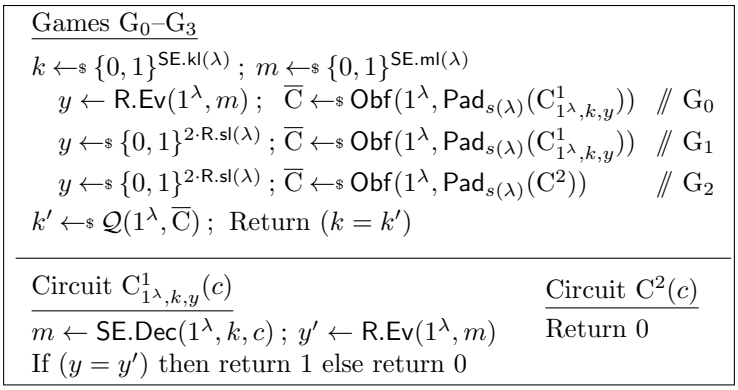


Figure 11: **Games for proof of claim (1) from Theorem 5.1.**

message. Instead, we directly exploit the assumed PR-security of the PRG. Thus, consider game $\mathrm{PRG}^{\mathcal{R}}_{\mathsf{R}}(\lambda)$ with adversary $\mathcal{R}$ as above. Letting g denote the challenge bit in the game, we have

$$\begin{aligned} \mathsf{Adv}^{\mathsf{pr}}_{\mathsf{R},\mathcal{R}}(\lambda) &= \Pr[\, g' = 1 \,|\, g = 1\,] - \Pr[\, g' = 1 \,|\, g = 0\,] \\ &\geq \Pr[\, b' = 1 \,|\, b = 0\,] - 2^{-\mathsf{R.sl}(\lambda)}\,. \end{aligned} \tag{5}$$

From Equation (5) and Equation (4), we have

$$\begin{aligned} \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{SE},\mathsf{X},\mathcal{A}}(\lambda) &= \Pr[\, b' = 1 \,|\, b = 1\,] - \Pr[\, b' = 1 \,|\, b = 0\,] \\ &\geq \mathsf{Adv}^{\mathsf{dec}}_{\mathsf{SE}}(\lambda) - \mathsf{Adv}^{\mathsf{pr}}_{\mathsf{R},\mathcal{R}}(\lambda) - 2^{-\mathsf{R.sl}(\lambda)}\,. \end{aligned} \tag{6}$$

Our weak correctness condition says the first term of Equation (6) is non-negligible. On the other hand, the second and third terms are negligible. This means $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{SE},\mathsf{X},\mathcal{A}}(\cdot)$ is not negligible, proving claim (2) of Theorem 5.1.

We proceed to prove claim (1). Let $\mathcal{Q}$ be a PT adversary. Consider the games and associated circuits of Fig. 11. Lines not annotated with comments are common to all three games. Game G_0 is equivalent to $\mathrm{PRED}^{\mathcal{Q}}_{\mathsf{X}}(\lambda)$, so

$$\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda) = \Pr[\mathrm{G}_2] + \sum_{i=0}^{1} \left(\Pr[\mathrm{G}_i] - \Pr[\mathrm{G}_{i+1}]\right). \tag{7}$$

We have $\Pr[\mathrm{G}_2] = 2^{-\mathsf{SE.kl}(\lambda)}$, where the latter is assumed to be negligible, because k is uniformly random and the circuit $\overline{\mathrm{C}}$ passed to adversary $\mathcal{Q}$ does not depend on k. We now show that $\Pr[\mathrm{G}_i] - \Pr[\mathrm{G}_{i+1}]$ is negligible for $i \in \{0,1\}$, which by Equation (7) implies that $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\cdot)$ is negligible and hence proves the claim.

First, we construct a PT adversary $\mathcal{R}$ against PRG R, as follows:

Adversary $\mathcal{R}(1^\lambda, y)$
$k \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}(\lambda)}$; $\overline{\mathrm{C}} \leftarrow\$ \mathsf{Obf}(1^\lambda, \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{1^\lambda,k,y}))$; $k' \leftarrow\$ \mathcal{Q}(1^\lambda, \overline{\mathrm{C}})$
If $(k = k')$ then return 1 else return 0

We have $\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1] = \mathsf{Adv}^{\mathsf{pr}}_{\mathsf{R},\mathcal{R}}(\lambda)$, where the advantage is negligible by the assumed PR-security of R.

Next, we construct a circuit sampler S^* and an iO-adversary $\mathcal{O}$, as follows:

Circuit Sampler $\mathsf{S}^*(1^\lambda)$	Adversary $\mathcal{O}(1^\lambda, \overline{\mathrm{C}}, aux)$
$k \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.kl}(\lambda)}$; $y \leftarrow\!\!\$\ \{0,1\}^{2\cdot \mathsf{R.sl}(\lambda)}$	$k' \leftarrow\!\!\$\ \mathcal{Q}(1^\lambda, \overline{\mathrm{C}})$
$\mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{1^\lambda,k,y})$; $\mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2)$	If $(k = k')$ then return 1
Return $(\mathrm{C}_0, \mathrm{C}_1, \varepsilon)$	Else return 0

It follows that $\Pr[\mathrm{G}_1] - \Pr[\mathrm{G}_2] = \mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S}^*,\mathcal{O}}(\lambda)$. We now show that $\mathsf{S}^* \in \boldsymbol{S}_{\mathrm{eq}}$, and hence $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S}^*,\mathcal{O}}(\lambda)$ is negligible by the assumed iO-security of Obf. Specifically, note that $\mathrm{C}^1_{1^\lambda,k,y}$ and C^2 are not equivalent only if y belongs to the range of R, which contains at most $2^{\mathsf{R.sl}(\lambda)}$ values. However, y is sampled uniformly at random from a set of size $2^{2\cdot \mathsf{R.sl}(\lambda)}$. It follows that

$$\Pr\left[\ \mathrm{C}_0 \equiv \mathrm{C}_1\ \ :\ \ (\mathrm{C}_0, \mathrm{C}_1, aux) \leftarrow\!\!\$\ \mathsf{S}^*(1^\lambda)\ \right] \geq 1 - 2^{-\mathsf{R.sl}(\lambda)},$$

where $2^{-\mathsf{R.sl}(\lambda)}$ is assumed to be negligible, and hence $\mathsf{S}^* \in \boldsymbol{S}_{\mathrm{eq}}$.

Corollaries. A consequence of Theorem 5.1 is the following.

Corollary E.1 *Let* SE *be a symmetric encryption scheme such that* $\mathsf{SE.ml}(\cdot) \in \Omega((\cdot)^\epsilon)$ *for some constant* $\epsilon > 0$*. Assume the existence of an indistinguishability obfuscator and a one-way function. Then* SE *is not KM-leakage resilient.*

Proof of Corollary E.1: The assumption on $\mathsf{SE.ml}$ implies that there exists a PR-secure PRG R with $\mathsf{R.sl} = \mathsf{SE.ml}$ [37]. To conclude we apply Theorem 5.1. ▮

F Proof of Theorem 5.2

Assuming for simplicity as in the construction that $\mathsf{H.il}$ is odd, let $\ell(\cdot) = (\mathsf{H.il}(\cdot) - 1)/2$. We now prove part (1). Let X be an unpredictable, uniform auxiliary information generator. Let $\mathcal{A}$ be a PT adversary. We build a PT source $\mathcal{S} \in \mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}$ and a PT distinguisher $\mathcal{D}$ such that

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{SE},\mathsf{X},\mathcal{A}}(\lambda)\ \leq\ 2 \cdot \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\lambda) \tag{8}$$

for all $\lambda \in \mathbb{N}$. The assumption that H is UCE$[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$-secure now implies part (1) of the theorem.

We proceed to build $\mathcal{S}, \mathcal{D}$. We let $\mathcal{S}$ be the split source $\mathcal{S} = \mathsf{Splt}[\mathcal{S}_0, \mathcal{S}_1]$, where algorithms $\mathcal{S}_0, \mathcal{S}_1$ are shown below, along with distinguisher $\mathcal{D}$:

Algorithm $\mathcal{S}_0(1^\lambda)$	Algorithm $\mathcal{S}_1(1^\lambda, \mathbf{y})$	Distinguisher $\mathcal{D}(1^\lambda, hk, L)$
$(k, m_1, a) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	Return $\mathbf{y}$	$((d, a), \mathbf{y}) \leftarrow L$; $c \leftarrow (hk, \mathbf{y})$
$m_0 \leftarrow\!\!\$\ \{0,1\}^{\ell(\lambda)}$; $d \leftarrow\!\!\$\ \{0,1\}$		$d' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, a, c)$
For $i = 1, \ldots, \ell(\lambda)$ do		If $(d = d')$ then $b' \leftarrow 1$
$\quad \mathbf{x}[i] \leftarrow k \| m_d[i] \| \langle i \rangle_{\ell(\lambda)}$		Else $b' \leftarrow 0$
Return $((d, a), \mathbf{x})$		Return b'

Here $\mathcal{S}_0$ calls the auxiliary information generator X to produce a key, a plaintext message and the corresponding auxiliary input. It then picks another plaintext message and the challenge bit d at random, and lets $\mathbf{x}$ consist of the inputs on which the hash function would be applied to create the challenge ciphertext. It leaks the challenge bit and auxiliary information. Algorithm $\mathcal{S}_1$ takes as input the result $\mathbf{y}$ of oracle HASH on $\mathbf{x}$, and leaks the entire vector $\mathbf{y}$. The distinguisher gets the leakage from both stages, together with the key hk. Using the latter, it can create the ciphertext c, which it passes to $\mathcal{A}$ to get back a decision. Its output reflects whether $\mathcal{A}$ wins its game.

Letting b denote the challenge bit in game $\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)$, we claim that

$$\Pr[\,b' = 1 \,|\, b = 1\,] = \frac{1}{2} + \frac{1}{2}\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{SE},\mathcal{A}}(\lambda) \qquad \text{and} \qquad \Pr[\,b' = 1 \,|\, b = 0\,] = \frac{1}{2}\ ,$$

from which Equation (8) follows. The first equation above should be clear from the construction. For the second, when $b = 0$, we know that HASH is a random oracle. But the entries of $\mathbf{x}$ are all distinct, due to the $\langle i\rangle_{\ell(\lambda)}$ components. So the entries of $\mathbf{y}$ are uniform and independent, and in particular independent of the challenge bit d.

This however does not end the proof: We still need to show that $\mathcal{S} \in \mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}$. We have ensured that $\mathcal{S} \in \mathbf{S}^{\mathrm{splt}}$ by construction. The crucial remainig step is to show that $\mathcal{S} \in \mathbf{S}^{\mathrm{cup}}$. This will exploit the assumed unpredictability of X. Let $\mathcal{P}$ be a PT predictor. We build PT adversary $\mathcal{Q}$ such that

$$\mathsf{Adv}^{\mathsf{pred}}_{\mathcal{S},\mathcal{P}}(\lambda) \ \le\ \mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda) \tag{9}$$

for all $\lambda \in \mathbb{N}$. The assumption that X is unpredictable now implies that $\mathcal{S} \in \mathbf{S}^{\mathrm{cup}}$. The construction of $\mathcal{Q}$ is as follows:

$$\begin{array}{l}\underline{\text{Adversary } \mathcal{Q}(1^\lambda, a)}\\ \text{For } i = 1,\ldots,\ell(\lambda) \text{ do } \mathbf{y}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}\\ d \leftarrow\!\!\$\ \{0,1\}\ ;\ x' \leftarrow\!\!\$\ \mathcal{P}(1^\lambda, ((d,a),\mathbf{y}))\ ;\ k \leftarrow x'[1..\ell(\lambda)]\ ;\ \text{Return } k\end{array}$$

Adversary $\mathcal{Q}$ computes leakage $((d,a),\mathbf{y})$ distributed exactly as it would be in game $\mathrm{PRED}^{\mathcal{P}}_{\mathcal{S}}(\lambda)$, where HASH is a random oracle. It then runs $\mathcal{P}$ to get a prediction x' of some oracle query of $\mathcal{S}$. If game $\mathrm{PRED}^{\mathcal{P}}_{\mathcal{S}}(\lambda)$ returns true, then x' must have the form $k\|m_d[i]\|\langle i\rangle_{\ell(\lambda)}$ for some $i \in \{1,\ldots,\ell(\lambda)\}$, where k, d are the key and challenge bit, respectively, chosen by $\mathcal{S}$. Adversary $\mathcal{Q}$ can then win its $\mathrm{PRED}^{\mathcal{Q}}_{\mathsf{X}}(\lambda)$ game by simply returning k, which establishes Equation (9).

This completes the proof of part (1) of the theorem. We prove part (2) by building a PT source $\mathcal{S} \in \mathbf{S}^{\mathrm{sup}} \cap \mathbf{S}^{\mathrm{splt}}$ and a PT distinguisher $\mathcal{D}$ such that

$$1 - \mathsf{Adv}^{\mathsf{dec}}_{\mathsf{SE}}(\lambda) \ \le\ \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\lambda) + \frac{\ell(\lambda)}{2^{\mathsf{H.ol}(\lambda)}} \tag{10}$$

for all $\lambda \in \mathbb{N}$. But we have assumed that H is $\mathrm{UCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}]$-secure, so it is also $\mathrm{UCE}[\mathbf{S}^{\mathrm{sup}} \cap \mathbf{S}^{\mathrm{splt}}]$-secure. We have also assumed $2^{-\mathsf{H.ol}(\cdot)}$ is negligible. Part (2) of the theorem follows.

We proceed to build $\mathcal{S}, \mathcal{D}$. We let $\mathcal{S}$ be the split source $\mathcal{S} = \mathsf{Splt}[\mathcal{S}_0, \mathcal{S}_1]$, where algorithms $\mathcal{S}_0, \mathcal{S}_1$ are shown below, along with distinguisher $\mathcal{D}$:

$$\begin{array}{l|l|l}
\underline{\text{Algorithm } \mathcal{S}_0(1^\lambda)} & \underline{\text{Algorithm } \mathcal{S}_1(1^\lambda, \mathbf{y})} & \underline{\text{Distinguisher } \mathcal{D}(1^\lambda, hk, (\varepsilon, \mathbf{y}))}\\
k \leftarrow\!\!\$\ \{0,1\}^{\ell(\lambda)} & \text{Return } \mathbf{y} & b' \leftarrow 0\\
\text{For } i = 1,\ldots,\ell(\lambda) \text{ do} & & \text{For } i = 1,\ldots,\ell(\lambda) \text{ do}\\
\quad \mathbf{x}[2i-1] \leftarrow k\|1\|\langle i\rangle_{\ell(\lambda)} & & \quad \text{If } (\mathbf{y}[2i-1] = \mathbf{y}[2i])\\
\quad \mathbf{x}[2i] \leftarrow k\|0\|\langle i\rangle_{\ell(\lambda)} & & \quad \text{Then } b' \leftarrow 1\\
\text{Return } (\varepsilon, \mathbf{x}) & & \text{Return } b'
\end{array}$$

Letting b denote the challenge bit in game $\mathrm{UCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)$, we claim that

$$\Pr[\,b' = 1 \,|\, b = 1\,] \ge 1 - \mathsf{Adv}^{\mathsf{dec}}_{\mathsf{SE}}(\lambda) \qquad \text{and} \qquad \Pr[\,b' = 1 \,|\, b = 0\,] \le \frac{\ell(\lambda)}{2^{\mathsf{H.ol}(\lambda)}}\ ,$$

from which Equation (10) follows. The first equation above is true because decryption errors only happen when hash outputs collide for different values of the message bit. For the second, when $b = 0$, we know that HASH is a random oracle. But the entries of $\mathbf{x}$ are all distinct. So the entries

of $\mathbf{y}$ are uniform and independent. The chance of a collision of two entries is thus $2^{-\mathsf{H.ol}(\lambda)}$, and the equation then follows from the union bound.

$\mathcal{S}$ is a split source by construction. To conclude the proof we need to show that $\mathcal{S} \in \mathbf{S}^{\mathrm{sup}}$. In the case HASH is a random oracle, the distinctness of the oracle queries of $\mathcal{S}$ means that the entries of $\mathbf{y}$ are uniformly and independently distributed. Since there is no leakage beyond $\mathbf{y}$, the leakage gives the predictor $\mathcal{P}$ no extra information about the entries of $\mathbf{x}$. The uniform choice of k by $\mathcal{S}$ then means that $\mathsf{Adv}^{\mathsf{pred}}_{\mathcal{S},\mathcal{P}}(\cdot) \leq 2^{-\ell(\cdot)}$, even if $\mathcal{P}$ is not restricted to PT. But our assumption on $\mathsf{H.il}(\cdot)$ in the theorem statement implies that $2^{-\ell(\cdot)}$ is negligible.