

# Contention in Cryptoland: Obfuscation, Leakage and UCE

Mihir Bellare<sup>1</sup>(✉), Igors Stepanovs<sup>1</sup>, and Stefano Tessaro<sup>2</sup>

<sup>1</sup> Department of Computer Science and Engineering,  
University of California San Diego, San Diego, USA  
mihir@eng.ucsd.edu,

<http://cseweb.ucsd.edu/~mihir/>,  
<https://sites.google.com/site/igorstepanovs/>

<sup>2</sup> Department of Computer Science,  
University of California Santa Barbara, Santa Barbara, USA  
<http://www.cs.ucsb.edu/~tessaro/>

**Abstract.** This paper addresses the fundamental question of whether or not different, exciting primitives now being considered actually exist. We show that we, unfortunately, cannot have them all. We provide results of the form  $\neg\mathbf{A} \vee \neg\mathbf{B}$ , meaning one of the primitives  $\mathbf{A}$ ,  $\mathbf{B}$  cannot exist. (But we don't know which.) Specifically, we show that: (1) **VGBO** (Virtual Grey Box Obfuscation) for all circuits, which has been conjectured to be achieved by candidate constructions, cannot co-exist with Canaletto's 1997 **AI-DHI** (auxiliary input DH inversion) assumption, which has been used to achieve many goals including point-function obfuscation (2) **iO** (indistinguishability obfuscation) for all circuits cannot co-exist with **KM-LR-SE** (key-message leakage-resilient symmetric encryption) (3) **iO** cannot co-exist with hash functions that are **UCE** secure for computationally unpredictable split sources.

## 1 Introduction

Cryptographic theory is being increasingly 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 [31] or is justified under other, new and strong assumptions [12, 34, 41]. 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 [27, 28, 33, 39]. 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 that we call *contentions*. These take the form  $\neg\mathbf{A} \vee \neg\mathbf{B}$  where  $\mathbf{A}$ ,  $\mathbf{B}$  are different primitives or assumptions. This shows that  $\mathbf{A}$ ,  $\mathbf{B}$  are not *both* achievable, meaning they cannot co-exist. We may not know which of the two fails, but at least

one of the two must, which is valuable and sometimes surprising information. Indeed, many intriguing contentions of this form have been provided in recent work [5, 11, 14, 20–22, 32, 36]. For example, we know that the following cannot co-exist: “Special-purpose obfuscation” and diO [32]; Multi-bit point function obfuscation and iO [22]; extractable one-way functions and iO [14].

In this paper we begin by addressing the question of whether VGBO (Virtual Grey Box Obfuscation) for all circuits is possible, as conjectured by BCKP [13, Section 1.1]. We show that this is in contention with the AI-DHI assumption of [15, 25]. We go on to show that iO is in contention with certain forms of leakage resilient encryption and UCE.

## 1.1 VGBO and AI-DHI

We show that  $\neg\text{VGBO} \vee \neg\text{AI-DHI}$ . That is, Virtual Grey Box Obfuscation (VGBO) of all circuits is in contention with Canaletto’s 1997 AI-DHI (Auxiliary-Input Diffie-Hellman Inversion) assumption [15, 25]. One of the two (or both) must 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) [4, 37], 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 [4, 16, 35] has led to efforts to define and achieve weaker forms of obfuscation. VGBO [10] 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 [12, 13] show that VGBO for  $\text{NC}^1$  is achievable (under a novel assumption). They then say “existing candidate indistinguishability obfuscators for all circuits [3, 19, 31] may also be considered as candidates for VGB obfuscation, for all circuits” [13, 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.

The AI-DHI assumption [15, 25] 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 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 [25], AIPO (auxiliary-input point-function obfuscation) [15] and zero-knowledge proofs [15].

Our result is that  $\neg\text{VGBO} \vee \neg\text{AI-DHI}$ . That is, either VGBO for all circuits is impossible or the AI-DHI 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  $\overline{C}$  of  $C_x$ . Now (2) is easy to see: the adversary, given challenge  $(u, v)$ , can win by returning  $\overline{C}(u, v)$ . But why is (1) true? We 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. This is shown through an information-theoretic argument that exploits the group structure.

The natural question about which one would be curious is, which of VGBO and AI-DHI is it that fails? This is an intriguing question and we do not know the answer at this point.

## 1.2 Key-Message Leakage Resilience

DKL [30] and CKVW [26] provide key leakage resilient symmetric encryption (K-LR-SE) schemes. This means they retain security even when the adversary has auxiliary information about the key, as long as this information is key-prediction-precluding, meaning does not allow one to compute the key. We consider a generalization that we call key-message leakage resilient symmetric encryption (KM-LR-SE). Here 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, meaning one cannot compute the key from the auxiliary information. 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  $\text{iO}$  for all circuits is possible. That is, we show in Theorem 3 that  $\neg \text{iO} \vee \neg \text{KM-LR-SE}$ . Since  $\text{iO}$  seems to be growing to be more broadly accepted, this indicates that KM-LR-SE is not likely to exist. We think this may be of direct interest from the perspective of leakage resilience, but its main importance for us is as a tool to establish new negative results for UCE as discussed in Sect. 1.3 below. The proof of Theorem 3 is a minor adaptation of the proof of BM [22] ruling out MB-AIPO under  $\text{iO}$ .

## 1.3 UCE for Split Sources

UCE is a class of assumptions for function families introduced in BHK [6] with the goal of instantiating random oracles. For a class  $\mathbf{S}$  of algorithms called sources, BHK define  $\text{UCE}[\mathbf{S}]$  security of a family of functions. The parameterization is necessary because security is not achievable for the class of all sources. BHK and subsequent work [5, 6, 20, 23, 29, 40] have considered several restricted classes of sources and, based on the assumption of UCE security for these, been able to instantiate random oracles to obtain secure, efficient instantiations for primitives including deterministic public-key encryption, message-locked encryption, encryption secure for key-dependent messages, encryption secure under related-key attacks, adaptive garbling, hardcore functions and IND-CCA public-key encryption.

However UCE here has functioned as an assumption. We know little about its achievability. The basic foundational question in this area is, for which source

classes  $\mathbf{S}$  is  $\text{UCE}[\mathbf{S}]$  security achievable? The first step towards answering this was taken by BFM [20], who showed that  $\neg \text{iO} \vee \neg \text{UCE}[\mathbf{S}^{\text{cup}}]$ . That is,  $\text{iO}$  for all circuits is in contention with UCE security relative to the class  $\mathbf{S}^{\text{cup}}$  of all computationally unpredictable sources. (These are sources whose leakage is computationally unpredictable when their queries are answered by a random oracle.) This lead BHK [6] to propose restricting attention to “split” sources. Such sources can leak information about an oracle query and its answer separately, but not together. This circumvents the BFM attack. Indeed,  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$  appeared plausible even in the presence of  $\text{iO}$ . However in this paper we show  $\neg \text{iO} \vee \neg \text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$ , meaning  $\text{iO}$  and  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$  security cannot co-exist. The interpretation is that  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$ -secure function families are unlikely to exist. We obtain our  $\neg \text{iO} \vee \neg \text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$  result by showing that  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}] \Rightarrow \text{KM-LR-SE}$ , meaning we can build a key-message leakage resilient symmetric encryption scheme given any  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$ -secure function family. But we saw above that  $\neg \text{iO} \vee \neg \text{KM-LR-SE}$  and can thus conclude that  $\neg \text{iO} \vee \neg \text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$ .

BM2 [23] show that  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}} \cap \mathbf{S}^1]$  security —  $\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}} \cap \mathbf{S}^q$  is the class of computationally unpredictable split sources making  $q$  oracle queries— is achievable. (They assume  $\text{iO}$  and  $\text{AIPO}$ .) Our  $\neg \text{iO} \vee \neg \text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$  result does not contradict this since our source makes a polynomial number of oracle queries. Indeed our result complements the BM2 one to show that a bound on the number of source oracle queries is necessary for a positive result. Together these results give a close to complete picture of the achievability of UCE for split sources, the remaining open question being achievability of  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}} \cap \mathbf{S}^q]$  for constant  $q > 1$ .

We note that we are not aware of any applications assuming  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$ . Prior applications have used either  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}} \cap \mathbf{S}^1]$  or quite different classes like  $\text{UCE}[\mathbf{S}^{\text{sup}}]$  —  $\mathbf{S}^{\text{sup}}$  is the class of statistically unpredictable sources [6, 20]— and neither of these is at risk from our results. However our  $\neg \text{iO} \vee \neg \text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$  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. The achievability of  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$  security was an open problem from prior work.

### 1.4 Discussion and Related Work

The idea of using an obfuscated circuit as an auxiliary input to obtain contention results has appeared in many prior works [5, 11, 14, 20–22, 32, 36]. Some of the contentions so established are between “Special-purpose obfuscation” and  $\text{diO}$  [32], between  $\text{MB-AIPO}$  and  $\text{iO}$  [22] and between extractable one-way functions and  $\text{iO}$  [14]. Our work follows in these footsteps.

$\text{KM-LR-SE}$  can be viewed as a symmetric encryption re-formulation of  $\text{MB-AIPO}$  following the connection of the latter to symmetric encryption established by  $\text{CKVW}$  [26]. The main change is in the correctness condition. We formulate a weak correctness condition, which is important for our application to UCE. In its absence, our negative result for split-source UCE would only be

for injective functions, which is much weaker. With this connection in mind, the proof of Theorem 3, as we have indicated above, is a minor adaptation of the proof of BM [22] ruling out MB-AIPO under iO. Our result about KM-LR-SE is thus not of technical novelty or interest but we think this symmetric encryption re-formulation of BM [22] is of interest from the leakage resilience perspective and as a tool to obtain more negative results, as exemplified by our application to UCE.

In independent and concurrent work, BM3 [24] show  $\neg\text{iO} \vee \neg\text{UCE}[\mathbf{S}^{\text{s-cup}}]$ , where  $\mathbf{S}^{\text{s-cup}}$  is the class of *strongly* computationally unpredictable sources as defined in [22]. But the latter show that  $\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}$  is a *strict* subset of  $\mathbf{S}^{\text{s-cup}}$ . This means that our  $\neg\text{iO} \vee \neg\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$  result is strictly stronger than the  $\neg\text{iO} \vee \neg\text{UCE}[\mathbf{S}^{\text{s-cup}}]$  result of [24]. (Under iO, our result rules out UCE security for a smaller, more restricted class of sources.)

Our results on UCE, as with the prior ones of BFM [20], are for the basic setting, where there is a single key or single user [6]. BHK [6] also introduce a multi-key (multi-user) setting. Some negative results about this are provided in [8].

## 2 Preliminaries

**Notation.** We denote by  $\lambda \in \mathbb{N}$  the security parameter and by  $1^\lambda$  its unary representation. If  $x \in \{0, 1\}^*$  is a string then  $|x|$  denotes its length,  $x[i]$  denotes its  $i$ -th bit, and  $x[i..j] = x[i] \dots x[j]$  for  $1 \leq i \leq j \leq |x|$ . We let  $\varepsilon$  denote the empty string. If  $s$  is an integer then  $\text{Pad}_s(C)$  denotes circuit  $C$  padded to have size  $s$ . We say that circuits  $C_0, C_1$  are equivalent, written  $C_0 \equiv C_1$ , 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  $x \leftarrow_s X$  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  $y \leftarrow A(x_1, \dots; r)$  denote running  $A$  with random coins  $r$  on inputs  $x_1, \dots$  and assigning the output to  $y$ . We let  $y \leftarrow_s A(x_1, \dots)$  be the result of picking  $r$  at random and letting  $y \leftarrow A(x_1, \dots; r)$ . We let  $[A(x_1, \dots)]$  denote the set of all possible outputs of  $A$  when invoked with inputs  $x_1, \dots$ . We say that  $f: \mathbb{N} \rightarrow \mathbb{R}$  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 [7]. (See Fig. 1 for an example.) By  $G^A(\lambda)$  we denote the event that the execution of game  $G$  with adversary  $\mathcal{A}$  and security parameter  $\lambda$  results in the game returning true.

**Auxiliary Information Generators.** Many of the notions we consider involve the computational unpredictability of some quantity even given “auxiliary information” about it. We abstract this out via our definition of an *auxiliary information generator*  $X$ . The latter specifies a PT algorithm  $X.\text{Ev}$  that takes  $1^\lambda$  to return a *target*  $k \in \{0, 1\}^{X.\text{tl}(\lambda)}$ , a *payload*  $m \in \{0, 1\}^{X.\text{pl}(\lambda)}$  and an *auxiliary information*  $a$ , where  $X.\text{tl}, X.\text{pl}: \mathbb{N} \rightarrow \mathbb{N}$  are the target and payload length functions associated to  $X$ , respectively. Consider game PRED of Fig. 1 associated to  $X$  and a predictor

adversary  $\mathcal{Q}$ . For  $\lambda \in \mathbb{N}$  let  $\text{Adv}_{\mathbb{X}, \mathcal{Q}}^{\text{pred}}(\lambda) = \Pr[\text{PRED}_{\mathbb{X}}^{\mathcal{Q}}(\lambda)]$ . We say that  $\mathbb{X}$  is *unpredictable* if  $\text{Adv}_{\mathbb{X}, \mathcal{Q}}^{\text{pred}}(\cdot)$  is negligible for every PT adversary  $\mathcal{Q}$ . We say that  $\mathbb{X}$  is *uniform* if  $\mathbb{X}.\text{Ev}(1^\lambda)$  picks the target  $k \in \{0, 1\}^{\mathbb{X}.\text{tl}(\lambda)}$  and the payload  $m \in \{0, 1\}^{\mathbb{X}.\text{pl}(\lambda)}$  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.

Game $\text{PRED}_{\mathbb{X}}^{\mathcal{Q}}(\lambda)$	Game $\text{PRG}_{\mathbb{R}}^{\mathcal{R}}(\lambda)$	Game $\text{IO}_{\text{Obf}, \mathbb{S}}^{\mathcal{O}}(\lambda)$
$(k, m, a) \leftarrow_s \mathbb{X}.\text{Ev}(1^\lambda)$	$b \leftarrow_s \{0, 1\}$	$b \leftarrow_s \{0, 1\}$
$k' \leftarrow_s \mathcal{Q}(1^\lambda, a)$	$m \leftarrow_s \{0, 1\}^{\mathbb{R}.\text{sl}(\lambda)}$	$(C_0, C_1, aux) \leftarrow_s \mathbb{S}(1^\lambda)$
Return $(k = k')$	$y_1 \leftarrow \mathbb{R}.\text{Ev}(1^\lambda, m)$	$\overline{C} \leftarrow_s \text{Obf}(1^\lambda, C_b)$
	$y_0 \leftarrow_s \{0, 1\}^{2 \cdot \mathbb{R}.\text{sl}(\lambda)}$	$b' \leftarrow_s \mathcal{O}(1^\lambda, \overline{C}, aux)$
	$b' \leftarrow_s \mathcal{R}(1^\lambda, y_b)$	Return $(b = b')$
	Return $(b = b')$	

**Fig. 1.** Games defining unpredictability of auxiliary information generator  $\mathbb{X}$ , PR-security of pseudorandom generator  $\mathbb{R}$  and iO-security of obfuscator  $\text{Obf}$  relative to circuit sampler  $\mathbb{S}$ .

**PRGs.** A pseudorandom generator  $\mathbb{R}$  [17, 44] specifies a deterministic PT algorithm  $\mathbb{R}.\text{Ev}$  where  $\mathbb{R}.\text{sl}: \mathbb{N} \rightarrow \mathbb{N}$  is the seed length function of  $\mathbb{R}$  such that  $\mathbb{R}.\text{Ev}(1^\lambda, \cdot): \{0, 1\}^{\mathbb{R}.\text{sl}(\lambda)} \rightarrow \{0, 1\}^{2 \cdot \mathbb{R}.\text{sl}(\lambda)}$  for all  $\lambda \in \mathbb{N}$ . We say that  $\mathbb{R}$  is PR-secure if the function  $\text{Adv}_{\mathbb{R}, \mathcal{R}}^{\text{pr}}(\cdot)$  is negligible for every PT adversary  $\mathcal{R}$ , where for  $\lambda \in \mathbb{N}$  we let  $\text{Adv}_{\mathbb{R}, \mathcal{R}}^{\text{pr}}(\lambda) = 2 \Pr[\text{PRG}_{\mathbb{R}}^{\mathcal{R}}(\lambda)] - 1$  and game PRG is specified in Fig. 1.

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

**Indistinguishability Obfuscation.** We use the BST [9] definitional framework which parameterizes security via classes of circuit samplers. Let  $\text{Obf}$  be an obfuscator. A *sampler* in this context is a PT algorithm  $\mathbb{S}$  that on input  $1^\lambda$  returns a triple  $(C_0, C_1, aux)$  where  $C_0, C_1$  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  $\text{Adv}_{\text{Obf}, \mathbb{S}, \mathcal{O}}^{\text{iO}}(\lambda) = 2 \Pr[\text{IO}_{\text{Obf}, \mathbb{S}}^{\mathcal{O}}(\lambda)] - 1$  where game  $\text{IO}_{\text{Obf}, \mathbb{S}}^{\mathcal{O}}(\lambda)$  is defined in Fig. 1. Now let  $\mathbf{S}$  be a class (set) of circuit samplers. We say that  $\text{Obf}$  is  $\mathbf{S}$ -secure if  $\text{Adv}_{\text{Obf}, \mathbb{S}, \mathcal{O}}^{\text{iO}}(\cdot)$  is negligible for every PT adversary  $\mathcal{O}$  and every circuit sampler  $\mathbb{S} \in \mathbf{S}$ . We say that circuit sampler  $\mathbb{S}$  produces equivalent circuits if there exists a negligible function  $\nu$  such that  $\Pr [ C_0 \equiv C_1 : (C_0, C_1, aux) \leftarrow_s \mathbb{S}(1^\lambda) ] \geq 1 - \nu(\lambda)$  for all  $\lambda \in \mathbb{N}$ . Let  $\mathbf{S}_{\text{eq}}$  be the class of all circuit samplers that produce equivalent circuits. We say that  $\text{Obf}$  is an indistinguishability obfuscator if it is  $\mathbf{S}_{\text{eq}}$ -secure [4, 31, 42].

**Function Families.** A family of functions  $F$  specifies the following. PT key generation algorithm  $F.Kg$  takes  $1^\lambda$  to return a key  $fk \in \{0, 1\}^{F.kl(\lambda)}$ , where  $F.kl: \mathbb{N} \rightarrow \mathbb{N}$  is the key length function associated to  $F$ . Deterministic, PT evaluation algorithm  $F.Ev$  takes  $1^\lambda$ , key  $fk \in [F.Kg(1^\lambda)]$  and an input  $x \in \{0, 1\}^{F.il(\lambda)}$  to return an output  $F.Ev(1^\lambda, fk, x) \in \{0, 1\}^{F.ol(\lambda)}$ , where  $F.il, F.ol: \mathbb{N} \rightarrow \mathbb{N}$  are the input and output length functions associated to  $F$ , respectively. We say that  $F$  is *injective* if the function  $F.Ev(1^\lambda, fk, \cdot): \{0, 1\}^{F.il(\lambda)} \rightarrow \{0, 1\}^{F.ol(\lambda)}$  is injective for every  $\lambda \in \mathbb{N}$  and every  $fk \in [F.Kg(1^\lambda)]$ .

**UCE Security.** Let us recall the Universal Computational Extractor (UCE) framework of BHK [6]. 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  $UCE_H^{\mathcal{S}, \mathcal{D}}(\lambda)$  in the left panel of Fig. 2. The source has access to an oracle  $HASH$  and we require that any query  $x$  made to this oracle have length  $H.il(\lambda)$ . When the challenge bit  $b$  is 1 (the “real” case) the oracle responds via  $H.Ev$  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  $Adv_{H, \mathcal{S}, \mathcal{D}}^{uce}(\lambda) = 2 \Pr[UCE_H^{\mathcal{S}, \mathcal{D}}(\lambda)] - 1$ . If  $\mathbf{S}$  is a class (set) of sources, we say that  $H$  is  $UCE[\mathbf{S}]$ -secure if  $Adv_{H, \mathcal{S}, \mathcal{D}}^{uce}(\cdot)$  is negligible for all sources  $\mathcal{S} \in \mathbf{S}$  and all PT distinguishers  $\mathcal{D}$ .

Game $UCE_H^{\mathcal{S}, \mathcal{D}}(\lambda)$	Game $PRED_S^{\mathcal{P}}(\lambda)$	Source $\mathcal{S}^{HASH}(1^\lambda)$
$b \leftarrow_s \{0, 1\}; hk \leftarrow_s H.Kg(1^\lambda)$	$X \leftarrow \emptyset$	$(L_0, \mathbf{x}) \leftarrow_s \mathcal{S}_0(1^\lambda)$
$L \leftarrow_s \mathcal{S}^{HASH}(1^\lambda)$	$L \leftarrow_s \mathcal{S}^{HASH}(1^\lambda)$	For $i = 1, \dots,  \mathbf{x} $ do
$b' \leftarrow_s \mathcal{D}(1^\lambda, hk, L)$	$x' \leftarrow_s \mathcal{P}(1^\lambda, L)$	$\mathbf{y}[i] \leftarrow_s HASH(\mathbf{x}[i])$
Return $(b' = b)$	Return $(x' \in X)$	$L_1 \leftarrow_s \mathcal{S}_1(1^\lambda, \mathbf{y})$
$HASH(x)$	$HASH(x)$	$L \leftarrow (L_0, L_1)$
If $T[x] = \perp$ then	If $T[x] = \perp$ then	Return $L$
If $b = 0$ then $T[x] \leftarrow_s \{0, 1\}^{H.ol(\lambda)}$	$T[x] \leftarrow_s \{0, 1\}^{H.ol(\lambda)}$	
Else $T[x] \leftarrow H.Ev(1^\lambda, hk, x)$	$X \leftarrow X \cup \{x\}$	
Return $T[x]$	Return $T[x]$	

**Fig. 2.** Games defining UCE security of function family  $H$ , unpredictability of source  $\mathcal{S}$ , and the split source  $\mathcal{S} = \text{Spl}[S_0, S_1]$  associated to  $S_0, S_1$ .

It is easy to see that  $UCE[\mathbf{S}]$ -security is not achievable if  $\mathbf{S}$  is the class of all PT sources [6]. To obtain meaningful notions of security, BHK [6] 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  $PRED_S^{\mathcal{P}}(\lambda)$  in the middle panel of Fig. 2. For  $\lambda \in \mathbb{N}$  we



let  $\text{Adv}_{\mathcal{S}, \mathcal{P}}^{\text{pred}}(\lambda) = \Pr[\text{PRED}_{\mathcal{S}}^{\mathcal{P}}(\lambda)]$ . We say that  $\mathcal{S}$  is computationally unpredictable if  $\text{Adv}_{\mathcal{S}, \mathcal{P}}^{\text{pred}}(\cdot)$  is negligible for all PT predictors  $\mathcal{P}$ , and let  $\mathbf{S}^{\text{cup}}$  be the class of all PT computationally unpredictable sources. We say that  $\mathcal{S}$  is statistically unpredictable if  $\text{Adv}_{\mathcal{S}, \mathcal{P}}^{\text{pred}}(\cdot)$  is negligible for all (not necessarily PT) predictors  $\mathcal{P}$ , and let  $\mathbf{S}^{\text{sup}} \subseteq \mathbf{S}^{\text{cup}}$  be the class of all PT statistically unpredictable sources.

BFM [20] show that  $\text{UCE}[\mathbf{S}^{\text{cup}}]$ -security is not achievable assuming that indistinguishability obfuscation is possible. This has lead applications to either be based on  $\text{UCE}[\mathbf{S}^{\text{sup}}]$  or on subsets of  $\text{UCE}[\mathbf{S}^{\text{cup}}]$ , meaning to impose further restrictions on the source.  $\text{UCE}[\mathbf{S}^{\text{sup}}]$ , introduced in [6, 20], 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 [6]. Let  $\mathcal{S}_0, \mathcal{S}_1$  be algorithms, neither of which have access to any oracles. The *split source*  $\mathcal{S} = \text{Spl}[\mathcal{S}_0, \mathcal{S}_1]$  associated to  $\mathcal{S}_0, \mathcal{S}_1$  is defined in the right panel of Fig. 2. Algorithm  $\mathcal{S}_0$  returns a pair  $(L_0, \mathbf{x})$ . Here  $\mathbf{x}$  is a vector over  $\{0, 1\}^{\text{H.il}(\lambda)}$  all of whose entries are required to be distinct. (If the entries are not required to be distinct, collisions can be used to communicate information between the two components of the source, and the BFM [20] attack continues to apply, as pointed out in [23].) 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  $\mathbf{S}^{\text{splt}}$  if there exist PT  $\mathcal{S}_0, \mathcal{S}_1$  such that  $\mathcal{S} = \text{Spl}[\mathcal{S}_0, \mathcal{S}_1]$ , meaning is defined as above. The class of interest is now  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$ , meaning  $\text{UCE}$ -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  $\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}} \cap \mathbf{S}^1$ , a class of computationally unpredictable split sources that make a single query. BM2 [23] show that  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}} \cap \mathbf{S}^1]$ -security is achievable assuming iO and AIPO.

### 3 VGBO and the AI-DHI Assumption

BCKP [12, 13] conjecture that existing 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 Canaletto’s AI-DHI 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.) That is, the long-standing AI-DHI assumption and VGBO are in contention; at most one of these can exist. We start by defining VGBO and recalling the AI-DHI assumption, and then give our result and its proof. We then suggest a weakening of AI-DHI that we call AI-DHI2 that is parameterized by a group generator. We show that our attack on AI-DHI extends to rule out AI-DHI2 for group generators satisfying a property we call verifiability. However there may be group generators that do not appear to be verifiable, making AI-DHI2 a potential alternative to AI-DHI.



Game $\text{VGB1}_{\text{Obf}, \text{Smp}}^A(\lambda)$	Game $\text{VGB0}_{\text{Smp}, q}^S(\lambda)$
$C \leftarrow_{\text{s}} \text{Smp}(1^\lambda)$	$C \leftarrow_{\text{s}} \text{Smp}(1^\lambda); i \leftarrow 0$
$\bar{C} \leftarrow_{\text{s}} \text{Obf}(1^\lambda, C)$	$b' \leftarrow_{\text{s}} \mathcal{S}^{\text{CIRC}}(1^\lambda)$
$b' \leftarrow_{\text{s}} \mathcal{A}(1^\lambda, \bar{C})$	Return ( $b' = 1$ )
Return ( $b' = 1$ )	$\text{CIRC}(x)$
	$i \leftarrow i + 1$
	If $i > q(\lambda)$ then return $\perp$
	$y \leftarrow C(x);$ Return $y$

**Fig. 3.** Games defining VGB security of obfuscator Obf.

**VGBO.** Let Obf be an obfuscator as defined in Sect. 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 [10, 12], which strengthens our results since they are negative relations with starting point VGBO.

A *sampler* Smp in this context is an algorithm that takes  $1^\lambda$  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

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

where the games are in Fig. 3. 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  $\text{Adv}_{\text{Obf}, \text{Smp}, q, \mathcal{A}, \mathcal{S}}^{\text{vgb}}(\cdot)$  is negligible for all  $\text{Smp} \in \text{SAMP}$ .

We note that [12] 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  $\pi(C)$ , rather than just the probabilities of outputting one, which is all we need here. Also note that our VGB definition is vacuously achievable whenever  $|\text{SAMP}| = 1$ , since  $\mathcal{S}$  can simulate game  $\text{VGB1}_{\text{Obf}, \text{Smp}}^A(\lambda)$  for any fixed choice of  $\mathcal{A}$  and Smp. Our applications however use a SAMP of size 2.

**The AI-DHI 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  $\lambda$ , 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  $\mathbf{X}$  for  $\mathcal{G}$  is an auxiliary information generator as per Sect. 2 with the additional property that the target  $k$  returned by  $\mathbf{X}.\text{Ev}(1^\lambda)$  is in  $\mathbb{Z}_{p(\lambda)}$  (i.e. is an exponent) and the payload  $m$  is  $\varepsilon$  (i.e. is effectively absent).

Now consider game AIDHI of Fig. 4 associated to  $\mathcal{G}, \mathbf{X}$  and an adversary  $\mathcal{A}$ . For  $\lambda \in \mathbb{N}$  let  $\text{Adv}_{\mathcal{G}, \mathbf{X}, \mathcal{A}}^{\text{aidhi}}(\lambda) = 2 \Pr[\text{AIDHI}_{\mathcal{G}, \mathbf{X}}^A(\lambda)] - 1$ . We say that  $\mathcal{G}$  is AI-DHI-secure if  $\text{Adv}_{\mathcal{G}, \mathbf{X}, \mathcal{A}}^{\text{aidhi}}(\cdot)$  is negligible for every unpredictable  $\mathbf{X}$  for  $\mathcal{G}$  and every PT adversary

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

**Fig. 4.** Games defining the AI-DHI assumption and the AI-DHI2 assumption.

$\mathcal{A}$ . The AI-DHI assumption [15, 25] is that there exists a family of groups  $\mathcal{G}$  that is AI-DHI secure.

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

**Theorem 1.** *Let  $\mathcal{G}$  be a family of groups. Then there is a pair  $\text{Smp}, \text{Smp}_0$  of PT samplers (defined in the proof) such that if there exists a VGB-secure obfuscator for the class  $\text{SAMP} = \{\text{Smp}, \text{Smp}_0\}$ , then  $\mathcal{G}$  is not AI-DHI-secure.*

*Proof (Theorem 1).* Let  $\text{Obf}$  be the assumed obfuscator. Let  $\mathbf{X}$  be the auxiliary information generator for  $\mathcal{G}$  defined as follows:

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

The auxiliary information  $a = \overline{\text{C}}$  produced by  $\mathbf{X}$  is an obfuscation of the circuit  $\text{C}_{1^\lambda, k}$  shown on the right above. The circuit has  $1^\lambda$  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  $\text{Adv}_{\mathcal{G}, \mathbf{X}, \mathcal{A}^*}^{\text{aidhi}}(\cdot)$  is non-negligible. On input  $1^\lambda, g, K_b, \overline{\text{C}}$ , it simply returns  $\overline{\text{C}}(g, K_b)$ . That is, it runs the obfuscated circuit  $\overline{\text{C}}$  on  $g$  and  $K_b$  and returns its outcome. If the challenge bit  $b$  in game  $\text{AIDHI}_{\mathcal{G}, \mathbf{X}}^{\mathcal{A}^*}(\lambda)$  is 1 then the adversary always outputs  $b' = 1$ . Otherwise, the adversary outputs  $b' = 1$  with probability  $1/p(\lambda)$ . We have  $\text{Adv}_{\mathcal{G}, \mathbf{X}, \mathcal{A}^*}^{\text{aidhi}}(\lambda) = 1 - 1/p(\lambda) \geq 1 - 2^{1-\lambda}$ , which is not negligible.

We now show that the constructed auxiliary information generator  $\mathbf{X}$  is unpredictable. In particular, for any PT adversary  $\mathcal{Q}$  we construct a PT adversary  $\mathcal{A}$  and samplers  $\text{Smp}, \text{Smp}_0$  such that for all simulators  $\mathcal{S}$  and all polynomials  $q$ ,

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

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

$\overline{k}' \leftarrow_s \mathcal{Q}(1^\lambda, \overline{C})$ $\overline{g} \leftarrow_s \mathbb{G}_\lambda^*$ Return $\overline{C}(\overline{g}, \overline{g}^{k'})$	Algorithm $\text{Smp}(1^\lambda)$ $k \leftarrow_s \mathbb{Z}_{p(\lambda)}$ Return $C_{1^\lambda, k}$	Algorithm $\text{Smp}_0(1^\lambda)$ Return $C_0$
---	--	---

In  $\text{Smp}_0$ , the circuit  $C_0$  takes as input a pair of group elements  $g, g'$  from  $\mathbb{G}_\lambda$  and always returns 0.

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

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

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

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

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

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

and

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

Moreover, we have  $\Pr [\text{VGB1}_{\text{Obf}, \text{Smp}_0}^{\mathcal{A}}(\lambda)] = 0$  by constructon. Namely, adversary  $\mathcal{A}$  never outputs 1 in game  $\text{VGB1}_{\text{Obf}, \text{Smp}_0}^{\mathcal{A}}(\lambda)$ , since it is given an obfuscation of the constant zero circuit  $C_0$ .

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

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

To compute  $\Pr [\text{Hit}]$ , we move from  $\text{VGB0}_{\text{Smp}, q}^{\mathcal{S}}(\lambda)$  to the simpler  $\text{VGB0}_{\text{Smp}_0, q}^{\mathcal{S}}(\lambda)$ , where all of  $\mathcal{S}$ 's queries are answered with 0. We extend the latter game to sample

a random key  $k \leftarrow_s \mathbb{Z}_{p(\lambda)}$ , and we define  $\text{Hit}'$  as the event in this game that for one of  $S$ 's queries  $(g, K)$  we have  $g^k = K$ . It is not hard to see that  $\Pr[\text{Hit}']$  and  $\Pr[\text{Hit}]$  are equal, as both games are identical as long as none of such queries occur. Since there are at most  $q(\lambda)$  queries, and exactly one  $k$  can produce the answer 1 for these queries, the union bound yields

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

which concludes the proof. □

**The AI-DHI2 Assumption.** We now suggest a relaxation AI-DHI2 of the AI-DHI assumption given above. The idea is that for each value of  $\lambda$  there is not one, but many possible groups. Formally, a *group generator* is a PT algorithm  $\text{GG}$  that on input  $1^\lambda$  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  $\lambda$ , including performing group operations in  $\mathbb{G}$  and picking at random from  $\mathbb{G}$  and from the set  $\text{Gen}(\mathbb{G})$  of generators of  $\mathbb{G}$ . An auxiliary information generator  $\text{X}$  for  $\text{GG}$  is an auxiliary information generator as per Sect. 2 with the additional property that the target  $k$  returned by  $\text{X.Ev}(1^\lambda)$  is in  $\mathbb{Z}_{2^{\lambda-1}}$ —this makes it a valid exponent for *any* group  $\mathbb{G}$  such that  $\langle \mathbb{G} \rangle \in [\text{GG}(1^\lambda)]$ —and the payload  $m$  is  $\varepsilon$  (i.e. is effectively absent).

Now consider game  $\text{AIDHI2}$  of Fig. 4 associated to  $\text{GG}, \text{X}$  and an adversary  $\mathcal{A}$ . For  $\lambda \in \mathbb{N}$  let  $\text{Adv}_{\text{GG}, \text{X}, \mathcal{A}}^{\text{aidhi2}}(\lambda) = 2\Pr[\text{AIDHI2}_{\text{GG}, \text{X}}^{\mathcal{A}}(\lambda)] - 1$ . We say that  $\text{GG}$  is AI-DHI2-secure if  $\text{Adv}_{\text{GG}, \text{X}, \mathcal{A}}^{\text{aidhi2}}(\cdot)$  is negligible for every unpredictable  $\text{X}$  for  $\text{GG}$  and every PT adversary  $\mathcal{A}$ . The (new) AI-DHI2 assumption is that there exists a group generator  $\text{GG}$  which is AI-DHI2 secure.

BP [15] give a simple construction of AIPO from AI-DHI. It is easy to extend this to use AI-DHI2.

A *verifier* for group generator  $\text{GG}$  is a deterministic, PT algorithm  $\text{GG.Vf}$  that can check whether a given string  $d$  is a valid description of a group generated by the generator  $\text{GG}$ . Formally,  $\text{GG.Vf}$  on input  $1^\lambda, d$  returns **true** if  $d \in [\text{GG}(1^\lambda)]$  and **false** otherwise, for all  $d \in \{0, 1\}^*$ . We say that  $\text{GG}$  is *verifiable* if it has a verifier and additionally, in time polynomial in  $1^\lambda, \langle \mathbb{G} \rangle$ , where  $\langle \mathbb{G} \rangle \in [\text{GG}(1^\lambda)]$ , one can test membership in  $\mathbb{G}$  and in the set  $\text{Gen}(\mathbb{G})$  of generators of  $\mathbb{G}$ . The following extends Theorem 1 to say that if VGB0 is possible then no verifiable group generator is AI-DHI2 secure.

**Theorem 2.** *Let  $\text{GG}$  be a verifiable group generator. Then there is a pair  $\text{Smp}, \text{Smp}_0$  of PT samplers such that if there exists a VGB-secure obfuscator for the class  $\text{SAMP} = \{\text{Smp}, \text{Smp}_0\}$ , then  $\text{GG}$  is not AI-DHI2-secure.*

We omit a full proof, as it is very similar to the one of Theorem 1. We only note that to adapt the proof, we require  $\text{X.Ev}(1^\lambda)$  to output a random  $k$  in  $\mathbb{Z}_{2^{\lambda-1}}$  together with the obfuscation of the following circuit  $C_{1^\lambda, k}$ . 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  $\text{GG.Vf}$  on input  $1^\lambda, d$  to check whether  $d \in [\text{GG}(1^\lambda)]$ ,

returning 0 if the check fails. If the check succeeds, so that we can write  $d = \langle \mathbb{G} \rangle$ , it further checks that  $g \in \text{Gen}(\mathbb{G})$  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  $\text{GG}$  that on input  $1^\lambda$  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 2. However there are group generators that do not appear to be verifiable and where Theorem 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.

**Discussion.** As we indicated, one of the main applications of AI-DHI was AIPO [15], and furthermore this connection is very direct. If VGB is in contention with AI-DHI, it is thus natural to ask whether it is also in contention with AIPO. We do not know whether or not this is true. One can also ask whether VGB is in contention with other, particular AIPO constructions, in particular the one of BP [15] based on the construction of Wee [43]. Again, we do not know the answer. We note that alternative constructions of AIPO and other forms of point-function obfuscation are provided in [8].

## 4 KM-Leakage Resilient Encryption

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 [30]. Such schemes have been designed in [26, 30]. 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 Theorem 3 shows that KM-leakage-resilience is in contention with iO. The interpretation is KM-leakage-resilience is not achievable.

Theorem 3 is of direct interest with regard to understanding what is and is not achievable in leakage-resilient cryptography. But for us its main importance will be as a tool to rule out UCE for computationally unpredictable split sources assuming iO in Sect. 5.

We use standard definitions of indistinguishability obfuscation [2, 4, 18, 31, 42] and pseudorandom generators [17, 44], as recalled in Sect. 2. We now start by formalizing KM-leakage resilience.

**KM-Leakage Resilient Encryption.** Let a symmetric encryption scheme SE specify the following. PT encryption algorithm  $\text{SE.Enc}$  takes  $1^\lambda$ , a key

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

**Fig. 5.** Games defining X-KM-leakage resilience of symmetric encryption scheme SE and decryption correctness of symmetric encryption scheme SE.

$k \in \{0, 1\}^{\text{SE.kl}(\lambda)}$  and a message  $m \in \{0, 1\}^{\text{SE.ml}(\lambda)}$  to return a ciphertext  $c$ , where  $\text{SE.kl}, \text{SE.ml}: \mathbb{N} \rightarrow \mathbb{N}$  are the key length and message length functions of SE, respectively. Deterministic PT decryption algorithm  $\text{SE.Dec}$  takes  $1^\lambda, k, c$  to return a plaintext  $m \in \{0, 1\}^{\text{SE.ml}(\lambda)}$ . Note that there is a key length but no prescribed key-generation algorithm.

For security, let  $X$  be an auxiliary information generator with  $X.\text{tl} = \text{SE.kl}$  and  $X.\text{pl} = \text{SE.ml}$ . Consider game  $\text{IND}_{\text{SE},X}^A(\lambda)$  of Fig. 5 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  $\text{Adv}_{\text{SE},X,\mathcal{A}}^{\text{ind}}(\lambda) = 2 \Pr[\text{IND}_{\text{SE},X}^A(\lambda)] - 1$ . We say that SE is X-KM-leakage resilient if the function  $\text{Adv}_{\text{SE},X,\mathcal{A}}^{\text{ind}}(\cdot)$  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  $\text{SE.Dec}(1^\lambda, k, \text{SE.Enc}(1^\lambda, k, m)) = m$  for all  $k \in \{0, 1\}^{\text{SE.kl}(\lambda)}$ , all  $m \in \{0, 1\}^{\text{SE.ml}(\lambda)}$  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  $\text{DEC}_{\text{SE}}(\lambda)$  of Fig. 5 associated to SE, and for  $\lambda \in \mathbb{N}$  let  $\text{Adv}_{\text{SE}}^{\text{dec}}(\lambda) = \Pr[\text{DEC}_{\text{SE}}(\lambda)]$  be the decryption correctness function of SE. We require that  $\text{Adv}_{\text{SE}}^{\text{dec}}(\cdot)$  be non-negligible.

$\neg\text{iO} \vee \neg\text{KM-LR-SE}$ . The following says that KM-leakage-resilient symmetric encryption is not achievable if iO and PRGs (which can be obtained from one-way functions [38]) exist:

**Theorem 3.** *Let SE be a symmetric encryption scheme. Let Obf be an indistinguishability obfuscator. Let R be a PR-secure PRG with  $R.\text{sl} = \text{SE.ml}$ . Assume that  $2^{-\text{SE.kl}(\lambda)}$  and  $2^{-R.\text{sl}(\lambda)}$  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 is a minor adaptation of the proof of BM [22] ruling out MB-AIPO under iO. Following BM [22], 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  $\overline{C}$  of C. The attack showing claim (2) of Theorem 3 is straightforward but its analysis is more work and exploits the security of the PRG. Next one shows that iO-security of the obfuscator coupled with security of the PRG implies claim (1), namely the unpredictability of X. For completeness we provide a self-contained proof in Appendix A. A consequence of Theorem 3 is the following.

**Corollary 4.** *Let SE be a symmetric encryption scheme such that  $\text{SE.ml}(\cdot) \in \Omega((\cdot)^\epsilon)$  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 (Corollary 4).* The assumption on SE.ml implies that there exists a PR-secure PRG R with  $R.\text{sl} = \text{SE.ml}$  [38]. To conclude we apply Theorem 3.  $\square$

**Related Work.** CKVW [26] show that symmetric encryption with weak keys satisfying a wrong key detection property is equivalent to MB-AIPO. Wrong key detection, a form of robustness [1], asks that, if you decrypt, under a certain key, a ciphertext created under a different key, then the result is  $\perp$ . This is not a requirement for KM-LR-SE. However, implicit in the proof of Theorem 3 is a connection between KM-LR-SE and a form of MB-AIPO with a relaxed correctness condition.

## 5 UCE for Split Sources

BFM [20] showed that  $\text{UCE}[\mathbf{S}^{\text{cup}}]$ -security is not possible if iO exists. We improve this to show that  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{split}}]$ -security is not possible if iO exists. We obtain this by giving a construction of a KM-leakage-resilient symmetric encryption scheme from  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{split}}]$  and then invoking our above-mentioned result. Definitions of UCE-secure function families [6] are recalled in Sect. 2.



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

<p style="margin: 0;"><u>Algorithm <math>\text{SE}.\text{Enc}(1^\lambda, k, m)</math></u>  <math>hk \leftarrow_{\mathcal{S}} \mathbf{H}.\text{Kg}(1^\lambda)</math>                  For <math>i = 1, \dots,  m </math> do  <math>\mathbf{y}[i] \leftarrow \mathbf{H}.\text{Ev}(1^\lambda, hk, k \  m[i] \  \langle i \rangle_{\ell(\lambda)})</math>                  Return <math>(hk, \mathbf{y})</math></p>	<p style="margin: 0;"><u>Algorithm <math>\text{SE}.\text{Dec}(1^\lambda, k, (hk, \mathbf{y}))</math></u>                  For <math>i = 1, \dots,  \mathbf{y} </math> do                  If <math>(\mathbf{H}.\text{Ev}(1^\lambda, hk, k \  1 \  \langle i \rangle_{\ell(\lambda)}) = \mathbf{y}[i])</math>                  Then <math>m[i] \leftarrow 1</math> else <math>m[i] \leftarrow 0</math>                  Return <math>m</math></p>
---	--

Here  $\langle i \rangle_{\ell(\lambda)} = 1^i 0^{\ell(\lambda)-i}$  denotes a particular, convenient encoding of integer  $i \in \{1, \dots, \ell(\lambda)\}$  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  $\mathbf{H}$  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  $\mathbf{H}$  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  $\mathbf{H}$ .

**Theorem 5.** *Let  $\mathbf{H}$  be a family of functions that is  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{spl}}]$ -secure. Assume  $\mathbf{H}.\text{il}(\cdot) \in \Omega((\cdot)^\epsilon)$  for some constant  $\epsilon > 0$  and  $2^{-\mathbf{H}.\text{ol}(\cdot)}$  is negligible. Let  $\text{SE} = \mathbf{H}\&\mathbf{C}[\mathbf{H}]$ . Then (1) symmetric encryption scheme  $\text{SE}$  is KM-leakage resilient, and (2)  $1 - \text{Adv}_{\text{SE}}^{\text{dec}}(\cdot)$  is negligible.*

*Proof (Theorem 5).* Assuming for simplicity as in the construction that  $\mathbf{H}.\text{il}$  is odd, let  $\ell(\cdot) = (\mathbf{H}.\text{il}(\cdot) - 1)/2$ . We now prove part (1). Let  $\mathbf{X}$  be an unpredictable, uniform auxiliary information generator. Let  $\mathcal{A}$  be a PT adversary. We build a PT source  $\mathcal{S} \in \mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{spl}}$  and a PT distinguisher  $\mathcal{D}$  such that

$$\text{Adv}_{\text{SE}, \mathbf{X}, \mathcal{A}}^{\text{ind}}(\lambda) \leq 2 \cdot \text{Adv}_{\mathbf{H}, \mathcal{S}, \mathcal{D}}^{\text{uce}}(\lambda) \tag{3}$$

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

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

<p style="margin: 0;"><u>Algorithm <math>\mathcal{S}_0(1^\lambda)</math></u>  <math>(k, m_1, a) \leftarrow_{\mathcal{S}} \mathbf{X}.\text{Ev}(1^\lambda)</math>  <math>m_0 \leftarrow_{\mathcal{S}} \{0, 1\}^{\ell(\lambda)}</math>; <math>d \leftarrow_{\mathcal{S}} \{0, 1\}</math>                  For <math>i = 1, \dots, \ell(\lambda)</math> do  <math>\mathbf{x}[i] \leftarrow k \  m_d[i] \  \langle i \rangle_{\ell(\lambda)}</math>                  Return <math>((d, a), \mathbf{x})</math></p>	<p style="margin: 0;"><u>Algorithm <math>\mathcal{S}_1(1^\lambda, \mathbf{y})</math></u>                  Return <math>\mathbf{y}</math></p>	<p style="margin: 0;"><u>Distinguisher <math>\mathcal{D}(1^\lambda, hk, L)</math></u>  <math>((d, a), \mathbf{y}) \leftarrow L</math>; <math>c \leftarrow (hk, \mathbf{y})</math>  <math>d' \leftarrow_{\mathcal{S}} \mathcal{A}(1^\lambda, a, c)</math>                  If <math>(d = d')</math> then <math>b' \leftarrow 1</math>                  Else <math>b' \leftarrow 0</math>                  Return <math>b'</math></p>
--	--	---

Here  $\mathcal{S}_0$  calls the auxiliary information generator  $\mathsf{X}$  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  $\mathsf{HASH}$  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  $\mathsf{UCE}_{\mathsf{H}}^{\mathcal{S},\mathcal{D}}(\lambda)$ , we claim that

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

from which Eq. (3) follows. The first equation above should be clear from the construction. For the second, when  $b = 0$ , we know that  $\mathsf{HASH}$  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  $\mathcal{S} \in \mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}$ . We have ensured that  $\mathcal{S} \in \mathbf{S}^{\text{splt}}$  by construction. The crucial remaining step is to show that  $\mathcal{S} \in \mathbf{S}^{\text{cup}}$ . This will exploit the assumed unpredictability of  $\mathsf{X}$ . Let  $\mathcal{P}$  be a PT predictor. We build PT adversary  $\mathcal{Q}$  such that

$$\mathsf{Adv}_{\mathcal{S},\mathcal{P}}^{\text{pred}}(\lambda) \leq \mathsf{Adv}_{\mathsf{X},\mathcal{Q}}^{\text{pred}}(\lambda) \tag{4}$$

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

Adversary  $\mathcal{Q}(1^\lambda, a)$   
 For  $i = 1, \dots, \ell(\lambda)$  do  $\mathbf{y}[i] \leftarrow_{\mathcal{S}} \{0, 1\}^{\mathsf{H.ol}(\lambda)}$   
 $d \leftarrow_{\mathcal{S}} \{0, 1\}$ ;  $x' \leftarrow_{\mathcal{P}}(1^\lambda, ((d, a), \mathbf{y}))$ ;  $k \leftarrow x'[1..\ell(\lambda)]$ ; Return  $k$

Adversary  $\mathcal{Q}$  computes leakage  $((d, a), \mathbf{y})$  distributed exactly as it would be in game  $\mathsf{PRED}_{\mathcal{S}}^{\mathcal{P}}(\lambda)$ , where  $\mathsf{HASH}$  is a random oracle. It then runs  $\mathcal{P}$  to get a prediction  $x'$  of some oracle query of  $\mathcal{S}$ . If game  $\mathsf{PRED}_{\mathcal{S}}^{\mathcal{P}}(\lambda)$  returns **true**, then  $x'$  must have the form  $k\|m_d[i]\|\langle i \rangle_{\ell(\lambda)}$  for some  $i \in \{1, \dots, \ell(\lambda)\}$ , where  $k, d$  are the key and challenge bit, respectively, chosen by  $\mathcal{S}$ . Adversary  $\mathcal{Q}$  can then win its  $\mathsf{PRED}_{\mathsf{X}}^{\mathcal{Q}}(\lambda)$  game by simply returning  $k$ , which establishes Eq. (4).

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

$$1 - \mathsf{Adv}_{\mathsf{SE}}^{\text{dec}}(\lambda) \leq \mathsf{Adv}_{\mathsf{H},\mathcal{S},\mathcal{D}}^{\text{uce}}(\lambda) + \frac{\ell(\lambda)}{2^{\mathsf{H.ol}(\cdot)}} \tag{5}$$

for all  $\lambda \in \mathbb{N}$ . But we have assumed that  $\mathsf{H}$  is  $\mathsf{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{splt}}]$ -secure, so it is also  $\mathsf{UCE}[\mathbf{S}^{\text{sup}} \cap \mathbf{S}^{\text{splt}}]$ -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  $\mathcal{D}$ :

Algorithm $\mathcal{S}_0(1^\lambda)$ $k \leftarrow_s \{0, 1\}^{\ell(\lambda)}$ For $i = 1, \dots, \ell(\lambda)$ do $\quad \mathbf{x}[2i - 1] \leftarrow k \  1 \  \langle i \rangle_{\ell(\lambda)}$ $\quad \mathbf{x}[2i] \leftarrow k \  0 \  \langle i \rangle_{\ell(\lambda)}$ Return $(\varepsilon, \mathbf{x})$	$\left  \right.$	Algorithm $\mathcal{S}_1(1^\lambda, \mathbf{y})$ Return $\mathbf{y}$	Distinguisher $\mathcal{D}(1^\lambda, hk, (\varepsilon, \mathbf{y}))$ $b' \leftarrow 0$ For $i = 1, \dots, \ell(\lambda)$ do $\quad$ If $(\mathbf{y}[2i - 1] = \mathbf{y}[2i])$ $\quad$ Then $b' \leftarrow 1$ Return $b'$
---	------------------	---	---

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

$$\Pr[b' = 1 \mid b = 1] \geq 1 - \text{Adv}_{\text{SE}}^{\text{dec}}(\lambda) \quad \text{and} \quad \Pr[b' = 1 \mid b = 0] \leq \frac{\ell(\lambda)}{2^{\text{H.ol}(\lambda)}},$$

from which Eq. (5) 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  $2^{-\text{H.ol}(\lambda)}$ , 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  $\mathcal{S} \in \mathbf{S}^{\text{sup}}$ . 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  $\mathcal{S}$  means that  $\text{Adv}_{\mathcal{S}, \mathcal{P}}^{\text{pred}}(\cdot) \leq 2^{-\ell(\cdot)}$ , even if  $\mathcal{P}$  is not restricted to PT. But our assumption on  $\text{H.il}(\cdot)$  in the theorem statement implies that  $2^{-\ell(\cdot)}$  is negligible.  $\square$

$\neg \text{iO} \vee \neg \text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{spl}}]$ . In the BFM [20] iO-based attack on  $\text{UCE}[\mathbf{S}^{\text{cup}}]$ , 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 [6] 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  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{spl}}]$ . However, we show that in fact  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{spl}}]$ -security is still not achievable assuming iO. This is now a simple corollary of Theorems 3 and 5 that in particular was the motivation for the latter:

**Theorem 6.** *Let  $\text{H}$  be a family of functions such that  $\text{H.il}(\cdot) \in \Omega((\cdot)^\epsilon)$  for some constant  $\epsilon > 0$  and  $2^{-\text{H.ol}(\cdot)}$  is negligible. Assume the existence of an indistinguishability obfuscator and a one-way function. Then  $\text{H}$  is not  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{spl}}]$ -secure.*

BM2 [23] show that  $\text{UCE}[\mathbf{S}^{\text{cup}} \cap \mathbf{S}^{\text{spl}} \cap \mathbf{S}^1]$ -security is achievable assuming iO and AIPO. Our negative result of Theorem 6 does not contradict this, and in fact complements it to give a full picture of the achievability of UCE security for split sources.

**Acknowledgments.** Bellare and Stepanovs were supported in part by NSF grants CNS-1116800, CNS-1228890 and CNS-1526801. Tessaro was supported in part by NSF grant CNS-1423566. This work was done in part while Bellare and Tessaro were visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant CNS-1523467.

We thank Huijia Lin for discussions and insights. We thank the TCC 2016-A reviewers for extensive and insightful comments.

### A Proof of Theorem 3

The construction and proof follow [22]. We specify uniform auxiliary information generator  $X$  as follows:

$\begin{array}{l} \text{Algorithm } X.\text{Ev}(1^\lambda) \\ \hline k \leftarrow_s \{0, 1\}^{\text{SE.kl}(\lambda)} \\ m \leftarrow_s \{0, 1\}^{\text{SE.ml}(\lambda)} ; y \leftarrow \text{R.Ev}(1^\lambda, m) \\ C \leftarrow \text{Pad}_s(\lambda)(C_{1^\lambda, k, y}) ; \bar{C} \leftarrow_s \text{Obf}(1^\lambda, C) \\ \text{Return } (k, m, \bar{C}) \end{array}$	$\begin{array}{l} \text{Circuit } C_{1^\lambda, k, y}(c) \\ \hline m \leftarrow \text{SE.Dec}(1^\lambda, k, c) \\ y' \leftarrow \text{R.Ev}(1^\lambda, m) \\ \text{If } (y = y') \text{ then return } 1 \\ \text{Else return } 0 \end{array}$
--	---

The circuit  $C_{1^\lambda, k, y}$  takes as input a ciphertext  $c$ , decrypts it under the embedded key  $k$  to get back a  $\text{SE.ml}(\lambda)$ -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 polynomial  $s$  so that  $s(\lambda)$  is an upper bound on  $\max(|C_{1^\lambda, k, y}^1|, |C^2|)$  where the circuits are defined in Fig. 6 and the maximum is over all  $k \in \{0, 1\}^{\text{SE.kl}(\lambda)}$  and  $y \in \{0, 1\}^{2 \cdot \text{R.sl}(\lambda)}$ . Let us first present an attack proving part (2) of the theorem. 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:

$\begin{array}{l} \text{Adversary } \mathcal{A}(1^\lambda, \bar{C}, c) \\ \hline b' \leftarrow \bar{C}(c) \\ \text{Return } b' \end{array}$	$\begin{array}{l} \text{Adversary } \mathcal{R}(1^\lambda, y) \\ \hline k \leftarrow_s \{0, 1\}^{\text{SE.kl}(\lambda)} ; m_0 \leftarrow_s \{0, 1\}^{\text{SE.ml}(\lambda)} \\ c \leftarrow_s \text{SE.Enc}(1^\lambda, k, m_0) ; m \leftarrow \text{SE.Dec}(1^\lambda, k, c) \\ y' \leftarrow \text{R.Ev}(1^\lambda, m) \\ \text{If } (y' = y) \text{ then } g' \leftarrow 1 \text{ else } g' \leftarrow 0 ; \text{Return } g' \end{array}$
---	---

Adversary  $\mathcal{A}$  has input  $1^\lambda$ , the auxiliary information (leakage) which here is the obfuscated circuit  $\bar{C}$ , and a ciphertext  $c$ . It simply computes and returns the bit  $\bar{C}(c) = C_{1^\lambda, k, y}(c)$ . For the analysis, consider game  $\text{IND}_{\text{SE}, X}^A(\lambda)$  of Fig. 5. If the challenge bit  $b$  is 1 and the decryption performed by  $\bar{C}$  is correct then  $y' = y$ , so

$$\Pr[b' = 1 \mid b = 1] \geq \text{Adv}_{\text{SE}}^{\text{dec}}(\lambda). \tag{6}$$

In the case  $b = 0$ , the corresponding analysis in [22] 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  $\text{SE.Dec}(1^\lambda, k, c)$  equals  $m_0$

and indeed have no guarantees on the distribution of decrypted plaintext message. Instead, we directly exploit the assumed PR-security of the PRG. Thus, consider game  $\text{PRG}_R^{\mathcal{R}}(\lambda)$  with adversary  $\mathcal{R}$  as above. Letting  $g$  denote the challenge bit in the game, we have

$$\begin{aligned} \text{Adv}_{R,\mathcal{R}}^{\text{pr}}(\lambda) &= \Pr[g' = 1 \mid g = 1] - \Pr[g' = 1 \mid g = 0] \\ &\geq \Pr[b' = 1 \mid b = 0] - 2^{-2 \cdot R \cdot \text{sl}(\lambda)}. \end{aligned} \tag{7}$$

From Eqs. (7) and (6), we have

$$\begin{aligned} \text{Adv}_{\text{SE},\mathcal{X},\mathcal{A}}^{\text{ind}}(\lambda) &= \Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0] \\ &\geq \text{Adv}_{\text{SE}}^{\text{dec}}(\lambda) - \text{Adv}_{R,\mathcal{R}}^{\text{pr}}(\lambda) - 2^{-2 \cdot R \cdot \text{sl}(\lambda)}. \end{aligned} \tag{8}$$

Our weak correctness condition implies that the first term of Eq. (8) is non-negligible. On the other hand, the second and third terms are negligible. This means  $\text{Adv}_{\text{SE},\mathcal{X},\mathcal{A}}^{\text{ind}}(\cdot)$  is not negligible, proving claim (2) of Theorem 3.

Games $G_0$ - $G_3$	
$k \leftarrow_{\$} \{0, 1\}^{\text{SE.kl}(\lambda)} ; m \leftarrow_{\$} \{0, 1\}^{\text{SE.ml}(\lambda)}$	
$y \leftarrow \text{R.Ev}(1^\lambda, m) ; \bar{C} \leftarrow_{\$} \text{Obf}(1^\lambda, \text{Pad}_{s(\lambda)}(C_{1^\lambda, k, y}^1)) \quad // G_0$	
$y \leftarrow_{\$} \{0, 1\}^{2 \cdot R \cdot \text{sl}(\lambda)} ; \bar{C} \leftarrow_{\$} \text{Obf}(1^\lambda, \text{Pad}_{s(\lambda)}(C_{1^\lambda, k, y}^1)) \quad // G_1$	
$y \leftarrow_{\$} \{0, 1\}^{2 \cdot R \cdot \text{sl}(\lambda)} ; \bar{C} \leftarrow_{\$} \text{Obf}(1^\lambda, \text{Pad}_{s(\lambda)}(C^2)) \quad // G_2$	
$k' \leftarrow_{\$} \mathcal{Q}(1^\lambda, \bar{C}) ; \text{Return } (k = k')$	
Circuit $C_{1^\lambda, k, y}^1(c)$	Circuit $C^2(c)$
$m \leftarrow \text{SE.Dec}(1^\lambda, k, c) ; y' \leftarrow \text{R.Ev}(1^\lambda, m)$	
If $(y = y')$ then return 1 else return 0	

**Fig. 6.** Games for proof of part (1) of Theorem 3.

We proceed to prove part (1) of the theorem statement. Let  $\mathcal{Q}$  be a PT adversary. Consider the games and associated circuits of Fig. 6. Lines not annotated with comments are common to all three games. Game  $G_0$  is equivalent to  $\text{PRED}_{\mathcal{X}}^{\mathcal{Q}}(\lambda)$ , so

$$\text{Adv}_{\mathcal{X},\mathcal{Q}}^{\text{pred}}(\lambda) = \Pr[G_2] + (\Pr[G_0] - \Pr[G_1]) + (\Pr[G_1] - \Pr[G_2]). \tag{9}$$

We have  $\Pr[G_2] = 2^{-\text{SE.kl}(\lambda)}$ , where the latter is assumed to be negligible, because  $k$  is uniformly random and the circuit  $\bar{C}$  that is passed to adversary  $\mathcal{Q}$  does not depend on  $k$ . We now show that  $\Pr[G_i] - \Pr[G_{i+1}]$  is negligible for  $i \in \{0, 1\}$ , which by Eq. (9) implies that  $\text{Adv}_{\mathcal{X},\mathcal{Q}}^{\text{pred}}(\cdot)$  is negligible and hence proves the claim.

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

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

We have  $\Pr[G_0] - \Pr[G_1] = \text{Adv}_{\mathcal{R}, \mathcal{R}}^{\text{Pr}}(\lambda)$ , where the advantage is negligible by the assumed PR-security of  $\mathcal{R}$ .

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

<p>Circuit Sampler <math>\mathcal{S}(1^\lambda)</math></p> <p><math>k \leftarrow_{\mathcal{S}} \{0, 1\}^{\text{SE.kl}(\lambda)} ; y \leftarrow_{\mathcal{S}} \{0, 1\}^{2 \cdot \text{R.sl}(\lambda)}</math></p> <p><math>C_1 \leftarrow \text{Pad}_{s(\lambda)}(C_{1^\lambda, k, y}^1) ; C_0 \leftarrow \text{Pad}_{s(\lambda)}(C^2)</math></p> <p><math>aux \leftarrow k ; \text{return } (C_0, C_1, aux)</math></p>	<p>Adversary <math>\mathcal{O}(1^\lambda, \overline{C}, aux)</math></p> <p><math>k \leftarrow aux ; k' \leftarrow_{\mathcal{S}} \mathcal{Q}(1^\lambda, \overline{C})</math></p> <p>If <math>(k = k')</math> then return 1</p> <p>Else return 0</p>
---	--

It follows that  $\Pr[G_1] - \Pr[G_2] = \text{Adv}_{\text{Obf}, \mathcal{S}, \mathcal{O}}^{\text{iO}}(\lambda)$ . We now show that  $\mathcal{S} \in \mathcal{S}_{\text{eq}}$ , and hence  $\text{Adv}_{\text{Obf}, \mathcal{S}, \mathcal{O}}^{\text{iO}}(\lambda)$  is negligible by the assumed iO-security of  $\text{Obf}$ . Specifically, note that  $C_{1^\lambda, k, y}^1$  and  $C^2$  are not equivalent only if  $y$  belongs to the range of  $\mathcal{R}$ , which contains at most  $2^{\text{R.sl}(\lambda)}$  values. However,  $y$  is sampled uniformly at random from a set of size  $2^{2 \cdot \text{R.sl}(\lambda)}$ . It follows that

$$\Pr [ C_0 \equiv C_1 : (C_0, C_1, aux) \leftarrow_{\mathcal{S}} \mathcal{S}(1^\lambda) ] \geq 1 - 2^{-\text{R.sl}(\lambda)},$$

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

## References

1. Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 480–497. Springer, Heidelberg (2010)
2. Ananth, P., Boneh, D., Garg, S., Sahai, A., Zhandry, M.: Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689 (2013). <http://eprint.iacr.org/2013/689>
3. Barak, B., Garg, S., Kalai, Y.T., Paneth, O., Sahai, A.: Protecting obfuscation against algebraic attacks. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 221–238. Springer, Heidelberg (2014)
4. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)
5. Bellare, M., Hoang, V.T.: Resisting randomness subversion: fast deterministic and hedged public-key encryption in the standard model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 627–656. Springer, Heidelberg (2015)
6. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating Random Oracles via UCEs. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 398–415. Springer, Heidelberg (2013)
7. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
8. Bellare, M., Stepanovs, I.: Point-function obfuscation: a framework and generic constructions. In: Kushilevitz, E., Malkin, T., (eds.) TCC 2016-A, Part II. LNCS, vol. 9563, pp. 565–594. Springer, Heidelberg (2016)
9. Bellare, M., Stepanovs, I., Tessaro, S.: Poly-many hardcore bits for any one-way function and a framework for differing-inputs obfuscation. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 102–121. Springer, Heidelberg (2014)

10. Bitansky, N., Canetti, R.: On Strong Simulation and Composable Point Obfuscation. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 520–537. Springer, Heidelberg (2010)
11. Bitansky, N., Canetti, R., Cohn, H., Goldwasser, S., Kalai, Y.T., Paneth, O., Rosen, A.: The impossibility of obfuscation with auxiliary input or a universal simulator. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 71–89. Springer, Heidelberg (2014)
12. Bitansky, N., Canetti, R., Kalai, Y.T., Paneth, O.: On virtual grey box obfuscation for general circuits. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 108–125. Springer, Heidelberg (2014)
13. Bitansky, N., Canetti, R., Kalai, Y.T., Paneth, O.: On virtual grey box obfuscation for general circuits. Cryptology ePrint Archive, Report 2014/554 (2014). <http://eprint.iacr.org/2014/554>
14. Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: On the existence of extractable one-way functions. In: Shmoys, D.B. (ed.) 46th ACM STOC, pp. 505–514. ACM Press, May / June (2014)
15. Bitansky, N., Paneth, O.: Point obfuscation and 3-round zero-knowledge. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 190–208. Springer, Heidelberg (2012)
16. Bitansky, N., Paneth, O., On the impossibility of approximate obfuscation and applications to resettable cryptography. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC, pp. 241–250. ACM Press, June 2013
17. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudorandom bits. *SIAM J. Comput.* **13**(4), 850–864 (1984)
18. Boyle, E., Chung, K.-M., Pass, R.: On extractability obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 52–73. Springer, Heidelberg (2014)
19. Brakerski, Z., Rothblum, G.N.: Virtual black-box obfuscation for all circuits via generic graded encoding. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 1–25. Springer, Heidelberg (2014)
20. Brzuska, C., Farshim, P., Mittelbach, A.: Indistinguishability Obfuscation and UCEs: The Case of Computationally Unpredictable Sources. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 188–205. Springer, Heidelberg (2014)
21. Brzuska, C., Farshim, P., Mittelbach, A.: Random-oracle uninstantiability from indistinguishability obfuscation. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 428–455. Springer, Heidelberg (2015)
22. Brzuska, C., Mittelbach, A.: Indistinguishability Obfuscation versus Multi-bit Point Obfuscation with Auxiliary Input. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 142–161. Springer, Heidelberg (2014)
23. Brzuska, C., Mittelbach, A.: Using indistinguishability obfuscation via UCEs. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 122–141. Springer, Heidelberg (2014)
24. Brzuska, C., Mittelbach, A.: Universal computational extractors and the superfluous padding assumption for indistinguishability obfuscation. Cryptology ePrint Archive, Report 2015/581 (2015). <http://eprint.iacr.org/2015/581>
25. Canetti, R.: Towards realizing random oracles: hash functions that hide all partial information. In: Kaliski Jr, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 455–469. Springer, Heidelberg (1997)
26. Canetti, R., Tauman Kalai, Y., Varia, M., Wichs, D.: On Symmetric Encryption and Point Obfuscation. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 52–71. Springer, Heidelberg (2010)



27. Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehlé, D.: Cryptanalysis of the multilinear map over the integers. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 3–12. Springer, Heidelberg (2015)
28. Coron, J.-S., Lepoint, T., Tibouchi, M.: Cryptanalysis of two candidate fixes of multilinear maps over the integers. Cryptology ePrint Archive, Report 2014/975 (2014). <http://eprint.iacr.org/2014/975>
29. Dodis, Y., Ganesh, C., Golovnev, A., Juels, A., Ristenpart, T.: A formal treatment of backdoored pseudorandom generators. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 101–126. Springer, Heidelberg (2015)
30. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: Mitzenmacher, M. (ed.) 41st ACM STOC, pp. 621–630. ACM Press, May / June (2009)
31. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS, pp. 40–49. IEEE Computer Society Press October 2013
32. Garg, S., Gentry, C., Halevi, S., Wichs, D.: On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 518–535. Springer, Heidelberg (2014)
33. Gentry, C., Halevi, S., Maji, H.K., Sahai, A.: Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero. Cryptology ePrint Archive, Report 2014/929 (2014). <http://eprint.iacr.org/2014/929>
34. Gentry, C., Lewko, C.A., Sahai, A., Waters, B.: Indistinguishability obfuscation from the multilinear subgroup elimination assumption. Cryptology ePrint Archive, Report 2014/309 (2014). <http://eprint.iacr.org/2014/309>
35. Goldwasser, S., Kalai, Y.T.: On the impossibility of obfuscation with auxiliary input. In: 46th FOCS, pp. 553–562. IEEE Computer Society Press, October 2005
36. Green, M.D., Katz, J., Malozemoff, A.J., Zhou, H.-S.: A unified approach to idealized model separations via indistinguishability obfuscation. Cryptology ePrint Archive, Report 2014/863 (2014). <http://eprint.iacr.org/2014/863>
37. Hada, S.: Zero-knowledge and code obfuscation. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 443–457. Springer, Heidelberg (2000)
38. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)
39. Lee, H.T., Seo, J.H.: Security analysis of multilinear maps over the integers. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 224–240. Springer, Heidelberg (2014)
40. Matsuda, T., Hanaoka, G.: Chosen ciphertext security via UCE. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 56–76. Springer, Heidelberg (2014)
41. Pass, R., Seth, K., Telang, S.: Indistinguishability obfuscation from semantically-secure multilinear encodings. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 500–517. Springer, Heidelberg (2014)
42. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) 46th ACM STOC, pp. 475–484. ACM Press, May / June 2014
43. Wee, H.: On obfuscating point functions. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 523–532. ACM Press, May 2005
44. Yao, A.C.-C.: Theory and applications of trapdoor functions (extended abstract). In: 23rd FOCS, pp. 80–91. IEEE Computer Society Press, November (1982)