A Unified Approach to Idealized Model Separations via Indistinguishability Obfuscation

Matthew D. Green* Jonathan Katz† Alex J. Malozemoff† Hong-Sheng Zhou‡

Abstract

It is well known that the random oracle model is not sound in the sense that there exist cryptographic systems that are secure in the random oracle model but when instantiated by any family of hash functions become insecure. However, all known separation results require the attacker to send an appropriately crafted message to the challenger in order to break security. Thus, this leaves open the possibility that some cryptographic schemes, such as bit-encryption, are still sound in the random oracle model.

In this work we refute this possibility, assuming the existence of indistinguishability obfuscation. We do so in the following way. First, we present a random oracle separation for bit-encryption; namely, we show that there exists a bit-encryption protocol secure in the random oracle model but *completely insecure* when the random oracle is instantiated by any concrete function. Second, we show how to adapt this separation to work for most natural simulation-based and game-based definitions. Our techniques can easily be adapted to other idealized models, and thus we present a *unified approach* to showing separations for most protocols of interest in most idealized models.

1 Introduction

A common technique in cryptography is the use of *idealized models*, where one assumes oracle access to some (ideal) process. Idealized models provide powerful mechanisms for constructing elegant and simple protocols while still being able to provide some provable guarantees. Some common examples of such models are the random oracle model [6], the generic group model [33], and more recently, the generic graded encoding model [2, 10], among others.

Over the past several years, there has been significant interest in understanding the implications of using these models when developing cryptographic protocols. Indeed, it is well-known that such models do not match "reality" in the sense that there exist secure schemes in generic models which are not secure when concretely instantiated. The first work to present such a result was that of Canetti, Goldreich, and Halevi [14], who showed such a separation for the random oracle model. Following this result, Dent showed a similar separation for the generic group model [19]. Likewise, a separation for the generic graded encoding model has been demonstrated by Brakerski and Rothblum [10], who construct virtual black-box obfuscation in the generic graded encoding model even though it is known that virtual black-box obfuscation is impossible in the standard model [3].

*Dept. of Computer Science, Johns Hopkins University. **Email:** mgreen@cs.jhu.edu

†Dept. of Computer Science, University of Maryland. **Email:** {jkatz,amaloz}@cs.umd.edu

‡Dept. of Computer Science, Virginia Commonwealth University. **Email:** hszhou@vcu.edu

However, each of the above separation results employs different techniques and only covers a small subset of cryptographic schemes. For example, the random oracle separation presented by Canetti et al. [14] does not apply to CPA-secure bit-encryption [20]. Thus, the existing results have left open the possibility that certain protocol classes (such as bit-encryption) are not subject to the counterexamples described in prior work. We note that this omission may have practical implications due to the renewed interest in fully-homomorphic bit-encryption systems secure in the random oracle model [17, 24]. More fundamentally, it leaves us with critical gaps in our theoretical understanding of the security of cryptosystems analyzed in idealized models. Might these exceptions provide a "loophole" through which certain protocols could be safely instantiated?

Our results. In this work, we refute this notion, assuming the existence of indistinguishability obfuscation. Specifically, we show that if there exists a secure indistinguishability obfuscator (in the standard model), then for a large class of cryptographic tasks there exists a variant of said protocol which is secure in a given idealized model but completely insecure when concretely instantiated.

The core of our technique is using indistinguishability obfuscation to obfuscate a circuit which hides some secret information needed to break security. Finding the proper input to reveal this secret information in the idealized model is hard, whereas as soon as the idealized model is instantiated by some concrete function, it is easy to construct a proper input.

Our specific results are as follows:

1. *A counterexample for bit-encryption in the random oracle model.* Our first result is to show a separation for bit-encryption in the random oracle model (Section 3). In presenting our counterexample we solve a longstanding open problem raised by Dent [19]. Simultaneously we demonstrate the generality of our technique by observing that the same result cannot be derived using the original Canetti et al. [14] or Maurer et al. [31] approaches, since these do not work for bit-encryption.

   Specifically, our result shows that if an indistinguishability obfuscation scheme exists in the standard model, then for any length ℓ, there exists an IND-CPA secure bit-encryption scheme that is provably secure when a hash function h is instantiated using a random oracle, but becomes *insecure* when h is instantiated using any concrete function which can be represented in $\leq \ell$ bits. To achieve this result we employ the obfuscation of a universal circuit, and show how an adversary with knowledge of the description of h can win the IND-CPA game with non-negligible probability. As in previous counterexamples, the proposed scheme breaks *catastrophically* when instantiated using a concrete function by revealing the secret key for the scheme.

2. *A generic approach to constructing idealized model separations for cryptographic tasks.* Next, we generalize our initial result to show random oracle model separations for most natural protocols secure under simulation-based or game-based definitions (Section 4). These results can be easily adapted to apply to other idealized models, including the generic group model (see Appendix A), the random permutation model, etc. Thus, we present a unified approach to constructing separations for most cryptographic tasks of interest in most idealized models of interest.

These results deepen our understanding of how to define "secure" protocols and help us to understand the implications of these idealized models. They also provide additional justification

for the ongoing effort to develop new and *instantiable* assumptions/models in which we may analyze these protocols (e.g., the UCE framework [5]).

1.1 Related Work

The random oracle model was first introduced formally by Bellare and Rogaway [6]; this was also the first work to put forward the notion of an "idealized model" as a way to simplify both cryptographic constructions and proofs. However, soon after Canetti et al. [14] demonstrated that the random oracle model is not *sound* in the sense that there exist schemes secure in the random oracle model but completely insecure when instantiated in the standard model. This separation spawned a large body of work showing separations for both new classes of protocols [4, 15, 21, 25, 29, 30, 32] as well as other idealized models [19]. However, for the most part, each result uses different techniques, and thus it is not immediately clear whether separations exist for all classes of protocols as well as all idealized models.

In a separate line of work, since the breakthrough result of Garg et al. [22] demonstrating a candidate indistinguishability obfuscator, many researchers have studied the implications of indistinguishability obfuscation as well as other obfuscation definitions. Two groups of researchers concurrently looked at the problem of *differing-inputs obfuscation* [1, 9]; however, Garg et al. [23] showed that the existence of a differing-inputs obfuscator is "implausible" in the sense that there exists a reasonable assumption which, if true, means that differing-inputs obfuscators cannot exist. Likewise, assuming indistinguishability obfuscation exists, Bitansky et al. [7, 8] showed both that auxiliary-input extractable functions cannot exist and that virtual-black-box obfuscation cannot exist for a certain class of functions. Finally, Brzuska et al. [11] showed that using indistinguishability obfuscation, several definitions within the UCE framework [5] (a framework which is designed to replace the random oracle model in many settings) cannot be instantiated.

Finally, a recent line of work constructs schemes which use *both* indistinguishability obfuscation and the random oracle model [26, 27, 18]. Our results serve as a warning sign that combining these two approaches may lead to insecure schemes when the random oracle is instantiated in the standard model.

Comparison with the work of Brzuska et al. [12]. Concurrently and independently, Brzuska, Farshim, and Mittelbach [12] use a similar technique to ours to show that some cryptographic transformations based on random oracles are uninstantiable in the standard model. Their underlying technique is very similar to ours at a high level, in that they obfuscate a universal circuit taking as input the description of a hash function; the technical details, however, differ. Besides the similarity in the underlying technique, both works are in some sense orthogonal, as Brzuska et al. [12] show separations for cryptographic *transformations* whereas we show separations for cryptographic *constructions*.

2 Preliminaries

We let n denote the security parameter. For a function f, we let $\langle f \rangle$ denote the (binary) *description* of f. We use the notation $x \leftarrow\!\!\$\, S$ to denote that x is chosen uniformly at random from the set S. As our main results show separations in the random oracle model (although our results can be extended to other idealized models), we review this idealized model and how to instatiate it in the standard model using *function ensembles* [14].

Random oracle model. Let $\ell_{\text{out}} : \mathbb{N} \to \mathbb{N}$ be a length function. The random oracle model is defined by a (stateful) function $\mathcal{O} : \{0,1\}^* \to \{0,1\}^{\ell_{\text{out}}(n)}$ available to all parties which works as follows: $\mathcal{O}$ maintains an internal table T which stores inputs and their associated outputs. If $x \in T$, let $T(x)$ denote the associated output. On input x, If $x \in T$, then $\mathcal{O}$ outputs $T(x)$; otherwise, $\mathcal{O}$ chooses $y \leftarrow\!\!\$\, \{0,1\}^{\ell_{\text{out}}(n)}$, adds (x, y) to T, and outputs y.

Function ensembles. We use the notion of *function ensembles* introduced by Canetti et al. [14], and we reproduce it here mostly verbatim. The idea of a function ensemble is to capture the intuitive notion of what it means to "instantiate" a random oracle.

Let $\ell_{\text{out}} : \mathbb{N} \to \mathbb{N}$ be a length function. An ℓ_{out}-*ensemble* is a sequence $\mathcal{F} = \{F_n\}_{n\in\mathbb{N}}$ of families of functions $F_n = \{f_s : \{0,1\}^* \to \{0,1\}^{\ell_{\text{out}}(n)}\}_{s\in\{0,1\}^n}$ such that the following condition holds:

1. There exists a polynomial-time algorithm Eval such that for every $s \in \{0,1\}^n$ and $x \in \{0,1\}^*$ it holds that $\mathsf{Eval}(s, x) = f_s(x)$.

Let $\ell_{\text{eval}}(n)$ be the length of the bitstring representation of Eval for function family F_n; we have that $\ell_{\text{eval}}(n) \leq \mathsf{poly}(n)$.

Let an $(\ell_{\text{out}}, \ell_{\text{eval}})$-ensemble be an ℓ_{out}-ensemble such that the bitstring representation of Eval is $\leq \ell_{\text{eval}}$. In what follows, we in general do not care what the output length of the function is, as long as it is polynomial in the security parameter. We denote this class of ensembles as (poly, ℓ)-ensembles; that is, the class of ℓ'-ensembles such that $\ell' \in \mathsf{poly}(n)$ and the bitstring representation of Eval is $\leq \ell$.

All our constructions use *indistinguishability obfuscation*, defined as follows.

Indistinguishability obfuscation. Let $\{\mathcal{C}_\lambda\}$ be the class of circuits of size at most λ. We utilize the notion of *family-indistinguishability obfuscators* [3, 22], and we reproduce it here mostly verbatim.

Definition 2.1. *A uniform* PPT *machine* iO *is a* family-indistinguishability obfuscator *for a circuit class* $\{\mathcal{C}_\lambda\}$ *if the following two conditions hold:*

1. *For all* $\lambda \in \mathbb{N}$*, for all* $C \in \mathcal{C}_\lambda$*, and for all inputs* x*, it holds that*

$$\Pr\left[C'(x) = C(x) : C' \leftarrow \mathsf{iO}(1^\lambda, C)\right] = 1.$$

2. *For all (not necessarily uniform)* PPT *adversaries* Samp *and* $\mathcal{A}$*, there exists a negligible function* negl *such that if* $\Pr\left[\forall x, C_0(x) = C_1(x) : (C_0, C_1, \sigma) \leftarrow \mathsf{Samp}(1^\lambda)\right] > 1 - \mathsf{negl}(\lambda)$ *then*

$$\Big|\Pr[\mathcal{A}(\sigma, \mathsf{iO}(1^\lambda, C_0)) = 1 \;:\; (C_0, C_1, \sigma) \leftarrow \mathsf{Samp}(1^\lambda)] - \Pr[\mathcal{A}(\sigma, \mathsf{iO}(1^\lambda, C_1)) = 1 \;:\; (C_0, C_1, \sigma) \leftarrow \mathsf{Samp}(1^\lambda)]\Big| \leq \mathsf{negl}(\lambda).$$

In this work we are interested in the obfuscation of polynomial-size circuits, and thus we only consider $\lambda \leq \mathsf{poly}(n)$.

3 Random Oracle Separation for Bit-Encryption

As our first result, we present a random oracle separation for the case of (public-key) bit-encryption. Note that existing techniques for showing idealized model separations work by having the adversary send some specially-crafted message to an oracle; the oracle, given this message, leaks the secret key and thus the adversary can easily break security. However, in the case of bit-encryption, the only values an adversary can send are bits, and thus the above technique does not work in this setting.

Consider the security game $\mathsf{PubK}_{\mathcal{A},\Pi}$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ for a public-key bit-encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$:

1. $\mathcal{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$ and sends pk to $\mathcal{A}$.
2. $\mathcal{C}$ chooses $b \leftarrow\!\!\$\, \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(b)$ and sends c to $\mathcal{A}$.
3. $\mathcal{A}$ has oracle access to $\mathsf{Enc}_{\mathsf{pk}}$, and can query it a polynomial number of times. Eventually, $\mathcal{A}$ outputs a bit b' and succeeds if $b = b'$.

Definition 3.1 (IND-CPA security)**.** *A public-key bit-encryption scheme* Π *is* IND-CPA-secure *if for all* PPT *adversaries* $\mathcal{A}$ *there exists a negligible function* negl *such that* $\Pr[\mathsf{PubK}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$.

We prove the following theorem:

Theorem 3.1. *Assume that there exists an IND-CPA-secure public-key bit-encryption scheme and an indistinguishability obfuscator secure in the standard model. Then for all* $\ell \in \mathsf{poly}(n)$*, there exists a public-key bit-encryption scheme that is IND-CPA-secure in the random oracle model but insecure when the random oracle is instantiated using any* (poly, ℓ)*-ensemble.*

Proof. Let Obf be an indistinguishability obfuscator, let $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ be an existing IND-CPA-secure public-key bit-encryption scheme, and let $\mathcal{O}$ be a random oracle.

Intuition. Our construction, at a high level, works as follows. Taking an existing bit-encryption scheme, we modify it by appending an obfuscated circuit to the public-key. The obfuscated circuit is built as follows. We choose ℓn random values x_i and compute $y_i \leftarrow H(x_i)$, where H is either a random oracle or a function ensemble, depending on whether we are operating in the ROM or standard model.[1] The circuit hardcodes the values x_i and y_i, along with the secret key to the original bit-encryption scheme. On input a description of a hash function h, the circuit outputs the secret key if and only if $y_i = h(x_i)$ for all i. Note that in the random oracle model, it is unlikely that such a hash function can be found to satisfy $y_i = h(x_i)$ for all i, whereas in the standard model this is easily satisfied (since h is public).

Note that this approach is similar to that given by Maurer et al. [31], who provide an alternate proof of the separation result given by Canetti et al. [14]. The main difference is our use of indistinguishability obfuscation, which allows the adversary to break security in the standard model *without* needing to send messages to the challenger. This has some significant advantages, one of which is that, because no "special" messages need to be sent to the challenger, our approach works for bit-encryption where the encryption oracle only takes as input single bits.

Formal Analysis. Fix some value $\ell \in \mathsf{poly}(n)$. The scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is constructed as follows. Note that all algorithms are provided oracle access to $\mathcal{O}$.

[1] The reason we need ℓn values rather than, say, ℓ, is in the case where ℓ is a small constant, such as 1.

Constants: $x_1, \ldots, x_{\ell n}$, $y_1, \ldots, y_{\ell n}$, sk.
Input: a description $\langle h \rangle$ of a function h.

1. For $i \in \{1, \ldots, \ell n\}$, compute $\widehat{y_i} := h(x_i)$.
2. If for $i \in \{1, \ldots, \ell n\}$ it holds that $\widehat{y_i} = y_i$, then output sk; otherwise, output $\perp$.

Figure 1: Program C.

Constants: $x_1, \ldots, x_{\ell n}$, $y_1, \ldots, y_{\ell n}$, sk.
Input: a description $\langle h \rangle$ of a function h.

1. Output $\perp$.

Figure 2: Program C'.

- Gen: On input 1^n, proceed as follows. For $i \in \{1, \ldots, \ell n\}$, choose $x_i \leftarrow\!\!\$\, \{0,1\}^n$ and compute $y_i \leftarrow \mathcal{O}(x_i)$. Next, run $(\mathsf{pk}', \mathsf{sk}') \leftarrow \mathsf{Gen}'(1^n)$, and set $\mathsf{sk} := \mathsf{sk}'$. Then, create an obfuscation of the program C as described in Figure 1. Denote this obfuscation as $\mathsf{Obf}(C)$. Finally, let $\mathsf{pk} := (\mathsf{pk}', \mathsf{Obf}(C))$ and output $(\mathsf{pk}, \mathsf{sk})$.
- Enc: On input pk and bit b, parse pk as $(\mathsf{pk}', \mathsf{Obf}(C))$ and compute $c \leftarrow \mathsf{Enc}'_{\mathsf{pk}'}(b)$. Output c.
- Dec: On input private key $\mathsf{sk} = \mathsf{sk}'$ and ciphertext c, compute $m := \mathsf{Dec}'_{\mathsf{sk}'}(c)$. Output m.

Lemma 3.1.1. *Assume that Obf is an indistinguishability obfuscator. Then, for any choice of $\ell \in \mathsf{poly}(n)$ the construction Π is an IND-CPA-secure bit-encryption scheme in the random oracle model.*

Proof. Consider the following two hybrids.

Hybrid $\mathbf{H_0}$: This is the IND-CPA game for scheme Π.

Hybrid $\mathbf{H_1}$: This hybrid is the same as $\mathbf{H_0}$ except that now we change program C into program C' as in Figure 2.

Claim. *If Obf is an indistinguishability obfuscator in the standard model, then with high probability over the choices of the random oracle the two hybrids $\mathbf{H_0}$ and $\mathbf{H_1}$ are computationally indistinguishable.*

Proof. The proof is by a reduction to the security of the indistinguishability obfuscator. The proof relies on the fact that with high probability there is no "small representation" of a random oracle. That is, the probability that there exists a description $\langle h \rangle \in \{0,1\}^\ell$ of a function h such that for $i \in \{1, \ldots, \ell n\}$ it holds that $y_i = h(x_i)$ is negligible. Thus, with high probability over the choices of the random oracle, programs C and C' are equivalent, and thus we can reduce security to that of indistinguishability obfuscation.

More formally, consider the class $\mathsf{Func}_{\ell_{\mathrm{out}}(n)}$ of all functions mapping $x_1, \ldots, x_{\ell n}$ to $\ell_{\mathrm{out}}(n)$-bit outputs; there are $2^{\ell n \ell_{\mathrm{out}}(n)}$ such functions. Also note that there exist $\leq 2^\ell$ functions capable of being represented by ℓ bits. Thus, the probability that a random function from $\mathsf{Func}_{\ell_{\mathrm{out}}(n)}$ can be represented in ℓ bits is $\leq 2^\ell / 2^{\ell n \ell_{\mathrm{out}}(n)} = \mathsf{negl}(n)$.

Thus, with all but negligible probability over the choices of the random oracle, programs C and C' are equivalent. Therefore, if there is a difference in advantage, we can create an algorithm $\mathcal{B}$ that breaks the security of indistinguishability obfuscation.

$\mathcal{B}$ runs as the challenger in the $\mathsf{IND\text{-}CPA}$ game. When it is time to create the obfuscated program it submits both programs $C_0 = C$ and $C_1 = C'$ to an indistinguishability obfuscation challenger. If the challenger chooses the first then we are in $\mathbf{H_0}$; if it chooses the second then we are in $\mathbf{H_1}$. Thus, any adversary with non-negligible advantages in the two hybrids leads to $\mathcal{B}$ as an attacker on the security of the indistinguishability obfuscation scheme. □

We now show that an adversary who can successfully attack hybrid $\mathbf{H_1}$ can be used to construct an adversary attacking the underlying $\mathsf{IND\text{-}CPA}$ scheme.

Claim. $\Pr\left[\mathsf{PubK}_{\mathcal{A},\mathbf{H_1}}(n) = 1\right] \leq \Pr\left[\mathsf{PubK}_{\mathcal{B},\Pi'}(n) = 1\right]$ *where* $\mathcal{A}$ *is the adversary in* $\mathbf{H_1}$ *and* $\mathcal{B}$ *is the* $\mathsf{IND\text{-}CPA}$ *adversary against the underlying encryption scheme* Π'*.*

Proof. We construct the adversary $\mathcal{B}$ as follows. The adversary $\mathcal{B}$ internally simulates $\mathcal{A}$. When $\mathcal{B}$ receives pk', it generates $\mathsf{Obf}(C')$ as in $\mathbf{H_1}$ and provides $\mathsf{pk} := (\mathsf{pk}', \mathsf{Obf}(C'))$ to $\mathcal{A}$. When $\mathcal{B}$ receives a challenge ciphertext c, it forwards c to $\mathcal{A}$. Finally, $\mathcal{B}$ outputs the bit b' output by $\mathcal{A}$.

Clearly, if $\mathcal{A}$ can win the $\mathbf{H_1}$ game with probability ϵ then $\mathcal{B}$ can win the $\mathsf{IND\text{-}CPA}$ game with at least ϵ. □

Together, these two claims show that $\Pr\left[\mathsf{PubK}_{\mathcal{A},\Pi}(n) = 1\right] \leq \Pr\left[\mathsf{PubK}_{\mathcal{B},\Pi'}(n) = 1\right]$ where $\mathcal{A}$ is the $\mathsf{IND\text{-}CPA}$ adversary against Π, and $\mathcal{B}$ is the $\mathsf{IND\text{-}CPA}$ adversary against the underlying encryption scheme Π'. Since the underlying Π' is $\mathsf{IND\text{-}CPA}$-secure, we have that $\Pr\left[\mathsf{PubK}_{\mathcal{B},\Pi'}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n)$. Therefore we obtain $\Pr\left[\mathsf{PubK}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n)$, which completes the proof. ■

Lemma 3.1.2. *For all* $\ell \in \mathsf{poly}(n)$*, there exists a public-key bit-encryption scheme secure in the random oracle model but insecure when implemented with any efficiently computable* (poly, ℓ)*-ensemble.*

Proof. Fix some $\ell \in \mathsf{poly}(n)$. We modify the scheme Π described above to use (poly, ℓ)-ensemble $\mathcal{F}$ to implement the random oracle, thus obtaining the scheme $\widetilde{\Pi} = (\widetilde{\mathsf{Gen}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Dec}})$:

- $\widetilde{\mathsf{Gen}}$: On input 1^n, choose $s \leftarrow\!\!\$\ \{0,1\}^n$. Run $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}^{f_s}(1^n)$, and output $((\mathsf{pk}, s), (\mathsf{sk}, s))$.
- $\widetilde{\mathsf{Enc}}$: Output $\mathsf{Enc}_{\mathsf{pk}}(b)$.
- $\widetilde{\mathsf{Dec}}$: Output $\mathsf{Dec}_{\mathsf{sk}}(c)$.

Now the seed s is part of the public key, and it is known to the adversary. Thus, the adversary can simply parse pk into $(\mathsf{pk}', \mathsf{Obf}(C))$, and provide as input to $\mathsf{Obf}(C)$ the description of Eval (recall that Eval is the algorithm such that $\mathsf{Eval}(s, x) = f_s(x)$ for all $s \in \{0,1\}^n$ and $x \in \{0,1\}^*$), thus learning sk. ■

4 Extensions

Our approach used in Section 3 can be applied to more than just bit-encryption. In this section we show how to extend our result to provide separations for protocols satisfying most "natural" simulation- or game-based definitions. In Section 4.1, we show how to adapt our random oracle separation for bit-encryption to work for a large class of protocols secure under simulation-based definitions. Likewise, in Section 4.2, we adapt our bit-encryption separation to work for many protocols secure under game-based definitions. Although the theorem statements below provide separations in the random oracle model, the same approach can be applied to other idealized models, given appropriate formalizations of these models. As an example, in Appendix A we demonstrate how to adapt the below results to the generic group model.

4.1 Separations for Simulation-based Definitions

Here we focus on the universal composability (UC) framework [13]; we believe the separation detailed below can be easily adapted to other simulation-based models. In what follows, we assume the reader is familiar with the simulation paradigm, and in particular the UC framework.

We consider well-formed functionalities [16]. We call an ideal functionality f *trivial* if it can be realized by a "strange" protocol π as described in the following:

Definition 4.1. *Let f be an ideal functionality in the UC framework, and let π be a protocol where, upon initialization, all parties broadcast their initial randomness and inputs. Then f is* $\mathsf{trivial}$ *if for all environments $\mathcal{E}$ and for all adversaries $\mathcal{A}$, there exists a simulator $\mathcal{S}$ such that*

$$\Pr[\text{EXEC}_{f,\mathcal{S},\mathcal{E}} = \text{EXEC}_{\pi,\mathcal{A},\mathcal{E}}] = 1.$$

We now prove the following.

Theorem 4.1. *Consider a non-trivial ideal functionality f in the UC framework, and let π be a protocol which UC-realizes f in the $\mathcal{F}$-hybrid world. Then for all choices of $\ell \in \mathsf{poly}(n)$, there exists some protocol π' which UC-realizes f in the $(\mathcal{F}, \mathcal{F}_{RO})$-hybrid world[2] but is not UC-realizable when instantiated with a (poly, ℓ)-ensemble.*

Proof. Fix some non-trivial ideal functionality f for some set of parties $\mathcal{P} = \{P_1, \ldots, P_m\}$, and let π be a t-round protocol which UC-realizes f. On protocol initialization, each party P_i is initialized with randomness r_i and given input x_i. Let $M_{i,j}^k$ denote the message sent from party P_i to party P_j in round k; without loss of generality, we assume that for all parties P_i and P_j and for all rounds $1 \leq k \leq t$, message $M_{i,j}^k$ exists[3].

Now fix some $\ell \in \mathsf{poly}(n)$. We construct a protocol π' as follows. Protocol π' runs exactly as π except for the first round of the protocol. In this round, each party P_i proceeds as follows. For $j \in \{1, \ldots, \ell n\}$, P_i chooses $z_j \leftarrow\!\!\$\, \{0,1\}^n$ and computes $y_j \leftarrow \mathcal{O}(z_j)$. Then, based on input x_i, randomness r_i, as well as $\{z_j, y_j\}_j$, party P_i creates an obfuscation of the program C_i as defined in Figure 3 and sends $\mathsf{Obf}(C_i)$ over the standard channel, in addition to sending message $M_{i,j}^1$ as normal (i.e., this message may be sent using some hybrid functionality).

[2] $\mathcal{F}_{\mathsf{RO}}$ is defined in Appendix B.

[3] This is without loss of generality because $M_{i,j}^k$ can always be the empty message.

Constants: $z_1, \ldots, z_{\ell n}$, $y_1, \ldots, y_{\ell n}$, r_i, x_i.
Input: a description $\langle h \rangle$ of a function h.

1. For $i \in \{1, \ldots, \ell n\}$, compute $\widehat{y}_i := h(z_i)$.
2. If for $i \in \{1, \ldots, \ell n\}$ it holds that $\widehat{y}_i = y_i$, then output r_i and x_i; otherwise, output $\perp$.

Figure 3: Program C_i.

Lemma 4.1.1. *Assume that Obf is an indistinguishability obfuscator. Then for any choice of $\ell \in \mathsf{poly}(n)$ the construction π' UC-realizes f in the $(\mathcal{F}, \mathcal{F}_{\mathsf{RO}})$-hybrid world.*

Proof (Sketch). This follows directly from the fact that with high probability there is no "small representation" of a random oracle, and the argument is very similar to that shown in Lemma 3.1.1. We thus only give the high-level idea below.

Let $\mathcal{A}'$ be an adversary attacking protocol π'; we construct a simulator $\mathcal{S}'$ as follows. The simulator $\mathcal{S}'$ simply runs the simulator $\mathcal{S}$ for protocol π and outputs whatever $\mathcal{S}$ outputs. Intuitively, the output of $\mathcal{S}'$ is indistinguishable from that of $\mathcal{A}'$ because π' is exactly the same as π except for the sending of $\mathsf{Obf}(C_i)$ by party P_i. However, with high probability over the choices of the random oracle (cf. Lemma 3.1.1), this obfuscation is identical to the obfuscation of the zero circuit, and thus $\mathcal{A}'$ gains no advantage from this additional information. □

Lemma 4.1.2. *Assume that Obf is an indistinguishability obfuscator. Then for any choice of $\ell \in \mathsf{poly}(n)$ the construction π' is completely insecure in the $\mathcal{F}$-hybrid world (i.e., when the random oracle is instantiated by any efficiently computable (poly, ℓ)-ensemble).*

Proof (Sketch). Let $\mathcal{A}$ be an adversary attacking π'. The adversary $\mathcal{A}$ reads the messages sent by all parties, and thus receives $\mathsf{Obf}(C_i)$ from all parties P_i (recall that $\mathsf{Obf}(C_i)$ is sent over the standard channel). Thus, $\mathcal{A}$ can extract P_i's initial randomness and input by providing the instantiation of the random oracle as input to $\mathsf{Obf}(C_i)$, and can thus reproduce the internal state and inputs of all parties.

Now suppose towards a contradiction that π' UC-realizes f. This implies that there exists some adversary $\mathcal{S}$ which when interacting with f produces a similar transcript to that produced by $\mathcal{A}$; namely, $\mathcal{S}$ must be able to reproduce the internal state and inputs of all honest parties given only access to f. However, this implies that f is trivial, a contradiction. □

This completes the proof. ■

It is an easy corollary to see that Theorem 4.1 can be adapted to other idealized models besides the random oracle model, such as the generic group model, the random permutation model, etc. Thus, assuming indistinguishability obfuscation in the standard model, we are able to show idealized model separations for most protocols secure under simulation-based definitions.

4.2 Separations for Game-based Definitions

We first give a general framework for what we mean by a "game-based" definition. We consider only single-stage games, where an adversary $\mathcal{A}$ interacts with some challenger $\mathcal{C}$. A *game-based definition* $\mathcal{G}$ is defined by a tuple $(\mathcal{C}, \mathcal{O}_1, \ldots, \mathcal{O}_k, \mathcal{O}_{k+1}, \ldots \mathcal{O}_m, k, f, T)$, where $\mathcal{C}$ denotes a PPT algorithm (i.e., the challenger's code), $\mathcal{O}_1, \ldots, \mathcal{O}_k$ denote oracles available to both $\mathcal{A}$ and $\mathcal{C}$, $\mathcal{O}_{k+1}, \ldots, \mathcal{O}_m$ denote

oracles available to *only* $\mathcal{C}$, f denotes a predicate function, and T denotes a threshold function. Each oracle $\mathcal{O}_i$ outputs tuples of strings. The randomness of all the oracles are initialized by $\mathcal{C}$. A scheme/protocol Π *implements* $\mathcal{G}$ if it implements the oracles $\mathcal{O}_1, \ldots, \mathcal{O}_m$.

For definition $\mathcal{G}$ and scheme Π which implements $\mathcal{G}$, let $z \leftarrow \mathcal{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_k}$ denote the output of the adversary after interacting with $\mathcal{C}$, where all the oracle calls are "routed through" $\mathcal{C}$. That is, each oracle available to $\mathcal{A}$ is first initialized by $\mathcal{C}$, where the initialization fixes both the oracle's randomness and (optionally) some of the oracle's inputs; all queries by $\mathcal{A}$ to oracle $\mathcal{O}_i$ go through this (fixed) oracle. For example, if $\mathcal{O}_i$ is an encryption oracle, $\mathcal{C}$ fixes both the initial randomness as well as the public key; any queries by $\mathcal{A}$ will thus be encrypted under the fixed public key using the fixed initial randomness. The predicate f takes as input the initial randomness of $\mathcal{C}$ and the output of $\mathcal{A}$, and outputs a bit.

We define $\mathcal{A}$'s success probability against scheme Π in $\mathcal{G}$ as

$$\mathsf{Succ}^{\mathcal{A}}[\mathcal{G}, \Pi] \stackrel{\text{def}}{=} \Pr_{r,r_1,\ldots,r_k}\left[z \leftarrow \mathcal{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_k} : f(r, z) = 1\right].$$

That is, $\mathcal{A}$'s success probability is the probability it can make the predicate f output 1, where the probability is over the choices of $\mathcal{C}$'s and the oracles' randomness. We say that a scheme Π *securely implements* $\mathcal{G}$, or is *secure*, if it holds that $\mathsf{Succ}^{\mathcal{A}}[\mathcal{G}, \Pi] \leq T(n) + \mathsf{negl}(n)$. A scheme Π *insecurely implements* $\mathcal{G}$, or is *insecure*, if it holds that $\mathsf{Succ}^{\mathcal{A}}[\mathcal{G}, \Pi] \geq T(n) + \epsilon(n)$ for some non-negligible function $\epsilon(n)$.

As an example, consider the definition for bit-encryption as presented in Section 3. This is captured in our framework as follows. We define three oracles, $\mathcal{O}_1 = \mathsf{Enc}$, $\mathcal{O}_2 = \mathsf{Gen}$, and $\mathcal{O}_3 = \mathsf{Dec}$, corresponding to the three algorithms required for bit-encryption. Since $\mathcal{A}$ only has access to the encryption oracle, we set $k = 1$. The challenger $\mathcal{C}$ is defined as in Section 3. The predicate $f(r, z)$ runs $\mathcal{C}(r)$ until $\mathcal{C}$ computes b, and outputs whether or not b equals z (where z is the value output by $\mathcal{A}$). The threshold function is set to $T(n) = 1/2$.

We call a game-based definition $\mathcal{G}$ *trivially secure* if for all secure schemes Π it holds that

$$\mathsf{Succ}^{\mathcal{A}}[\mathcal{G}, \Pi] = \Pr_{r,r_1,\ldots,r_k}\left[z \leftarrow \mathcal{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_k}(r) : f(r, z) = 1\right].$$

That is, a definition is trivially secure if a scheme satisfying the definition is as secure as the setting where the adversary is given all the initial randomness to $\mathcal{C}$. As an example, note that bit encryption is *not* trivially secure, as if $\mathcal{A}$ was given the randomness r of $\mathcal{C}$, it could simply run $\mathcal{C}$ internally and extract the secret key sk, thus succeeding with probability 1, whereas without r we have that $\mathcal{A}$ succeeds with probability $1/2 + \mathsf{negl}(n)$ (assuming some underlying hard problem, of course). However, consider a game where $\mathcal{C}$ chooses a random x, computes $y := H(x)$ for some cryptographic hash function H, and sends y to $\mathcal{A}$; security holds if $\mathcal{A}$ cannot find an $x' \neq x$ such that $H(x') = y$. In this setting, whether $\mathcal{A}$ has x or not does not necessarily help it break security, and thus this definition may be trivially secure for certain instantiations of H.

Note that we can easily integrate idealized models, such as the random oracle model, into this framework by including an additional oracle which implements the desired idealized functionality to both $\mathcal{A}$ and $\mathcal{C}$.

Now we want to show that for all game-based definitions $\mathcal{G}$, for all protocols Π which securely implement $\mathcal{G}$ in the random oracle model, and for all choices of $\ell \in \mathsf{poly}(n)$, there exists some protocol Π' secure in the random oracle model but insecure in the standard model when instantiated with a (poly, ℓ)-ensemble.

Constants: $x_1, \ldots, x_{\ell n}$, $y_1, \ldots, y_{\ell n}$, r.
Input: a description $\langle h \rangle$ of a function h.

1. For $i \in \{1, \ldots, \ell n\}$, compute $\widehat{y_i} := h(x_i)$.
2. If for $i \in \{1, \ldots, \ell n\}$ it holds that $\widehat{y_i} = y_i$, then output r; otherwise, output $\perp$.

Figure 4: Program C.

However, it turns out that the notion of game-based definitions defined above is too strong to prove this result. This is because we place no restrictions on the challenger $\mathcal{C}$. As an example, consider a modified bit-encryption game where the challenger acts exactly as before, except it refuses to send any bits to $\mathcal{A}$ that "look like" an obfuscated circuit. This simple modification to the challenger prevents our attack from working for particular implementations of Obf, e.g., ones that prepend each obfuscated circuit with the string "this is an obfuscated circuit".

We thus consider a restriction on the above framework, and in particular, a restriction on the actions of $\mathcal{C}$. Consider a challenger which, on input randomness r, runs with oracle access to $\mathcal{O}_1, \ldots, \mathcal{O}_m$ as before. When $\mathcal{C}$ queries an oracle, it receives back a tuple $(s_1, \ldots)$. We call a challenger *weakened* if all messages sent to $\mathcal{A}$ are values within the tuples output by the oracle queries. For example, if $\mathcal{C}$ queries an oracle which implements key generation for some public-key cryptosystem, it receives back the tuple $(\mathsf{pk}, \mathsf{sk})$. If the challenger is weakened, it can send pk, sk, both or neither to $\mathcal{A}$, but it cannot send $f(\mathsf{pk})$ for some arbitrary function f, and likewise it cannot send some value x not output by an oracle. Note that most game-based definitions use this weakened challenger notion.

We call $\mathcal{G}$ a *weak game-based definition* if it is a game-based definition as defined above, except with the requirement that $\mathcal{C}$ be a weakened challenger. We are now ready to prove the following theorem.

Theorem 4.2. *Consider a non-trivially secure weak game-based definition $\mathcal{G}$, and let Π be a protocol which securely implements $\mathcal{G}$. Then for all choices of $\ell \in \mathsf{poly}(n)$, there exists some protocol Π' secure in the random oracle model but insecure when instantiated with a (poly, ℓ)-ensemble.*

Proof. Fix some non-trivially secure weak game-based definition $\mathcal{G}$, and let Π be a protocol which securely implements $\mathcal{G}$ (Π need not be in the random oracle model). Fix some $\ell \in \mathsf{poly}(n)$. We construct a protocol Π' as follows. Protocol Π' runs exactly as Π except for the first message sent from $\mathcal{C}$ to $\mathcal{A}$. Let M be this message. In protocol Π', $\mathcal{C}$ proceeds as follows. Let r be the initial randomness of $\mathcal{C}$. For $i \in \{1, \ldots, \ell n\}$, $\mathcal{C}$ chooses $x_i \leftarrow\!\!\$\, \{0,1\}^n$ and computes $y_i \leftarrow \mathcal{O}(x_i)$. Then, $\mathcal{C}$ creates an obfuscation $\mathsf{Obf}(C)$ of the program C defined in Figure 4 and sends $\widehat{M}$ to $\mathcal{A}$, where $\widehat{M} = (M, \mathsf{Obf}(C))$.

Lemma 4.2.1. *Assume that Obf is an indistinguishability obfuscator. Then for any choice of $\ell \in \mathsf{poly}(n)$ the construction Π' securely implements $\mathcal{G}$ in the random oracle model.*

Proof (Sketch). This follows exactly as in Lemma 4.1.1. □

Lemma 4.2.2. *Assume that Obf is an indistinguishability obfuscator. Then for any choice of $\ell \in \mathsf{poly}(n)$ the construction Π' is insecure when the random oracle is instantiated by any efficiently computable (poly, ℓ)-ensemble.*

Proof (Sketch). We apply the same idea as in Lemma 4.1.2. Let $\mathcal{A}$ be the adversary. Upon receiving the first message from $\mathcal{C}$, $\mathcal{A}$ can extract $\mathcal{C}$'s initial randomness r and thus reproduce the internal state of $\mathcal{C}$. By our assumption that $\mathcal{G}$ is not trivially secure, Π' is thus insecure. □

This completes the proof. ■

Note that as in the simulation-based case, we can easily adapt Theorem 4.2 to other idealized models and thus achieve idealized model separations for most game-based protocols, assuming indistinguishability obfuscation in the standard model.

Acknowledgments

The authors would like to thank Brent Waters and Susan Hohenberger for helpful conversations during the course of this work.

The work of Jonathan Katz was supported in part by NSF award #1223623. The work of Alex J. Malozemoff was conducted with Government support through the National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFG 168a, awarded by DoD, Air Force Office of Scientific Research.

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.

[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, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 221–238, Copenhagen, Denmark, May 11–15, 2014. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2013/631.

[3] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2), 2012. Full version available at https://eprint.iacr.org/2001/069.

[4] M. Bellare, A. Boldyreva, and A. Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 171–188, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2003/077.

[5] M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 398–415, Santa Barbara, CA, USA, Aug. 18–22, 2013. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2013/424.

[6] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, Nov. 3–5, 1993. ACM Press. Full version available at http://cseweb.ucsd.edu/~mihir/papers/ro.html.

[7] N. Bitansky, R. Canetti, O. Paneth, and A. Rosen. Indistinguishability obfuscation vs. auxiliary-input extractable functions: One must fall. Cryptology ePrint Archive, Report 2013/641, 2013. http://eprint.iacr.org/2013/641.

[8] N. Bitansky, R. Canetti, O. Paneth, and A. Rosen. More on the impossibility of virtual-black-box obfuscation with auxiliary input. Cryptology ePrint Archive, Report 2013/701, 2013. http://eprint.iacr.org/2013/701.

[9] E. Boyle, K.-M. Chung, and R. Pass. On extractability obfuscation. In Y. Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 52–73, San Diego, CA, USA, Feb. 24–26, 2014. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2013/650.

[10] Z. Brakerski and G. N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Y. Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 1–25, San Diego, CA, USA, Feb. 24–26, 2014. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2013/563.

[11] 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, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 188–205, Santa Barbara, CA, USA, Aug. 17–21, 2014. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2014/099.

[12] C. Brzuska, P. Farshim, and A. Mittelbach. Random oracle uninstantiability from indistinguishability obfuscation. Cryptology ePrint Archive, Report 2014/867, 2014. https://eprint.iacr.org/2014/867.

[13] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science*, pages 136–145, Las Vegas, Nevada, USA, Oct. 14–17, 2001. IEEE Computer Society Press. Full version available at https://eprint.iacr.org/2000/067.

[14] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998. ACM Press. Full version available at https://eprint.iacr.org/1998/011.

[15] R. Canetti, O. Goldreich, and S. Halevi. On the random-oracle methodology as applied to length-restricted signature schemes. In M. Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 40–57, Cambridge, MA, USA, Feb. 19–21, 2004. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2003/150.

[16] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *34th Annual ACM Symposium on Theory of Computing*, pages 494–503, Montréal, Québec, Canada, May 19–21, 2002. ACM Press. Full version available at https://eprint.iacr.org/2002/140.

[17] J.-S. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 446–464, Cambridge, UK, Apr. 15–19, 2012. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2011/440.

[18] A. De Caro, V. Iovino, A. Jain, A. O'Neill, O. Paneth, and G. Persiano. On the achievability of simulation-based security for functional encryption. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 519–535, Santa Barbara, CA, USA, Aug. 18–22, 2013. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2013/364.

[19] A. W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Y. Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 100–109, Queenstown, New Zealand, Dec. 1–5, 2002. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2002/086.

[20] A. W. Dent. Fundamental problems in provable security and cryptography. *Philosophical Transactions of the Royal Society A*, 364:3215–3230, 2006. Full version available at https://eprint.iacr.org/2006/278.

[21] Y. Dodis, R. Oliveira, and K. Pietrzak. On the generic insecurity of the full domain hash. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 449–466, Santa Barbara, CA, USA, Aug. 14–18, 2005. Springer, Berlin, Germany.

[22] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49, Berkeley, CA, USA, Oct. 26–29, 2013. IEEE Computer Society Press. Full version available at https://eprint.iacr.org/2013/601.

[23] 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, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 518–535, Santa Barbara, CA, USA, Aug. 17–21, 2014. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2013/860.

[24] C. Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, 2008.

[25] S. Goldwasser and Y. T. Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th Annual Symposium on Foundations of Computer Science*, pages 102–115, Cambridge, Massachusetts, USA, Oct. 11–14, 2003. IEEE Computer Society Press. Full version available at https://eprint.iacr.org/2003/034.

[26] D. Hofheinz, T. Jager, D. Khurana, A. Sahai, B. Waters, and M. Zhandry. How to generate and use universal parameters. Cryptology ePrint Archive, Report 2014/507, 2014. https://eprint.iacr.org/2014/507.

[27] D. Hofheinz, A. Kamath, V. Koppula, and B. Waters. Adaptively secure constrained pseudorandom functions. Cryptology ePrint Archive, Report 2014/720, 2014. https://eprint.iacr.org/2014/720.

[28] D. Hofheinz and J. Müller-Quade. Universally composable commitments using random oracles. In M. Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 58–76, Cambridge, MA, USA, Feb. 19–21, 2004. Springer, Berlin, Germany.

[29] E. Kiltz and K. Pietrzak. On the security of padding-based encryption schemes - or - why we cannot prove OAEP secure in the standard model. In A. Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 389–406, Cologne, Germany, Apr. 26–30, 2009. Springer, Berlin, Germany.

[30] G. Leurent and P. Q. Nguyen. How risky is the random-oracle model? In S. Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 445–464, Santa Barbara, CA, USA, Aug. 16–20, 2009. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2008/441.

[31] U. M. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In M. Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39, Cambridge, MA, USA, Feb. 19–21, 2004. Springer, Berlin, Germany. Full version available at https://eprint.iacr.org/2003/161.

[32] J. B. Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In M. Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126, Santa Barbara, CA, USA, Aug. 18–22, 2002. Springer, Berlin, Germany.

[33] V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer, Berlin, Germany.

A Adapting Theorem 4.1 and Theorem 4.2 to the Generic Group Model

To demonstrate how to adapt Theorem 4.1 and Theorem 4.2 to other idealized models, we provide here an adaptation to the generic group model. We first define the generic group model and how this model is instantiated using *encoding ensembles* [19] (which can be thought of as analogous to the *function ensembles* used for instantiating the random oracle model).

Generic group model. Let $\ell_{\text{out}} : \mathbb{N} \to \mathbb{N}$ be a length function with $\ell_{\text{out}}(n) \geq n$, and define the set $S = \{0,1\}^{\ell_{\text{out}}(n)}$. Let p be an n-bit prime. The generic group model is defined by two oracles,

$\mathcal{O}_{\mathsf{enc}}$ and $\mathcal{O}_{\mathsf{add}}$, available to all parties, where $\mathcal{O}_{\mathsf{enc}} : \mathbb{Z}_p \to S$ such that $\mathcal{O}_{\mathsf{enc}}(x) = \mathcal{O}_{\mathsf{enc}}(y)$ iff $x = y$ and $\mathcal{O}_{\mathsf{add}} : S \times S \times \mathbb{Z}_2 \to S$ such that $\mathcal{O}_{\mathsf{add}}(\mathcal{O}_{\mathsf{enc}}(x), \mathcal{O}_{\mathsf{enc}}(y), b) = \mathcal{O}_{\mathsf{enc}}(x + (-1)^b y)$.[4]

Encoding ensembles. Let $\ell_{\mathrm{out}} : \mathbb{N} \to \mathbb{N}$ be a length function with $\ell_{\mathrm{out}}(n) \geq n$. An ℓ_{out}-*encoding-ensemble* is a sequence $\mathcal{F} = \{F_n\}_{n \in \mathbb{N}}$ of families of functions $F_n = \{f_s : \mathbb{Z}_p \to \{0,1\}^{\ell_{\mathrm{out}}(n)}\}_{s \in \{0,1\}^n}$ such that the following conditions hold:

1. There exists a polynomial-time algorithm Eval such that for every $s \in \{0,1\}^n$ and $x \in \mathbb{Z}_p$ it holds that $\mathsf{Eval}(s, x) = f_s(x)$.
2. There exists a polynomial-time algorithm Add such that $\mathsf{Add}(s, f_s(x), f_s(y), b) = f_s(x + (-1)^b y)$.

As in the function ensemble case, let $\ell_{\mathrm{eval}}(n)$ be the length of the bitstring representation of Eval. Let a (poly, ℓ)-encoding-ensemble be a class of ℓ'-encoding-ensembles such that $\ell' \in \mathsf{poly}(n)$ (with the restriction that $\ell' \geq n$) and the bitstring representation of Eval is $\leq \ell$.

Let $\mathcal{F}_{\mathsf{GG}}$ denote the "natural" adaptation of the generic group model to the UC framework (see Figure 6). We can now prove the following theorem.

Theorem A.1. *Consider a non-trivial ideal functionality f in the UC framework, and let π be a protocol which UC-realizes f in the $\mathcal{F}$-hybrid world. Then for all choices of $\ell \in \mathsf{poly}(n)$, there exists some protocol π' which UC-realizes f in the $(\mathcal{F}, \mathcal{F}_{\mathsf{GG}})$-hybrid world but is not UC-realizable when instantiated with a (poly, ℓ)-encoding-ensemble.*

Proof. The proof structure follows exactly that shown in Theorem 4.1. The only difference is that instead of each party querying the random oracle when constructing the obfuscated circuit, they instead query $\mathcal{O}_{\mathsf{enc}}$. The proof follows immediately from the fact that with high probability there is no "small representation" of $\mathcal{O}_{\mathsf{enc}}$, whereas when $\mathcal{O}_{\mathsf{enc}}$ is instantiated with a concrete function, the adversary can easily extract the hidden information to break security. ■

The adaptation of Theorem 4.2 is similar, and thus we only present the theorem statement.

Theorem A.2. *Consider a non-trivially secure weak game-based definition $\mathcal{G}$, and let Π be a protocol which securely implements $\mathcal{G}$. Then for all choices of $\ell \in \mathsf{poly}(n)$, there exists some protocol Π' secure in the generic group model but insecure when instantiated with a (poly, ℓ)-encoding-ensemble.*

B Ideal Functionalities

We define $\mathcal{F}_{\mathsf{RO}}$ in Figure 5 and $\mathcal{F}_{\mathsf{GG}}$ in Figure 6, which represent ideal functionalities implementing a random oracle and a generic group, respectively. We note that the definition of $\mathcal{F}_{\mathsf{RO}}$ is adapted from the work of Hofheinz and Müller-Quade [28]. It is straightforward to design ideal functionalities for other idealized models.

[4] Note that we only need $\mathcal{O}_{\mathsf{enc}}$ to prove our separations results.

Functionality $\mathcal{F}_{\mathsf{RO}}$

Initialization: Security parameter 1^n, length function $\ell_{\text{out}} : \mathbb{N} \to \mathbb{N}$, list of parties $P_1, \ldots, P_m$, and adversary $\mathcal{A}$.

1. $\mathcal{F}_{\mathsf{RO}}$ maintains a table T which stores inputs and their associated outputs; if $x \in T$, let $T(x)$ denote the associated output.
2. On receiving value $x \in \{0,1\}^*$ from party P_i or $\mathcal{A}$, $\mathcal{F}_{\mathsf{RO}}$ does the following:
   - If $x \in T$, output $T(x)$ to the appropriate party.
   - Otherwise, choose $y \leftarrow\!\!\$\, \{0,1\}^{\ell_{\text{out}}(n)}$, add (x, y) to T, and output y to the appropriate party.

Figure 5: Functionality $\mathcal{F}_{\mathsf{RO}}$.

Functionality $\mathcal{F}_{\mathsf{GG}}$

Initialization: Security parameter 1^n, length function $\ell_{\text{out}} : \mathbb{N} \to \mathbb{N}$, n-bit prime p, set $S = \{0,1\}^{\ell_{\text{out}}(n)}$, list of parties $P_1, \ldots, P_m$, and adversary $\mathcal{A}$.

1. $\mathcal{F}_{\mathsf{GG}}$ maintains a table T which stores inputs and their associated outputs; if $x \in T$, let $T(x)$ denote the associated output.
2. On receiving tuple (enc, x) from party P_i or $\mathcal{A}$, $\mathcal{F}_{\mathsf{GG}}$ does the following:
   - If $x \notin \mathbb{Z}_p$, output $\perp$ to the appropriate party.
   - If $x \in T$, output $T(x)$ to the appropriate party.
   - Otherwise, choose $y \leftarrow\!\!\$\, S$ conditioned on y not appearing as an output in T, add (x, y) to T, and output y to the appropriate party.
3. On receiving tuple (add, y, y', b) from party P_i or $\mathcal{A}$, $\mathcal{F}_{\mathsf{GG}}$ does the following:
   - If $y \notin S$ or $y' \notin S$ or $b \notin \{0,1\}$, output $\perp$ to the appropriate party.
   - If there does not exist an x (resp., x') such that (x, y) (resp., (x', y')) is in T, output $\perp$ to the appropriate party.
   - Otherwise, compute $x'' = x + (-1)^b x'$, store x'' in T, and output the associated output y'' to the appropriate party.

Figure 6: Functionality $\mathcal{F}_{\mathsf{GG}}$.

Changelog

- Version 1.1 (October 26, 2014):
  - Added comparison to the work of Brzuska et al. [12] in Section 1.1.
  - Changed text in Appendix B.
  - Updated Acknowledgments.
- Version 1.0 [20141022:202118] (October 20, 2014): First release.