

Reset Indifferentiability and its Consequences

Paul Baecher¹

Christina Brzuska²

Arno Mittelbach¹

¹Darmstadt University of Technology, Germany

www.cryptoplexity.de

²Tel-Aviv University, Israel

www.brzuska.de/christina

baecher@cs.tu-darmstadt.de

brzuska@post.tau.ac.il

arno.mittelbach@cased.de

Abstract. The equivalence of the random oracle model and the ideal cipher model has been studied in a long series of results. Holenstein, Künzler, and Tessaro (STOC, 2011) have recently completed the picture positively, assuming that, roughly speaking, equivalence is indifferentiability from each other. However, under the stronger notion of reset indifferentiability this picture changes significantly, as Demay et al. (EUROCRYPT, 2013) and Luykx et al. (ePrint, 2012) demonstrate.

We complement these latter works in several ways. First, we show that any simulator satisfying the reset indifferentiability notion must be stateless and pseudo-deterministic. Using this characterization we show that, with respect to reset indifferentiability, two ideal models are either equivalent or incomparable, that is, a model cannot be strictly stronger than the other model. In the case of the random oracle model and the ideal cipher model, this implies that the two are incomparable. Finally, we examine weaker notions of reset indifferentiability that, while not being able to allow composition in general, allow composition for a large class of multi-stage games. Here we show that the seemingly much weaker notion of 1-reset indifferentiability proposed by Luykx et al. is equivalent to reset indifferentiability. Hence, the impossibility of coming up with a reset indifferentiable construction transfers to the setting where only one reset is permitted, thereby re-opening the quest for an achievable and meaningful notion in between the two variants.

Contents

1	Introduction	3
2	Preliminaries	7
2.1	Indifferentiability	8
3	Pseudo-deterministic Stateless Simulators for Indifferentiability	9
3.1	Multi-stage Indifferentiability	10
3.2	Pseudo-Deterministic Algorithms	11
4	The Random Oracle and Ideal Cipher Model are Incomparable	13
4.1	The Duality Lemma for Multi-Stage Indifferentiability	14
5	Single versus Multi-Reset	15
6	Acknowledgements	19
A	Missing Proofs for Section 3	22
A.1	Proof of Lemma 3.5	22
B	Pseudo-Deterministic vs. Deterministic Simulators	24

1 Introduction

IDEALIZED MODELS. The standard approach to cryptographic security is to reduce the security of a scheme to a (hopefully) well-studied algebraic or combinatorial complexity assumption. Unfortunately, a large number of cryptographic schemes does not admit a security reduction in the standard model. In these cases, the community often resorts to an idealized model, where we can sometimes obtain a proof of security. It is, of course, highly controversial whether or not proofs in idealized models are acceptable, but there is a tendency to prefer an analysis in an idealized model over the utter absence of any proof at all—in particular, when one is concerned with schemes that are widely deployed in practice [BR94, BR96, BRSS10].

Arguably the most popular model of this kind is the random oracle model (ROM) where all parties have oracle access to a public, randomly chosen function [BR93]. Somewhat related is the ideal cipher model (ICM) which gives all parties oracle access to a public, randomly chosen (keyed) blockcipher [Sha49]. Knowing that there is a close relation between pseudorandom functions and pseudorandom permutations—namely existential equivalence—one could suspect that the random oracle model and the ideal cipher model are equivalent, too. However, formalizing the notion of equivalence is delicate and so are the proofs.

EQUIVALENCE OF THE ROM AND ICM UNDER INDIFFERENTIABILITY. Maurer, Renner and Holenstein [MRH04] introduced the concept of indifferenciability, which since then has been regarded as the prevalent and actually only notion of equivalence between ideal primitives. A construction G^π with access to some primitive π is called indifferenciability from another ideal primitive Π , if there is a simulator \mathcal{S} such that the construction G^π implements an oracle that is indistinguishable from Π , even if the distinguisher \mathcal{D} additionally gets access to π . Now, demanding the distinguisher \mathcal{D} to distinguish (G^π, π) from Π is of little sense. Additional to the oracle Π , the distinguisher gets access to the simulator \mathcal{S} which tries to emulate π 's behavior consistently with Π . Thus, the distinguisher tries to distinguish the pair of oracles (G^π, π) from the pair of oracles (Π, \mathcal{S}^Π) .

In the case of the ideal cipher model and the random oracle model, considerable effort has led to a proof of equivalence [CDMP05, CPS08, HKT11] under indifferenciability. The reason why indifferenciability was considered a suitable notion of equivalence is the appealing composition theorem established by Maurer et al. [MRH04]. Namely, they transform any reductionist argument in the presence of the ideal primitive Π into a proof that relies on the existence of π only. Their theorem thus transforms a reduction \mathcal{R} into a reduction \mathcal{R}' , where the latter locally implements a single copy of the simulator \mathcal{S} . Jumping ahead, it will turn out that in this step, they rely on an implicit assumption.

MULTI-STAGE ADVERSARIES. Ristenpart et al. [RSS11] were the first to point out scenarios where indifferenciability of G^π from Π was not sufficient to replace Π by G^π . Their counterexamples involve adversaries that run in multiple stages, i.e., an adversary \mathcal{A} con-

sists of two or more sub-adversaries, say $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, that do not share state (or at least not arbitrary state). Now, a reduction \mathcal{R} that reduces to such a *multi-stage game* also needs to be split into two parts $(\mathcal{R}_1, \mathcal{R}_2)$ where the same restrictions upon the sharing of state applies. Hence, for the composition theorem by Maurer et al., each part of the reduction \mathcal{R}_1 and \mathcal{R}_2 needs to implement its own, independent copy of the simulator \mathcal{S} . However, in this case, the two copies of the simulator will not necessarily behave in the same way as opposed to the “real” primitive π which is, roughly, what makes the composition theorem collapse in the setting of multi-stage games.

Curiously, their composition holds in the presence of *strong*, colluding adversaries, while it does not in the setting of weaker, non-colluding ones. Usually in cryptography, a conservative approach corresponds to considering the strongest possible adversary, as a primitive that is secure against a strong adversary is also secure against a weaker adversary. However, the indistinguishability composition theorem is not, by itself, a security model or a proof of security. Instead, it is a tool to transform any proof in a security model in the presence of one ideal primitive into a security proof in the same security model in the presence of another ideal primitive. Hence, one tries to cover any type of security model, which, in particular, includes security models where stage-sharing adversaries can mount trivial attacks. And thus, a conservative approach in the setting of indistinguishability demands including also weaker, namely non-colluding, state-sharing adversaries. Technically, the composition theorem is *harder* to prove for *weaker* adversaries, because it transforms an adversary of one type into another adversary of the same type. Considering a stronger adversary corresponds to a stronger assumption in the theorem, but also to a harder statement to prove, and vice versa for weaker adversaries.

One might hope that the distinction is of technical interest only. Unfortunately, as we argue, in basically all real-life scenarios, we need to consider multi-stage adversaries. Ristenpart et al. give several examples of multi-stage games for notions such as deterministic encryption [BBO07, BBN⁺09], key-dependent message security [BRS03], related key attacks [BK03], and non-malleable hash functions [BCFW09]. On the other hand, many classical notions of security seem inherently single-stage: IND-CPA or IND-CCA security for encryption, or signature schemes which are existentially unforgeable under (adaptive) chosen message attacks. However, any classical definition of security becomes multi-staged if it is augmented with a leakage oracle. The reason is that, in the random oracle model, every party should have access to the random oracle. In particular, this includes the leakage oracle and the adversarially specified leakage function, resulting in an implicit second stage [DKW11]. Hence, whenever side-channel attacks are reflected in a model, adversaries act at least in two-stages—and for real-life applications, we cannot discard side-channel attacks.

In order to cope with the new challenge of multi-stage adversaries, Ristenpart et al. put forward a strengthened notion called *reset indistinguishability*. Roughly speaking, in this game, the distinguisher may reset the simulator’s internal state between any two queries. Returning to ROM/ICM equivalence, an inspection of the simulators defined in [CDMP05] and [HKT11] (as well as [CPS08], for that matter) reveals that their behavior varies sub-

stantially with their state and, thus, they are not reset indifferntiable.

EQUIVALENCE OF THE ROM AND ICM UNDER RESET INDIFFERENTIABILITY. As plain indifferntiability is not sufficient to argue that two primitives are equivalent, the question regarding the ideal cipher model and the random oracle model is, thus, again open. Recently, it has been shown that reset indifferntiable constructions cannot be built via domain extension [DGHM13, LAMP12], thereby ruling out constructions from ideal ciphers that are reset indifferntiable from a random oracle; note that random oracles are usually perceived as having an infinite domain while ideal ciphers have a finite domain. With this result at hand, we thus know that ideal ciphers cannot be used to obtain random oracles via a reset indifferntiable construction, but it might still be possible to construct an ideal cipher from a random oracle, i.e., either the two models are entirely incomparable, or the random oracle model is strictly stronger.

We rule out such a possibility. Our so-called duality lemma establishes that if there is no construction G_1^π that is reset indifferntiable from primitive Π , then also vice versa, there is no construction G_2^Π that is reset indifferntiable from primitive π . Hence, our theorem complements the results by Demay et al. and Luykx et al. [DGHM13, LAMP12] showing that there can also not be a domain-shrinking construction.

Proving that according to plain indifferntiability, the ICM and ROM are equivalent had been a serious challenge and finally involved a Feistel-network with many rounds. A Feistel network is a domain-doubling construction, and is thus ruled out by the previous impossibility results. The few leverages that remain to bypass the current impossibility results possibly require quite new techniques. Firstly, it might still be possible to build a construction that is neither domain-shrinking, nor domain-extending. However, as we will see later, that means settling either direction (RO from IC and vice versa) simultaneously, and this might be quite challenging. The second leverage is a distinction that has been irrelevant in most works in the area of indifferntiability so far and that we would like to point out. Namely, strong indifferntiability requires the simulator \mathcal{S} to work for any distinguisher \mathcal{D} , while weak indifferntiability only demands that for every \mathcal{D} , there exists a good simulator \mathcal{S} . Known constructions are usually strongly indifferntiable, while most existing impossibility results rule out even weakly indifferntiable constructions. In contrast, we do not rule out weakly indifferntiable constructions. It would be interesting to see techniques that make non-black-box use of the distinguisher \mathcal{D} and establish a reset indifferntiable construction that is domain shrinking.

NOTIONS BETWEEN INDIFFERENTIABILITY AND RESET INDIFFERENTIABILITY. From the current state-of-the-art, there are two ways to proceed: firstly, we can develop new techniques to exploit the few remaining leverages left to bypass the existing impossibility results. Secondly, we might weaken the notion of reset indifferntiability as introduced by Ristenpart et al., to a notion that is achievable by constructions and which is sufficient for a sub-class of multi-stage games.

Demay et al. [DGHM13] introduce resource-restricted indifferntiability where adver-

saries may share a limited amount of state. If a certain amount s of shared state is allowed, then their impossibility result shows that a reset indiffereniable construction cannot extend the domain by more than $s + \lceil \log(s) \rceil$ bits. Maybe, the additional bits allow to bypass the impossibility results more easily, as proving domain extension by a few bits might be easier than requiring equality of the domain sizes—however, in this setting, the composition results accounts for a certain class of games only.

Another approach that has been put forward by Luykx et al. [LAMP12] is to reduce the number of resets. Indeed, allowing for a polynomial of resets/stages seems to be an overkill, as some games such as the security model for deterministic encryption [BBO07, BBN⁺09] and also certain forms of leakage require a constant number of adversarial stages only. To this end, Luykx et al. propose the notion of *single reset indiffereniable* where a distinguisher can make a single reset call only; naturally, a construction that is single-stage reset indiffereniable would be sufficient in any security game consisting of exactly two distinct adversarial stages such as deterministic encryption. Analogously, one can define n -reset indiffereniable for $n + 1$ adversarial stages.

However, as we prove, single-reset indiffereniable is already equivalent to full-reset indiffereniable and so are all notions of n -reset indiffereniable. Hence, reducing the number of allowed reset queries does not help us to establish composition results for a restricted class of games. Thus, if a general indiffereniable result is indeed impossible, then it is a curious open question how to cope with the uncomfortable situation. It might be possible to establish indiffereniable results and composition theorems for a class of games that is restricted in another way than by the number of queries. Indeed, it would be interesting to see how such a class could look like and whether there are games for which, in general, finding a suitable, indiffereniable construction is impossible.

SUMMARY OF OUR CONTRIBUTIONS. We first introduce the notion of *pseudo-deterministic* algorithms, which captures, that a probabilistic algorithm almost always returns the same answer on the same queries and thus shares many properties with deterministic algorithms. Essentially, a probabilistic (and possibly stateful) algorithm \mathcal{A} is called pseudo-deterministic, if no efficient distinguisher with black-box access to \mathcal{A} can make \mathcal{A} return two different answers on the same input. This notion of pseudo-determinism can be seen as an average-case version of the pseudo-deterministic algorithms that were recently introduced by Goldreich, Goldwasser, and Ron [GGR12]. While they require probabilism to be hard to detect on any input, we only require indistinguishability for efficiently generatable inputs, on the average. As stressed by Goldreich et al. [GGR12], pseudo-deterministic algorithms are practically as useful as deterministic algorithms, but they are also easier to construct—which we indeed exploit in our paper.

We will show in Section 3 that simulators for reset indiffereniable need to be stateless and pseudo-deterministic. Simplifying pseudo-deterministic to deterministic for the moment (we discuss this in Appendix B), this allows us to establish what we call the duality lemma. Perhaps surprisingly, it states that, with respect to reset indiffereniable, two idealized models are either *equivalent* or *incomparable*. The reason is that a deterministic

and stateless simulator can act as a construction and vice versa. Consequently, in order to prove equivalence in terms of reset indifferntiability, this lemma makes it sufficient to prove the “easier” direction, whichever this might be. In turn, for impossibility results, one might use this as a tool to prove impossibility more easily. In fact, we use the duality lemma to establish that not only domain-extending constructions are impossible, but also domain-shrinking constructions (Section 4) thereby complementing the results of [DGHM13]. Note that the duality lemma covers strong indifferntiability, leaving non-black-box use of the distinguisher as a potential leverage to bypass this impossibility.

The recently proposed [RSS11] notion of single reset indifferntiability intends to define a notion of indifferntiability that is easier to achieve and simultaneously covers an interesting class of multi-stage games that has two adversary stages only. Interestingly, as we establish, restricting the number of resets does not yield a weaker notion of equivalence. We prove that single (and n) reset indifferntiability is equivalent to reset indifferntiability (Section 5). Maybe surprisingly, our proof does not rely on a hybrid argument; instead, we establish a tight reduction that merely reduces the distinguisher’s advantage by a factor of 2.

2 Preliminaries

For a natural number $n \in \mathbb{N}$ we denote by $\{0, 1\}^n$ the set of all bit strings of length n . By $\{0, 1\}^*$ we denote the set of all bit strings of finite length. As usual $|\mathcal{M}|$ denotes the cardinality of a set \mathcal{M} and logarithms are to base 2. For some probabilistic algorithm \mathcal{A} and input x we denote by $\mathcal{A}(x; R)$ the output of \mathcal{A} on x using randomness R . Throughout this paper we assume that λ is a security parameter (if not explicitly given then implicitly assumed) and that algorithms (resp., Turing machines) run in polynomial time with respect to λ .

In this paper we consider random oracles and ideal ciphers (defined below) which we will collectively refer to as *ideal primitives*. Although we present most of the results directly for ideal ciphers and random oracles, the following more general notion of ideal primitives allows us to generalize some of our results:

Definition 2.1. *An ideal primitive Π_λ is a distribution on functions indexed by the security parameter λ . For some algorithm \mathcal{A} , security parameter λ and ideal primitive Π_λ we say that \mathcal{A} has access to Π if \mathcal{A} has oracle access to a function f chosen from the distribution Π_λ .*

We simply write Π , i.e., omit the security parameter, if it is clear from the context.

Remark. We will usually encounter only single instances of an ideal primitive Π at a time. Unless stated otherwise, if multiple parties have access to Π , then we implicitly assume that the corresponding function f was chosen from the distribution Π using the same randomness for all parties, i.e., all parties have oracle access to the same function f .

RANDOM ORACLES AND IDEAL CIPHERS. A random oracle $(\mathcal{R}_{\ell,m})_\lambda$ is the uniform distribution on all functions mapping $\{0,1\}^\ell$ to $\{0,1\}^m$ with $\ell := \ell(\lambda)$ and $m := m(\lambda)$. An ideal cipher $(\mathcal{E}_{k,n})_\lambda$ is the uniform distribution on all keyed permutations of the form $\{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ with $k := k(\lambda)$ and $n := n(\lambda)$. That is, for a cipher in the support of $(\mathcal{E}_{k,n})_\lambda$ each key $\kappa \in \{0,1\}^k$ describes a random (independent) permutation $\mathcal{E}_{k,n}(\kappa, \cdot) : \{0,1\}^n \rightarrow \{0,1\}^n$. By abuse of notation, the term random oracle (resp., ideal cipher) also refers to a specific instance chosen from the respective distribution.

KEYED VS. UNKEYED CIPHERS. The idealized cipher model has either been considered as a public *unkeyed* permutation or as a public *keyed* permutation. We present our results in the keyed setting since we feel that the ideal cipher model is usually perceived in this way. However, we want to point out that the results are equally valid for the unkeyed setting because our proofs do not rely on the presence of a key.

Independently of this, one might be tempted to argue that the settings are interchangeable since we know, for example, constructions of a keyed permutation from an ideal public permutation (Even and Mansour, [EM97]). Note though, that in order to make this argument work, one needs to show that these constructions are reset indifferentiable. However, the construction by Even and Mansour is a domain extender where the key size is twice the message size and we rule out reset indifferentiability for such extending constructions in Section 4. We note that it is an interesting open problem whether or not such (reset) indifferentiable non-extending transformations exist.

2.1 Indifferentiability

Let us now recall the indifferentiability notion of Maurer et al. [MRH04] in the version by Coron et al. [CDMP05] who replace *random systems* by oracle Turing machines (resp., ideal primitives). Since we are concerned with different types of indifferentiability, we will sometimes use the term *plain* indifferentiability when referring to this original notion of indifferentiability.

Definition 2.2. *A Turing machine G with black-box access to an ideal primitive π is strongly indifferentiable from an ideal primitive Π if there exists a simulator \mathcal{S}^Π , such that for any distinguisher \mathcal{D} there exists negligible function negl , such that:*

$$\left| \text{Prob} \left[\mathcal{D}^{G^\pi, \pi}(1^\lambda) = 1 \right] - \text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}^\Pi}(1^\lambda) = 1 \right] \right| \leq \text{negl}(\lambda) \quad (1)$$

We say that the construction is weakly indifferentiable if for any \mathcal{D} there exists a simulator \mathcal{S} such that (1) holds.

We will use the term *real world* to denote that the distinguisher \mathcal{D} talks to the construction G^π and the primitive π , whereas in the *ideal world*, the distinguisher \mathcal{D} talks to the “target” primitive Π and simulator \mathcal{S}^Π . The goal of the distinguisher is to determine which of the two *pairs* of oracles he is talking to. Towards this goal, the distinguisher \mathcal{D} queries its two oracles, of which one is called the honest interface \mathfrak{h} which is either G^π (in

the real world) or Π (in the ideal world). The other oracle is called the adversarial interface \mathfrak{a} and corresponds to either π (real world) or \mathcal{S}^Π (ideal world) Thus, $(\mathfrak{h}, \mathfrak{a}) := (G^\pi, \pi)$ if distinguisher \mathcal{D} is in the real world and $(\mathfrak{h}, \mathfrak{a}) := (\Pi, \mathcal{S}^\Pi)$ if it is in the ideal world. The names \mathfrak{h} (honest) and \mathfrak{a} (adversarial) are in the style of [RSS11] and suggestive: an honest party uses a construction as the designer intended; an adversary could, however, use the underlying building blocks to gain an advantage.

RESET INDIFFERENTIABILITY. Ristenpart et al. show [RSS11] that, in general, we cannot only securely replace a primitive Π by a construction G^π from primitive π , if the construction is indiffereniable only. Instead, G^π needs to be (weakly) *reset* indiffereniable from Π which extends the original indiffereniable definition by giving the distinguisher the power to reset the simulator at arbitrary times:

Definition 2.3. *Let the setup be as in Definition 2.2. An oracle Turing machine G^π is called strongly (resp. weakly) reset indiffereniable from ideal primitive Π if the distinguisher \mathcal{D} can reset the simulator \mathcal{S} to its initial state arbitrarily many times during the respective experiment.*

For reset indiffereniable the adversarial interface \mathfrak{a} in the real world simply ignores reset queries. Reset indiffereniable now allows composition in arbitrary games and not only in single-stage games, as does the original indiffereniable notion [RSS11, MRH04].

3 Pseudo-deterministic Stateless Simulators for Indiffereniable

tiability

Recall that the composition theorem by Maurer et al. [MRH04] for plain indiffereniable holds for single-stage adversaries only. Their theorem says, that if (i) the construction G^π is indiffereniable from the ideal primitive Π and if (ii) there is a reduction \mathcal{R} that transforms a successful adversary \mathcal{A} against some notion of security into an adversary $\mathcal{R}^\mathcal{A}$ against a single-stage game in the presence of the ideal primitive Π , then also in the presence of the construction G^π there is a reduction \mathcal{R}' that transforms a successful adversary \mathcal{A} into an adversary $\mathcal{R}'^\mathcal{A}$ against the single-stage game.

In order to prove a general composition theorem, Ristenpart et al. [RSS11] strengthen the notion of indiffereniable to account for the different stages of the adversary. They introduce the notion of (weak) reset indiffereniable and prove that the aforementioned theorem works for arbitrary games, if the construction G^π is reset indiffereniable from the ideal primitive Π . In contrast to plain indiffereniable, here, the distinguisher gets extra powers, namely to reset the simulator at arbitrary times. Ristenpart et al. [RSS11] and Demay et al. [DGHM13] remark that reset indiffereniable is equivalent to plain indiffereniable with stateless simulators. Intuitively, this follows from the observation that the distinguisher in the reset indiffereniable game can simply reset the simulator after each query it asks. We believe that, albeit equivalent, stateless simulators are often easier to

handle than reset-resistant simulators and thus explicitly introduce indifferenciability with stateless simulators as *multi-stage indifferenciability* and then prove that it is equivalent to reset indifferenciability.

In Subsection 3.2, we prove that strong multi-stage indifferenciability implies that the simulators are also *pseudo-deterministic*, a notion that we put forward in this section. Relative to a random oracle or an ideal cipher, we show how to derandomize pseudo-deterministic simulators, if the simulators are allowed to depend on the number of queries made by the distinguisher.

3.1 Multi-stage Indifferenciability

A stateless interactive algorithm is an algorithm whose behavior is statistically independent from the call/answer history of the algorithm. We now prove that indifferenciability with stateless simulators is equivalent to reset indifferenciability.

Definition 3.1. *A construction G with black-box access to primitive π is strongly multi-stage indifferenciability from primitive Π if there exists a stateless probabilistic polynomial-time simulator \mathcal{S} (with access to Π), such that for any probabilistic polynomial-time distinguisher \mathcal{D} there exists negligible function negl such that:*

$$\left| \text{Prob} \left[\mathcal{D}^{G^\pi, \pi}(1^\lambda) = 1 \right] - \text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}^\Pi}(1^\lambda) = 1 \right] \right| \leq \text{negl}(\lambda) \quad (2)$$

We say that a construction G^π is weakly multi-stage indifferenciability from Π if for any probabilistic polynomial-time distinguisher \mathcal{D} there exists a stateless probabilistic polynomial-time simulator \mathcal{S} such that (2) holds.

Lemma 3.2. *A construction G with black-box access to primitive π is weakly (resp., strongly) multi-stage indifferenciability from primitive Π if and only if G is weakly (resp., strongly) reset indifferenciability from primitive Π .*

Proof. First note that any stateless simulator is, naturally, indifferent to resets and thus multi-stage indifferenciability implies reset indifferenciability. Moreover, strong reset indifferenciability implies strong multi-stage indifferenciability since the simulator for reset indifferenciability must work for any distinguisher, in particular for those which reset after each query. Hence this stateful simulator can be simply initialized and run by a stateless simulator (the stateless simulator does this for each query it receives).

We now prove the remaining relation, i.e., that weak reset indifferenciability implies weak multi-stage indifferenciability. Assume that reset indifferenciability holds and consider an arbitrary distinguisher \mathcal{D} in the multi-stage indifferenciability game. From this we construct a distinguisher \mathcal{D}' for the reset indifferenciability game which runs \mathcal{D} and sends a reset query to its adversarial \mathfrak{a} -interface after every \mathfrak{a} -query issued by \mathcal{D} . Let \mathcal{S}' be the simulator for \mathcal{D}' guaranteed to exist by reset indifferenciability. We construct a stateless simulator \mathcal{S} for multi-stage indifferenciability which simply runs (the stateful) \mathcal{S}' and resets its own state

after each query. Now the following equations hold for $b \in \{0, 1\}$:

$$\text{Prob} \left[\mathcal{D}'^{\Pi, \mathcal{S}'}(1^\lambda) = b \right] = \text{Prob} \left[\mathcal{D}'^{\Pi, \mathcal{S}}(1^\lambda) = b \right] = \text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}}(1^\lambda) = b \right].$$

Thus, if equation (2) holds for $(\mathcal{D}', \mathcal{S}')$, then it holds equally for $(\mathcal{D}, \mathcal{S})$. \square

3.2 Pseudo-Deterministic Algorithms

Our notion of *pseudo-deterministic* algorithms intuitively captures that no distinguisher can query the algorithm on an input such that it returns something different from the most likely output. That is, the adversary wins if in its set of input/output pairs to the algorithm there is a query for which the algorithm did not return the most likely response. We also introduce a weak notion of this property, where we call \mathcal{A} pseudo-deterministic for a specific distinguisher if the probability of the distinguisher winning in the above experiment is negligible.

Our notion of pseudo-determinism can be seen as an average-case version of the pseudo-deterministic algorithms as recently introduced by Goldreich et al. [GGR12]. While they require probabilism to be hard to detect on any input, we only require indistinguishability for efficiently generatable inputs, on average.

Definition 3.3. *Let λ be a security parameter and $\mathcal{A}^\mathcal{O}$ a stateless probabilistic polynomial-time oracle Turing machine with access to some oracle \mathcal{O} . Let $L[\mathcal{D}, \mathcal{A}, \mathcal{O}]$ denote the induced set of input/output pairs (x, y) of $\mathcal{A}^\mathcal{O}$ when queried arbitrarily many times by the distinguisher \mathcal{D} , where \mathcal{A} uses fresh coins in each run. We say that $\mathcal{A}^\mathcal{O}$ is pseudo-deterministic if for all probabilistic polynomial-time distinguishers \mathcal{D} there exists a negligible function negl , such that*

$$\text{Prob}_{\mathcal{D}, \mathcal{A}, \mathcal{O}} \left[\forall (x, y) \in L[\mathcal{D}, \mathcal{A}, \mathcal{O}] \quad y = y_{x, \mathcal{A}^\mathcal{O}} \right] \geq 1 - \text{negl}(\lambda). \quad (3)$$

The notation $y_{x, \mathcal{A}^\mathcal{O}}$ denotes the most likely output of \mathcal{A} on input x over the randomness of \mathcal{A} , i.e., conditioned on a fixed oracle \mathcal{O} . If there are two equally likely answers on input x , we choose $y_{x, \mathcal{A}^\mathcal{O}}$ to be the lexicographically smaller one.

We say algorithm $\mathcal{A}^\mathcal{O}$ is pseudo-deterministic for distinguisher $\mathcal{D}^{\mathcal{A}^\mathcal{O}(1^\lambda, \cdot)}(1^\lambda)$, if there exists negligible function negl , such that equation (3) holds for \mathcal{D} .

Note that the definition of \mathcal{A} being pseudo-deterministic for distinguisher \mathcal{D} does not imply that it is hard to distinguish whether \mathcal{A} is probabilistic or deterministic—it is only hard for a particular algorithm \mathcal{D} . Although this might sound like a weak and somewhat useless property, it will be sufficient to show that if a simulator is pseudo-deterministic for a distinguisher, then the simulator can be entirely derandomized via random oracles/ideal ciphers.

We now show that strong multi-stage indistinguishability implies that the simulators are not only stateless but also pseudo-deterministic. This is captured by the following lemma.

Lemma 3.4. *Let G^π be a construction with black-box access to primitive π which is strongly multi-stage indistinguishable from primitive Π . Then there is a stateless pseudo-deterministic*

probabilistic polynomial-time simulator \mathcal{S} such that for all probabilistic polynomial-time distinguishers \mathcal{D} equation (2) holds in the strong case.

Proof. Let us assume there exists stateless simulator \mathcal{S} such that for all distinguishers \mathcal{D} equation (2) holds and such that \mathcal{S} is not pseudo-deterministic. The latter implies that there exists distinguisher \mathcal{D}_{pd} against the pseudo-determinism of simulator \mathcal{S} , i.e., there is a non-negligible probability that \mathcal{D}_{pd} asks a query to \mathcal{S} , where \mathcal{S} has a non-negligible probability of returning a different value than the most likely one. We now construct distinguisher \mathcal{D}' against strong multi-stage indistinguishability. Distinguisher \mathcal{D}' runs \mathcal{D}_{pd} on the adversarial \mathbf{a} -interface. Let q_1, \dots, q_t be the queries asked by \mathcal{D}_{pd} . Distinguisher \mathcal{D}' then sends the same queries once more to its \mathbf{a} -interface and returns 1 if at least one response does not match and 0 otherwise. If \mathcal{D}' is in the real world, talking to G^π and π algorithm \mathcal{D}' will always output 0 as π is a function. If on the other hand, \mathcal{D}' is in the ideal world, then \mathcal{D}_{pd} will succeed with noticeable probability and hence \mathcal{D}' will distinguish both worlds with noticeable probability, a contradiction. \square

DETERMINISTIC SIMULATORS. Bennett and Gill prove in [BG81] that relative to a random oracle the complexity classes \mathcal{BPP} and \mathcal{P} are equivalent. Let us quickly sketch their idea. Given a probabilistic polynomial time oracle Turing machine $\mathcal{M}^{\mathcal{R}}$ which has access to random oracle \mathcal{R} and which decides a language \mathcal{L} in \mathcal{BPP} we can prove the existence of a *deterministic* polynomial time Turing machine $\mathcal{D}^{\mathcal{R}}$ which also decides \mathcal{L} . Let us by $p(|x|)$ denote the runtime of machine $\mathcal{M}^{\mathcal{R}}$ for inputs of length $|x|$. As $\mathcal{M}^{\mathcal{R}}$ runs in polynomial time there exists a polynomial upper bound $p(|x|)$ on the length of queries $\mathcal{M}^{\mathcal{R}}$ can pose to the random oracle. To derandomize $\mathcal{M}^{\mathcal{R}}$ we construct a deterministic machine $\mathcal{D}^{\mathcal{R}}$ which works analogously to $\mathcal{M}^{\mathcal{R}}$ with the single exception that when $\mathcal{M}^{\mathcal{R}}$ requests a random coin then $\mathcal{D}^{\mathcal{R}}$ generates this coin deterministically by querying the random oracle on the next smallest input that cannot have been queried by $\mathcal{M}^{\mathcal{R}}$ due to its runtime restriction. As the random oracle produces perfect randomness, the machines decide the same language with probability 1 over the choice of random oracle.

Using the techniques developed by Bennet and Gill [BG81] we now show that in the multi-stage indistinguishability setting, if a simulator is pseudo-deterministic for a distinguisher \mathcal{D} , then it can be derandomized, in case the constructed primitive Π is a random oracle or an ideal cipher. When applied to a simulator \mathcal{S} that is universal for all distinguishers (strong indistinguishability), these derandomization techniques yield a family of simulators that depends only on the number of queries made by the distinguisher (weak indistinguishability). We give the proof in Appendix A.

Lemma 3.5. *Let \mathcal{A}^Π be a stateless probabilistic polynomial-time algorithm with oracle access to a random oracle $\mathcal{R}_{\ell,m}$ or an ideal cipher $\mathcal{E}_{k,n}$ for $\ell \in \omega(\log \lambda)$ (resp., $(k+n) \in \omega(\log \lambda)$). Let s be polynomial in λ . From \mathcal{A}^Π , we construct a deterministic algorithm \mathcal{B}^Π such that the following holds: for all efficient distinguisher \mathcal{D} that make less than s queries to their oracle, it holds that if \mathcal{A}^Π is pseudo-deterministic for \mathcal{D} , then*

$$\left| \text{Prob}_{R,\Pi} \left[\mathcal{D}^{\Pi, \mathcal{A}^\Pi(R, \cdot)}(1^\lambda) = 1 \right] - \text{Prob}_\Pi \left[\mathcal{D}^{\Pi, \mathcal{B}^\Pi(\cdot)}(1^\lambda) = 1 \right] \right|$$

is negligible, where the probability is over the choice of oracle Π and algorithm \mathcal{A} 's and distinguisher \mathcal{D} 's internal coin tosses for the first case and over the choice of oracle Π and distinguisher \mathcal{D} 's internal coin tosses in the second.

Remark. While lemma 3.4 says that simulators need to be pseudo-deterministic, we will, for the sake of readability, make the simplifying assumption that simulators for reset indistinguishability are deterministic throughout the paper. Nevertheless, our results hold in either case. In Appendix B we point out how our proofs would be affected if this assumption were not in place.

4 The Random Oracle and Ideal Cipher Model are Incomparable

In this section we prove that the random oracle model and the ideal cipher model are incomparable with respect to strong multi-stage indistinguishability. We start by giving an alternative, simpler proof of the fact that multi-stage indistinguishable constructions cannot be built via domain extension [DGHM13, LAMP12] (Lemma 4.1). [DGHM13] rule out domain extension even for a single bit of extension. In turn, we obtain an easier proof in the setting where the extension factor is super-logarithmic. In Section 4.1 we then present our duality lemma for multi-stage indistinguishability which allows us to conclude that the ROM and the ICM are incomparable with respect to strong multi-stage indistinguishability.

Lemma 4.1. *Let \mathcal{R} be a random oracle with domain $\{0, 1\}^\ell$ (resp., \mathcal{E} be an ideal cipher with domain $\{0, 1\}^k \times \{0, 1\}^n$) and π be any ideal primitive with domain size 2^v . For $\ell - v \in \omega(\log(\lambda))$ (resp., $k + n - v \in \omega(\log(\lambda))$) there exists no construction G^π that is weakly multi-stage indistinguishable from \mathcal{R} (resp., \mathcal{E}).*

We prove Lemma 4.1 for the random oracle case; the proof for ideal ciphers works analogously. Note that we prove the statement for weak multi-stage indistinguishability, thereby essentially ruling out any (possibly non-black-box) construction.

In the following proof we consider a particular distinguisher that tests for the ideal world by forcing the simulator to query its oracle on a particular value M . We show that no simulator is able to do this with more than negligible probability since M is drawn from a very large set while the simulator, being stateless, is only able to make queries from a negligible fraction of this large set; it thus fails to pass the test.

Proof of Lemma 4.1. Assume towards contradiction that there exists construction G^π that is weakly multi-stage indistinguishable from random oracle \mathcal{R} and, hence, for every distinguisher \mathcal{D} there exists a stateless simulator \mathcal{S} such that \mathcal{D} cannot distinguish between the real and ideal world.

We consider a distinguisher $\mathcal{D}^{\mathbf{h}, \mathbf{a}}$ with access to honest and adversarial interfaces (\mathbf{h}, \mathbf{a}) which implement the random oracle \mathcal{R} and simulator \mathcal{S} in the ideal world and construction G^π and ideal primitive π in the real world. The distinguisher \mathcal{D} chooses a message $M \in$

$\{0, 1\}^\ell$ uniformly at random and executes construction G via an internal simulation using its adversarial interface \mathfrak{a} , i.e., it computes $G^{\mathfrak{a}}(M)$. Then, the distinguisher asks its honest interface on message M to compute $h(M)$ and returns 1 if the two results agree and 0 otherwise. Note that in the real world distinguisher \mathcal{D} will always output 1. Thus, the simulator \mathcal{S} has to ensure that $G^{\mathcal{S}^{\mathcal{R}}}(M)$ is equal to $\mathcal{R}(M)$ with overwhelming probability over the choice of the random oracle \mathcal{R} . We now prove that, in the ideal world, the two values match only with negligible probability over the choice of the message M and the two settings can thus be distinguished by \mathcal{D} .

Let us assume the ideal world and denote the query/response pairs to the \mathfrak{a} -interface with $(q_i, r_i)_{1 \leq i \leq t}$. We analyze the simulator's behavior when it is asked these queries q_1, \dots, q_t . If for none of the q_i the simulator \mathcal{S} asks the random oracle on M , then the answer of $G^{\mathcal{S}^{\mathcal{R}}}(M)$ is independent of $\mathcal{R}(M)$ and thus different with overwhelming probability. By a simple counting argument, we now prove that, with high probability over the choice of M , on *no* query (not even one outside of the set $(q_i, r_i)_{1 \leq i \leq t}$), the simulator \mathcal{S} asks \mathcal{R} on M . For this, note that the queries which simulator \mathcal{S} receives are of length v . Hence there are at most 2^v distinct possible queries to \mathcal{S} . Denote by c the upper bound on the number of queries that \mathcal{S} asks to its random oracle over all possible queries that \mathcal{S} itself receives. As the simulator \mathcal{S} runs in polynomial time c exists and is polynomial. Noting that \mathcal{S} is stateless, we conclude that \mathcal{S} asks at most $c2^v \ll 2^\ell$ queries. Hence the probability that the distinguisher's M is in the set

$$\{M : \exists q \mathcal{S}^{\mathcal{R}} \text{ asks } M \text{ on input } q\}$$

is negligible. The probability that the distinguisher \mathcal{D} returns 1 in the ideal world where it is given access to simulator \mathcal{S} and a random oracle \mathcal{R} is therefore also negligible. Thus, the distinguisher \mathcal{D} has a distinguishing advantage of almost 1 which concludes the proof. \square

4.1 The Duality Lemma for Multi-Stage Indifferentiability

We now prove the inverse direction, that is an ideal cipher cannot be build from a random oracle with larger domain. In contrast to the previous section we here give an impossibility result for strong multi-stage indifferentiability. Our result is, however, more general and of independent interest. Strong multi-stage indifferentiability guarantees the existence of a simulator that is stateless and deterministic. Constructions of ideal primitives often need to be stateless and deterministic as well. If for example, the construction, implements a publicly accessible function such as a hash function, it needs has to be stateless. Note that this is the case both for random oracles and ideal ciphers.

If we assume constructions to be deterministic and stateless this allows us to exchange the role of construction and simulator thereby turning an impossibility result (resp. feasibility result) in one direction into an impossibility result (resp. feasibility result) in the other direction, or likewise, we can transform a positive result in one direction into a positive result in the other direction. We formalize this idea in the Duality Lemma and see how to prove it in the case of strong multi-stage indifferentiability.

Lemma 4.2 (Duality Lemma for Multi-Stage Indifferentiability). *Let π and π' be two ideal primitives. Assuming constructions are stateless and deterministic, then one of two following statements holds:*

1. *The two primitives are computationally equivalent, i.e., there exist constructions G_1, G_2 such that G_1^π is strongly multi-stage indifferentiable from π' and $G_2^{\pi'}$ is strongly multi-stage indifferentiable from π , or*
2. *π and π' are incomparable with respect to strong multi-stage indifferentiability.*

In essence this means that a positive or negative result in either direction gives us a result for the other direction. As we have already seen a negative result for domain extenders this gives us the result for the other directions, i.e., going from a large random oracle \mathcal{R} to a small ideal cipher \mathcal{E} , or from a large ideal cipher \mathcal{E} to a small random oracle \mathcal{R} .

Proof of Lemma 4.2. Assume construction G^π with black-box access to ideal primitive π is strongly multi-stage indifferentiable from π' . Then by definition there exists a deterministic, stateless simulator \mathcal{S} such that no distinguisher \mathcal{D} can tell apart the ideal world $(\pi', \mathcal{S}^{\pi'})$ from the real world (G^π, π) . Likewise, by definition, G is stateless and deterministic. We now exchange the roles of construction G and simulator \mathcal{S} , thereby getting a new “construction” $\mathcal{S}^{\pi'}$ implementing primitive π . It remains to show that $\mathcal{S}^{\pi'}$ is strongly multi-stage indifferentiable from π .

Let us assume the contrary. Then there exists distinguisher \mathcal{D} that can distinguish between the settings $(\pi', \mathcal{S}^{\pi'})$ and the setting (G^π, π) . This, however, contradicts the assumption that G^π is strongly multi-stage indifferentiable from π' . \square

An immediate consequence of the duality lemma and Lemma 4.1 is captured by the following corollary:

Corollary 4.3. *The ideal cipher model and the random oracle model are incomparable with respect to strong multi-stage indifferentiability.*

Remark. One interesting consequence of the duality lemma is best seen by an example: Can a random oracle with smaller domain be constructed from a random oracle with a larger domain? Intuitively, it feels natural to assume that this works. However, Lemma 4.1 tells us, that the inverse is not possible and, thus, by the duality lemma we can directly conclude that any construction using a large random oracle cannot be strongly multi-stage indifferentiable from a small random oracle. So far, we have failed to either prove impossibility for weak multi-stage indifferentiability or to come up with a construction. We leave this for future work.

5 Single versus Multi-Reset

Luykx et al. [LAMP12] introduce the presumably weaker notion of n -reset indifferentiability, where the distinguisher is allowed to reset the simulator only n times. Naturally, for a

construction that is n -reset indifferntiable the composition theorem holds for games that have $n + 1$ or less stages. In the following we show that, however, already the extreme single-reset notion implies full reset indifferntiability for simulators that do not depend on the distinguisher (i.e., the *strong* case). This yields that also for n -reset indifferntiability all our separations hold in a black-box fashion.

Lemma 5.1. *Let G^π be a deterministic, stateless construction with black-box access to primitive π which is strongly 1-reset indifferntiable from primitive Π via a deterministic, stateless simulator. Then, G^π is also strongly multi-stage indifferntiable from primitive Π .*

The proof of this lemma makes use of the following Claim 5.2, which we deal with later. For this, let $Q[\mathcal{D}, r_{\mathcal{D}}, \Pi, \mathcal{S}]$ denote the ordered list of query/answer pairs induced by running \mathcal{D} with randomness $r_{\mathcal{D}}$ and access to oracles Π and \mathcal{S} .

Claim 5.2. *Let G^π be a construction with black-box access to primitive π which is strongly 1-reset indifferntiable from primitive Π and denote the corresponding simulator by \mathcal{S} . Define the simulator \mathcal{S}' as running \mathcal{S} and resetting itself after each query it receives. Then, for any probabilistic polynomial-time algorithm \mathcal{D}_{SS} with access to an ideal primitive Π and a simulator \mathcal{S} , with high probability over the choice of Π and r , it holds that their induced query-answer lists $Q[\mathcal{D}_{\text{SS}}, r, \Pi, \mathcal{S}]$ and $Q[\mathcal{D}_{\text{SS}}, r, \Pi, \mathcal{S}']$ are equal, formally, there exists negligible function negl such that*

$$\text{Prob}_{\Pi, r} [Q[\mathcal{D}_{\text{SS}}, r, \Pi, \mathcal{S}] = Q[\mathcal{D}_{\text{SS}}, r, \Pi, \mathcal{S}']] \geq 1 - \text{negl}(\lambda). \quad (4)$$

Let us see how we can prove Lemma 5.1 with the help of Claim 5.2.

Proof of Lemma 5.1. Let us by \approx denote that two values are negligibly close, that is

$$a \approx b \iff |a - b| \leq \text{negl}(\lambda)$$

for some negligible function negl .

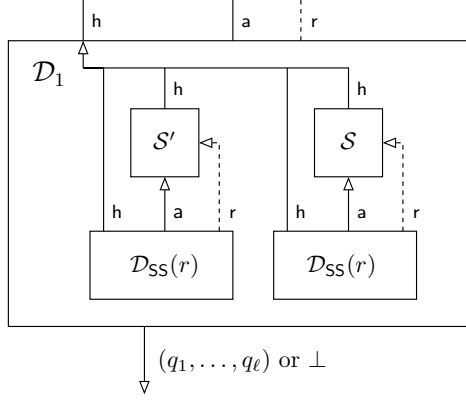
Let \mathcal{S} be the simulator guaranteed by the fact that G^π is strongly 1-reset indifferntiable from primitive Π , that is, for all efficient distinguishers \mathcal{D} we have that

$$\text{Prob} \left[\mathcal{D}^{G, \pi}(1^\lambda) = 1 \right] \approx \text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}^\Pi}(1^\lambda) = 1 \right]. \quad (5)$$

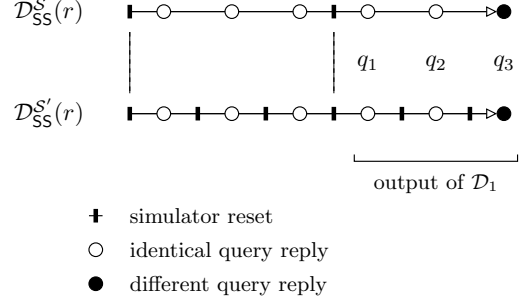
Let \mathcal{S}' be the simulator from Claim 5.2. By Claim 5.2, we have that for all efficient distinguishers \mathcal{D} , it holds that

$$\text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}^\Pi}(1^\lambda) = 1 \right] \approx \text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}'^\Pi}(1^\lambda) = 1 \right].$$

Thus, we can transform \mathcal{D} into a non-resetting distinguisher with exactly the same behavior. Let \mathcal{D}' be as \mathcal{D} with the exception that it does not forward \mathcal{D} 's reset queries. Then, we



(a) Schematic setup of the first phase \mathcal{D}_1 of \mathcal{D} . Wires labeled with r indicate the reset interface.



(b) Schematic view of the queries from either \mathcal{D}_{SS} to their simulators.

Figure 1

have that

$$\begin{aligned}
 & \text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}^{\Pi}}(1^\lambda) = 1 \right] \\
 & \approx \text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}'^{\Pi}}(1^\lambda) = 1 \right] && \text{(Claim 5.2)} \\
 & = \text{Prob} \left[\mathcal{D}'^{\Pi, \mathcal{S}'}(1^\lambda) = 1 \right] && (\mathcal{S}' \text{ resets after each step}) \\
 & = \text{Prob} \left[\mathcal{D}'^{\Pi, \mathcal{S}}(1^\lambda) = 1 \right] && \text{(Claim 5.2)} \\
 & \approx \text{Prob} \left[\mathcal{D}'^{G^\pi, \pi}(1^\lambda) = 1 \right] && \text{(Equation 5)} \\
 & = \text{Prob} \left[\mathcal{D}^{G^\pi, \pi}(1^\lambda) = 1 \right] && (6) \\
 & && (7)
 \end{aligned}$$

where Equation 6 follows from the assumption that \mathcal{S} is good for all \mathcal{D}' that make less than a certain number of reset queries which is certainly satisfied for \mathcal{D}' who does not make even a single reset query. And Equation 7 follows, as resets do not have any effect in the real world. As \mathcal{S}' is clearly reset indistinguishable this concludes the proof. \square

What is left is to prove Claim 5.2:

Proof of Claim 5.2. Assume towards contradiction that there exists a probabilistic polynomial-time algorithm \mathcal{D}_{SS} such that Equation (4) does not hold, that is, there is a non-negligible function $\nu(\lambda)$ such that

$$1 - \nu(\lambda) = \text{Prob}_{\Pi, r} \left[Q[\mathcal{D}_{SS}, r, \Pi, \mathcal{S}] = Q[\mathcal{D}_{SS}, r, \Pi, \mathcal{S}'] \right].$$

We now transform \mathcal{D}_{SS} into a successful distinguisher \mathcal{D} for 1-reset indistinguishability, thereby contradicting the assumption. The distinguisher $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ is a two phase

algorithm. In the first phase, \mathcal{D}_1 (depicted in Figure 1) only makes queries to the honest interface but no queries to its adversarial interface. In turn, in the second phase, \mathcal{D}_2 makes queries to both, the adversarial and the honest interface.

In the first phase, \mathcal{D}_1 runs two local separate copies of $\mathcal{D}_{\mathcal{S}\mathcal{S}}$ using the same uniformly sampled randomness r . It keeps a record of all queries, answers, and resets performed in both runs. In one run, $\mathcal{D}_{\mathcal{S}\mathcal{S}}$ has access to a locally simulated \mathcal{S} and in the other run $\mathcal{D}_{\mathcal{S}\mathcal{S}}$ has access to a locally simulated \mathcal{S}' . The reset requests issued by $\mathcal{D}_{\mathcal{S}\mathcal{S}}$ are honored in both cases within the simulation. Note that in this phase no query (reset or other) to the adversarial interface is forwarded to the adversarial interface that \mathcal{D} has access to. In contrast, \mathcal{D}_1 *does* forward any query that either copy of $\mathcal{D}_{\mathcal{S}\mathcal{S}}$ makes to the honest interface to its own honest interface. Similarly, any query by either of the simulated simulators \mathcal{S} and \mathcal{S}' to the honest interface are forwarded by \mathcal{D}_1 to its own honest interface. This simulation is performed until both distinguishers terminate or until there is a query that is answered differently by \mathcal{S} and \mathcal{S}' . Let (q_1, \dots, q_ℓ) be the queries in the interaction between $\mathcal{D}_{\mathcal{S}\mathcal{S}}$ and \mathcal{S} since the last reset of \mathcal{S} , or since the beginning of the simulation if \mathcal{S} did not receive any reset queries.

We define the following event \mathbf{E} :

\mathbf{E} : There is no query that is answered differently by \mathcal{S} and \mathcal{S}' .

$\bar{\mathbf{E}}$: There is a query that is answered differently by \mathcal{S} and \mathcal{S}' .

If event \mathbf{E} happens, then \mathcal{D}_1 passes a special symbol to \mathcal{D}_2 ; and \mathcal{D}_2 returns a random bit. If event $\bar{\mathbf{E}}$ happens, then \mathcal{D}_1 passes the queries q_1, \dots, q_ℓ to \mathcal{D}_2 . Note that q_ℓ denotes the query that previously answered differently. The distinguisher \mathcal{D}_2 replays these queries on its adversarial interface and records the answers y_1, \dots, y_ℓ . It then issues a reset request and asks the single query q_ℓ once again, storing the answer as y'_ℓ . Finally, it outputs 0 if $y_\ell = y'_\ell$ and 1 if $y_\ell \neq y'_\ell$.

ANALYSIS. Before we continue let us provide some intuition on why $\mathcal{D} := (\mathcal{D}_1, \mathcal{D}_2)$ distinguishes the real from the ideal world. In the real world y_ℓ will always match y'_ℓ as π is a function, thus, giving the same answer on the same query. In the ideal world, on the other hand, the two values differ with the same probability that the two internal simulations produced a different value, that is with probability $\text{Prob}[\bar{\mathbf{E}}]$.

By definition, $\nu(\lambda)$ denotes the probability that event $\bar{\mathbf{E}}$ happens in the ideal world, that is, if the honest interface of \mathcal{D} is implemented by Π :

$$\nu(\lambda) := \text{Prob}_{\Pi, \mathcal{D}} [Q[\mathcal{D}_{\mathcal{S}\mathcal{S}}, r, \Pi, \mathcal{S}] \neq Q[\mathcal{D}_{\mathcal{S}\mathcal{S}}, r, \Pi, \mathcal{S}']] .$$

The probability is over the choice of Π and \mathcal{D} 's internal coins. Note that the two simulators \mathcal{S} and \mathcal{S}' are deterministic. The outer simulator that \mathcal{D} is connected to is irrelevant for event $\bar{\mathbf{E}}$.

We now prove that the probability of \mathcal{D} outputting 1 in the ideal world, that is, $\text{Prob}[\mathcal{D}^{\Pi, \mathcal{S}^{\Pi}}(1^\lambda) = 1]$ is greater than $\frac{1}{2} + \frac{1}{2}\nu(n)$. Moreover, the probability that the distinguisher \mathcal{D} outputs 1 in the real world, i.e, $\text{Prob}[\mathcal{D}^{G^\pi, \pi}(1^\lambda) = 1]$ is at most $\frac{1}{2}$, because

the distinguisher always flips a random coin, as the deterministic construction G^π always produces identical queries. Thus,

$$\left| \text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}^\Pi}(1^\lambda) = 1 \right] - \text{Prob} \left[\mathcal{D}^{G^\pi, \pi}(1^\lambda) = 1 \right] \right| \geq \frac{1}{2} \nu(\lambda) \quad (8)$$

contradicting the assumption that construction G^π is single-reset indistinguishable from Π and hence such a \mathcal{D}_{SS} cannot exist.

We now estimate $\text{Prob} \left[\mathcal{D}^{\Pi, \mathcal{S}^\Pi}(1^\lambda) = 1 \right]$, that is, the probability of \mathcal{D} outputting 1 in the ideal world.

$$\begin{aligned} \text{Prob}_{\Pi, \mathcal{D}} \left[\mathcal{D}^{\Pi, \mathcal{S}^\Pi}(1^\lambda) = 1 \right] &= \frac{1}{2} \cdot \text{Prob}_{\Pi, \mathcal{D}}[\mathbf{E}] + \text{Prob}_{\Pi, \mathcal{D}}[\bar{\mathbf{E}}] \cdot \text{Prob}_{\Pi, \mathcal{D}} \left[\mathcal{D}^{\Pi, \mathcal{S}^\Pi}(1^\lambda) = 1 \mid \bar{\mathbf{E}} \right] \\ &= \frac{1}{2} \cdot (1 - \nu(\lambda)) + \nu(\lambda) \cdot \text{Prob}_{\Pi, \mathcal{D}} \left[\mathcal{D}^{\Pi, \mathcal{S}^\Pi}(1^\lambda) = 1 \mid \bar{\mathbf{E}} \right] \\ &\approx \frac{1}{2} - \frac{1}{2} \nu(n) + \nu(n) \cdot 1 \\ &= \frac{1}{2} + \frac{1}{2} \nu(n), \end{aligned} \quad (9)$$

where the second to last transformation (Equation 9) follows from the determinism of \mathcal{S} and the fact that for a single query (since the last reset), $\mathcal{S} = \mathcal{S}'$. In turn,

$$\begin{aligned} \text{Prob}_{\Pi, \mathcal{D}} \left[\mathcal{D}^{G^\pi, \pi}(1^\lambda) = 1 \right] &= \frac{1}{2} \cdot \text{Prob}_{\Pi, \mathcal{D}}[\mathbf{E}] + \text{Prob}_{\Pi, \mathcal{D}}[\bar{\mathbf{E}}] \cdot \text{Prob}_{\Pi, \mathcal{D}} \left[\mathcal{D}^{G^\pi, \pi}(1^\lambda) = 1 \mid \bar{\mathbf{E}} \right] \\ &= \frac{1}{2} \cdot \text{Prob}_{\Pi, \mathcal{D}}[\mathbf{E}] + \text{Prob}_{\Pi, \mathcal{D}}[\bar{\mathbf{E}}] \cdot 0 \\ &\leq \frac{1}{2}. \end{aligned}$$

□

6 Acknowledgements

We thank Pooya Farshim and Giorgia Azzurra Marson for their valuable comments on preliminary versions of this work. Paul Baecher is supported by grant Fi 940/4-1 of the German Research Foundation (DFG). Christina Brzuska is supported by the Israel Science Foundation (grant 1076/11 and 1155/11), the Israel Ministry of Science and Technology (grant 3-9094), and the German-Israeli Foundation for Scientific Research and Development (grant 1152/2011). Arno Mittelbach is supported by CASED (www.cased.de).

References

[BBN⁺09] Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect

- against bad randomness. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 232–249, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany. (Cited on pages 4 and 6.)
- [BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O’Neill. Deterministic and efficiently searchable encryption. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Berlin, Germany. (Cited on pages 4 and 6.)
- [BCFW09] Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi. Foundations of non-malleable hash and one-way functions. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 524–541, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany. (Cited on page 4.)
- [BG81] C. H. Bennett and J. Gill. Relative to a random oracle A , $P^A \neq NP^A \neq coNP^A$ with probability 1. *SIAM Journal on Computing*, 10(1):96–113, 1981. (Cited on page 12.)
- [BK03] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 491–506, Warsaw, Poland, May 4–8, 2003. Springer, Berlin, Germany. (Cited on page 4.)
- [BR93] Mihir Bellare and Phillip 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, November 3–5, 1993. ACM Press. (Cited on page 3.)
- [BR94] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT’94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111, Perugia, Italy, May 9–12, 1994. Springer, Berlin, Germany. (Cited on page 3.)
- [BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT’96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, Saragossa, Spain, May 12–16, 1996. Springer, Berlin, Germany. (Cited on page 3.)
- [BRS03] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M.

- Heys, editors, *SAC 2002: 9th Annual International Workshop on Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 62–75, St. John’s, Newfoundland, Canada, August 15–16, 2003. Springer, Berlin, Germany. (Cited on page 4.)
- [BRSS10] John Black, Phillip Rogaway, Thomas Shrimpton, and Martijn Stam. An analysis of the blockcipher-based hash functions from PGV. *Journal of Cryptology*, 23(4):519–545, October 2010. (Cited on page 3.)
- [CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany. (Cited on pages 3, 4, and 8.)
- [CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany. (Cited on pages 3 and 4.)
- [DGHM13] Grégory Demay, Peter Gaži, Martin Hirt, and Ueli Maurer. Resource-restricted indifferenciability. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 665–684, Athens, Greece, May 26–30, 2013. Springer, Berlin, Germany. (Cited on pages 5, 7, 9, and 13.)
- [DKW11] Stefan Dziembowski, Tomasz Kazana, and Daniel Wichs. Key-evolution schemes resilient to space-bounded leakage. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 335–353, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Berlin, Germany. (Cited on page 4.)
- [EM97] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–162, 1997. (Cited on page 8.)
- [GGR12] Oded Goldreich, Shafi Goldwasser, and Dana Ron. On the possibilities and limitations of pseudodeterministic algorithms. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:101, 2012. (Cited on pages 6 and 11.)
- [HKT11] Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 89–98, San Jose, California, USA, June 6–8, 2011. ACM Press. (Cited on pages 3 and 4.)

- [LAMP12] Atul Luykx, Elena Andreeva, Bart Mennink, and Bart Preneel. Impossibility results for indifferntiability with resets. Cryptology ePrint Archive, Report 2012/644, 2012. <http://eprint.iacr.org/>. (Cited on pages 5, 6, 13, and 15.)
- [MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferntiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany. (Cited on pages 3, 8, and 9.)
- [RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferntiability framework. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany. (Cited on pages 3, 7, and 9.)
- [Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949. (Cited on page 3.)

A Missing Proofs for Section 3

A.1 Proof of Lemma 3.5

Lemma (Lemma 3.5, restated). *Let \mathcal{A}^Π be a stateless probabilistic polynomial-time algorithm with oracle access to a random oracle $\mathcal{R}_{\ell,m}$ or an ideal cipher $\mathcal{E}_{k,n}$ for $\ell \in \omega(\log \lambda)$ (resp., $(k+n) \in \omega(\log \lambda)$). Let s be polynomial in λ . From \mathcal{A}^Π , we construct a deterministic algorithm \mathcal{B}^Π such that the following holds: for all efficient distinguisher \mathcal{D} that make less than s queries to their oracle, it holds that if \mathcal{A}^Π is pseudo-deterministic for \mathcal{D} , then*

$$\left| \text{Prob}_{R,\Pi} \left[\mathcal{D}^{\Pi, \mathcal{A}^\Pi(R, \cdot)}(1^\lambda) = 1 \right] - \text{Prob}_\Pi \left[\mathcal{D}^{\Pi, \mathcal{B}^\Pi(\cdot)}(1^\lambda) = 1 \right] \right|$$

is negligible, where the probability is over the choice of oracle Π and algorithm \mathcal{A} 's and distinguisher \mathcal{D} 's internal coin tosses for the first case and over the choice of oracle Π and distinguisher \mathcal{D} 's internal coin tosses in the second.

Proof. Let \mathcal{A}^Π be a stateless algorithm with access to ideal primitive Π where Π is either a random oracle $\mathcal{R}_{\ell,m}$ or an ideal cipher $\mathcal{E}_{k,n}$.

Let \mathcal{D} be an efficient distinguisher for which \mathcal{A}^Π is pseudo-deterministic. As distinguisher \mathcal{D} is efficient, there exists an upper bound $p(|\lambda|)$ on the number of queries to the Π -interface by \mathcal{D} . We construct a deterministic algorithm \mathcal{B} which works as \mathcal{A} with the only exception that \mathcal{B} deterministically generates “random” bits by querying its random oracle, whenever \mathcal{A} makes use of a random bit. For the j th requested random bit, algorithm \mathcal{B} calls the Π -oracle (either random oracle \mathcal{R} or ideal cipher \mathcal{E} where it uses the encryption interface of

\mathcal{E}) on $p(|\lambda|) + j$ distinct values xor-ing the result and choosing a bit from this result. Note that as $\ell \in \omega(\log \lambda)$ (resp., $n + k \in \omega(\log \lambda)$) there exist sufficiently many distinct values.

Remember that we denote by $y_{q, \mathcal{A}^\mathcal{O}}$ the most likely output of algorithm \mathcal{A} on input q conditioned on fixed oracle \mathcal{O} . We want to prove that

$$\left| \text{Prob}_{\Pi, \mathcal{D}, \mathcal{A}} \left[\mathcal{D}^{\Pi, \mathcal{A}^\Pi}(1^\lambda) = 1 \right] - \text{Prob}_{\Pi, \mathcal{D}} \left[\mathcal{D}^{\Pi, \mathcal{B}^\Pi}(1^\lambda) = 1 \right] \right|$$

is negligible in λ . We prove a stronger statement, namely, that the outputs of \mathcal{A} and \mathcal{B} are likely to be identical. We define event \mathbf{C} capturing that “the outputs of \mathcal{A} and \mathcal{B} agree on all inputs.” Towards this goal we define event \mathbf{A} as “algorithm \mathcal{A} returns y_{q_i, \mathcal{A}^Π} for all queries q_i ” where y_{q_i, \mathcal{A}^Π} is the most likely answer of \mathcal{A}^Π on input q_i , i.e., we set $y_{q_i, \mathcal{A}^\Pi} := \arg \max_y \{ \text{Prob}_R [\mathcal{A}^\Pi(q_i; R) = y] \}$ (cf. Definition 3.3). Likewise, we define event \mathbf{B} as “algorithm \mathcal{B} returns y_{q_i, \mathcal{A}^Π} for all queries q_i .” We will show that

$$\text{Prob}_{\Pi, \mathcal{D}, \mathcal{A}}[\mathbf{A}] \geq 1 - \text{negl} \tag{10}$$

and

$$\text{Prob}_{\Pi, \mathcal{D}}[\mathbf{B}] \geq 1 - \text{negl}. \tag{11}$$

Clearly, the probability that \mathcal{A} and \mathcal{B} produce the same answers for all q_i is lower bounded by the probability that \mathcal{A} and \mathcal{B} both output y_{q_i, \mathcal{A}^Π} for all q_i . Thus,

$$\begin{aligned} \text{Prob}_{\Pi, \mathcal{D}, \mathcal{A}}[\mathbf{C}] &\geq \text{Prob}_{\Pi, \mathcal{D}, \mathcal{A}}[\mathbf{A} \wedge \mathbf{B}] \\ &= 1 - \text{Prob}_{\Pi, \mathcal{D}, \mathcal{A}}[\neg \mathbf{A} \vee \neg \mathbf{B}] \\ &\geq 1 - (\text{Prob}_{\Pi, \mathcal{D}, \mathcal{A}}[\neg \mathbf{A}] + \text{Prob}_{\Pi, \mathcal{D}}[\neg \mathbf{B}]) \\ &\geq 1 - \text{negl} - \text{negl}. \end{aligned}$$

Let us now make these statements formal as well as prove inequalities (10) and (11). We denote with q_i the queries to \mathcal{A} by \mathcal{D} and by R_i the randomness used by \mathcal{A} on query q_i . We say that event \mathbf{A} occurs (over $\Pi, \mathcal{D}, R_1, \dots, R_n$), if

$$\forall i \mathcal{A}^\Pi(q_i; R_i) = y_{q_i, \mathcal{A}^\Pi}.$$

Note that the pseudo-determinism of \mathcal{A} for \mathcal{D} directly implies that

$$\text{Prob}_{\Pi, \mathcal{D}, R_1, \dots, R_n} [\forall i \mathcal{A}^\Pi(q_i; R_i) = y_{q_i, \mathcal{A}^\Pi}] \geq 1 - \text{negl}, \tag{12}$$

which establishes inequality (10). We say that event \mathbf{B} occurs (over Π, \mathcal{D}), if

$$\forall i \mathcal{B}^\Pi(q_i) = y_{q_i, \mathcal{A}^\Pi},$$

where q_i now denotes the queries by \mathcal{D} to algorithm \mathcal{B} . Inequality (11) we derive from inequality (10) via an averaging argument. Note that in inequality (12) we consider fresh

randomness R_i for every query q_i . If for all queries q_i a random choice of randomness is good with overwhelming probability, then a random choice of randomness is good for all q_i with overwhelming probability:

$$\text{Prob}_{\Pi, \mathcal{D}, R} [\forall i \mathcal{A}^\Pi(q_i; R) = y_{q_i, \mathcal{A}^\Pi}] \geq 1 - \text{negl}. \quad (13)$$

Moreover, when considering the random oracle via lazy sampling, one can observe that the randomness generated by \mathcal{B} from Π is independent from the part of Π that is used in the experiment, which yields that

$$\begin{aligned} \text{Prob}_{\Pi, \mathcal{D}} [\forall i; \mathcal{B}^\Pi(q_i) = y_{q_i, \mathcal{A}^\Pi}] &= \text{Prob}_{\Pi, \mathcal{D}, R} [\forall i; \mathcal{A}^\Pi(q_i; R) = y_{q_i, \mathcal{A}^\Pi}] \\ &\geq 1 - \text{negl} \end{aligned}$$

as desired. □

B Pseudo-Deterministic vs. Deterministic Simulators

Let us discuss how proofs need to be modified, if we replace deterministic simulators by pseudo-deterministic ones.

Duality Lemma The main idea of the Duality Lemma 4.2 is to exchange the role of the simulator and the construction. Namely, a construction G^π that is strongly reset indifferentiable from a primitive Π comes with a simulator \mathcal{S} , which, by Lemma 3.3 is also pseudo-deterministic and thus, \mathcal{S}^Π is actually strongly reset indifferentiable from primitive π using G as a simulator. However, if \mathcal{S} is not deterministic, but only pseudo-deterministic, then we need to slightly adapt the definition of a construction to allow for those algorithms. As pseudo-deterministic constructions are as useful as deterministic ones, all proofs still go through. In the remaining part of this section, we will also explain, where proofs change, if we consider pseudo-deterministic *constructions*.

Single-Stage versus Multi-Stage We will examine how the proofs of Lemma 5.1 change, if we consider pseudo-deterministic simulators \mathcal{S} and pseudo-deterministic constructions G . Namely, in the proof of Claim 5.2, the transformation in Equation 9 follows from pseudo-determinism of the simulator instead of determinism of the simulator \mathcal{S} and thus, the transformation is not an equality, but rather a transformation into a negligibly close term. Similarly, we argue before Equation 8 that the deterministic construction always returns consistent answers. If the construction is pseudo-deterministic only, then we loose a negligibly small amount that accounts for the pseudo-determinism of the construction G .