Impossibility Results for Indifferentiability with Resets

Atul Luykx, Elena Andreeva, Bart Mennink, and Bart Preneel

Dept. Electrical Engineering, ESAT/COSIC, KU Leuven, and IBBT, Belgium
{atul.luykx, elena.andreeva, bart.mennink, bart.preneel}@esat.kuleuven.be

Abstract. The indifferentiability framework of Maurer, Renner, and Holenstein (MRH) has gained immense popularity in recent years and has proved to be a powerful way to argue security of cryptosystems that enjoy proofs in the random oracle model. Recently, however, Ristenpart, Shacham, and Shrimpton (RSS) showed that the composition theorem of MRH has a more limited scope than originally thought, and that extending its scope required the introduction of reset-indifferentiability, a notion which no practical domain extenders satisfy with respect to random oracles.

In light of the results of RSS, we set out to rigorously tackle the specifics of indifferentiability and reset-indifferentiability by viewing the notions as special cases of a more general definition. Our contributions are twofold. Firstly, we provide the necessary formalism to refine the notion of indifferentiability regarding composition. By formalizing the definition of stage minimal games we expose new notions lying in between regular indifferentiability (MRH) and reset-indifferentiability (RSS).

Secondly, we answer the open problem of RSS by showing that it is impossible to build any domain extender which is reset-indifferentiable from a random oracle. This result formally confirms the intuition that reset-indifferentiability is too strong of a notion to be satisfied by any hash function. As a consequence we look at the weaker notion of single-reset-indifferentiability, yet there as well we demonstrate that there are no "meaningful" domain extenders which satisfy this notion. Not all is lost though, as we also view indifferentiability in a more general setting and point out the possibility for different variants of indifferentiability.

Keywords. indifferentiability, reset-indifferentiability, random oracle, hash functions

1 Introduction

The notion of indifferentiability was introduced by Maurer, Renner, and Holenstein (MRH) in [12] as an extension of the classical notion of indistinguishability. The main result of [12] allows one to securely replace a functionality φ_1 (i.e. a random oracle) in any game G with a different functionality φ_2 (i.e. a hash function), as long as φ_2 is shown to be indifferentiable from φ_1. Usually φ_1 is a functionality which allows for easier security analysis of a game G, thereby reducing the work needed in order to show that φ_2 is secure with various games. Since the introduction of the random oracle (RO) model by Bellare and Rogaway in [3], the security for many simple and efficient employed cryptographic schemes, such as OEAP [4] and FDH [5], has been proven in the RO model. An important implication of the result of MRH is that proving that your hash function is indifferentiable from a random oracle allows one to focus on the modular hash function design and conclude security against generic attacks.

Coron et al. [8] developed the idea of using the work of MRH [12] in the case of hash function domain extenders. Both the works of MRH [12] and Coron et al. [8] set a line of research on indifferentiability of hash domain extenders versus random oracles, which resulted in numerous indifferentiability proofs of hash domain extenders, such as [1, 2, 6, 7, 10, 11].

Several variants of indifferentiability [9, 13, 14, 16] have appeared in the literature, where random oracles are replaced with other functionalities. But despite the wide use of the indifferentiability security notion, the seminal indifferentiability framework of MRH has received little further theoretical attention and treatment. Recently, however, Ristenpart, Shacham, and Shrimpton (RSS) [15] revealed a surprising result. They showed that the types of games in which indifferentiable constructions can be securely replaced is limited. To understand their core observation and the underlying intricacies of the indifferentiability framework, we will

use a slightly modified form of the notation of RSS (a translation of terms from RSS to MRH can be found in Table 1 in Appendix A).

In [12], MRH define a functionality φ_1 to be at least as secure as a functionality φ_2, if for any given game G and adversaries $\mathcal{A}_1$, there exist adversaries $\mathcal{A}_2$ such that the quantity

$$\mathbf{Pr}\left(G_{\varphi_1}^{\mathcal{A}_1} \to 1\right) - \mathbf{Pr}\left(G_{\varphi_2}^{\mathcal{A}_2} \to 1\right) \tag{1}$$

is small. The game G is any procedure with binary output, which may run the adversaries, $\mathcal{A}_1$ or $\mathcal{A}_2$, and the honest interfaces to either φ_1 or φ_2. Here, by the honest interface we simply mean that part of φ_1 or φ_2 with which the game may interact, as opposed to the adversarial interface which only the adversaries may access. A useful way of thinking about making (1) small is that the objective of G is to distinguish the interaction of $\mathcal{A}_1$ and φ_1 from that of $\mathcal{A}_2$ and φ_2, whereas the objective of φ_1 is to mimic φ_2 as well as possible no matter what adversaries $\mathcal{A}_1$ it gets. We would like to explicitly mention that the above definition is given in terms of all games, which becomes an important point later on.

The connection between φ_1 being at least as secure as φ_2 and the indifferentiability of φ_1 with respect to φ_2 is made in Theorem 1 of [12] (which we refer to as the MRH theorem), where it is stated that these two concepts are equivalent. The proof relies on the fact that the adversaries $\mathcal{A}_2$ can be explicitly created by defining it to be a combination of the adversaries $\mathcal{A}_1$ and an extra procedure, usually called the *simulator*. The simulator is introduced via indifferentiability, where it is defined to be the adversary of a *distinguisher*. The distinguisher in turn can be considered as a convenient way of describing all games (anything a game can do, a distinguisher can do, and vice versa), hence one describes G and $\mathcal{A}_1$ in terms of a distinguisher and the MRH theorem is proved.

Yet in [15], RSS point out that there is a hidden assumption used in the proof of the MRH theorem which restricts the type of games to which the MRH theorem is applicable. In particular, the games should be restricted to *one-games* (single-stage games in RSS terminology), namely games which use *one* stateful adversary. This is because RSS realized that the distinguishers with which all indifferentiability proofs have been performed are only as powerful as one-games (hence, the distinguishers do not cover all games), therefore the simulators are only designed to work against one-games. As a result, if φ_1 is indifferentiable from φ_2, then the quantity in (1) is only guaranteed to be small when G is a one-game. To restore the scope of the MRH theorem, RSS introduced *reset-indifferentiability* as a generalization, where the simulator must be designed to withstand a distinguisher which can reset the simulator's state an arbitrary number of times, allowing the MRH theorem to apply to any n-game (n-stage or multi-stage game in RSS terminology).

Our Contributions

We first start by formalizing what exactly an n-game is in Section 2. Although RSS informally describe in [15] what they mean by an n(-stage)-game and a minimal n(-stage)-game, in order to further expand upon the topic of indifferentiability we need a rigorous definition of an n-game.

Then, by looking at indifferentiability from a more general point of view, in Section 3 we put the MRH theorem [12] in perspective and note that there are no faults in the application of indifferentiability. Rather, all existing proofs limit the scope of indifferentiability due to the nature of the distinguishers used. From this viewpoint we are led to a generalization of indifferentiability: $\mathcal{G}$-indifferentiability, where $\mathcal{G}$ is a class of games in which composition will hold. Indifferentiability, as commonly used, corresponds to the class of one-games, games with one stateful adversary. The reset-indifferentiability of RSS corresponds to the class of all games, thereby restoring the implication for any n-game. There are many more types of indifferentiability, as $\mathcal{G}$ could be any class of games. One of them is *single-reset-indifferentiability*, where

the distinguisher may reset the state of the simulator only once (as opposed to an arbitrary number of times).

As a first step we consider the possibilities of reset-indifferentiability and single-reset-indifferentiability. We answer the open problem of RSS [15] in proving that there are no domain extenders which are reset-indifferentiable from a random oracle (Section 4), a generalization of the impossibility result from [15] that no single-pass domain extenders (processing the message only once) can be reset-indifferentiable from a random oracle. The intuition behind this impossibility result is that the distinguisher does not allow the simulator to maintain any state as it resets the simulator after every call to it.

Going a step further, as a main contribution we prove that there are no "meaningful" domain extenders which are single-reset-indifferentiable from a random oracle (Section 5). Here, "meaningful" domain extenders are ones where the state size of the domain extender has a finite upper bound (e.g. it cannot grow with the size of the input message), and where modifications in the input message lead to different results with high probability (also known as the "avalanche effect"). The intuition behind the distinguisher is that it bases its strategy on the type of construction. If the domain extender is roughly single-pass, a generalization of the distinguisher by RSS does the job. On the other hand, if the domain extender processes a significant amount of bits more than once, the distinguisher resets at a deliberate time and smartly manipulates the simulator inputs in order to distinguish with high probability.

The observation that one cannot hope to find any meaningful domain extender that achieves single-reset-indifferentiability (let alone reset-indifferentiability) leads to the following conclusion: either there is no hope of finding a convenient way of linking the security of domain extenders with that of random oracles for n-games ($n > 1$), or the notions we are looking at are too strong. The distinguishers derived in this work provide evidence as to why the notions are too strong: if we attempt to prove indifferentiability for the class of all n-games, or the class of games covered by single-reset-indifferentiability, then the games corresponding to our distinguishers, which are rather unnatural, are members of these classes and must be taken into account. Hence, rather than ruling out indifferentiability, the question becomes what classes of games allow for composition and are meaningful to consider. A possible direction would be to consider more restricted, but perhaps more natural, classes of games in which to achieve composition. After all, the alternative of proving security for each game individually already restricts one to games which are considered natural. We elaborate on future work in Section 6.

2 Minimal Games

Our notation builds on the terminology used in [15] and introduces small modifications which facilitate the discussion on how we handle the state.

2.1 Procedures and Functionalities

We always talk of sequences of procedures in order to be able to leave procedures unspecified when defining games. These sequences are always assumed to be finite and are denoted with calligraphic capital letters (e.g. $\mathcal{S}$ as opposed to S). If $\mathcal{A}$ and $\mathcal{B}$ are two procedure sequences, then we say that they export the same interface if and only if they are of the same length and the ith procedure in $\mathcal{A}$ exports the same interface as the ith procedure in $\mathcal{B}$. A functionality is a pair of procedure sequences $\varphi := (\mathcal{H}, \mathcal{P})$ (instead of $\varphi.hon$ and $\varphi.adv$ as in [15]). An unspecified procedure sequence is a sequence of interfaces where the procedures have not been defined; an unspecified functionality is defined similarly. If $\mathcal{A}$ is a procedure sequence which has access to some unspecified procedure sequence $\mathcal{B}$, and $\mathcal{C}$ is a procedure sequence which exports the same interface as $\mathcal{B}$, then we define $\mathcal{AC}$ to be the procedure sequence where calls to $\mathcal{B}$ are executed as calls to $\mathcal{C}$.

2.2 Games

A game G consists of a main procedure with output in $\{0,1\}$. The main procedure is given access to an unspecified functionality and an unspecified procedure sequence called the adversaries; we denote this by $G_{\varphi}^{\mathcal{A}}$, where $\mathcal{A}$ denotes the adversaries and φ the unspecified functionality. By default the adversaries do not know any of the decisions or knowledge of the other adversaries. The game needs to explicitly specify which adversaries can communicate with each other. To this end, the game specifies for each pair of adversaries a storage procedure which exposes a hash table so that the pair may communicate with each other. The game may specify the amount of storage for each pair of adversaries ranging from none to unlimited. If the storage is limited and an adversary asks to store data exceeding the specified size, then the data is truncated and the remainder is ignored. These storage procedures are one way of formalizing the principle of resetting a simulator by a distinguisher as was proposed by Ristenpart et al. [15]: a reset corresponds to two simulators which may not communicate any information. Each adversary in $\mathcal{A}$ is given access to all procedures in $\mathcal{P}$ and all storage procedures involving it as part of the pair. The adversaries may not call other adversaries, nor any procedure in $\mathcal{H}$, nor any storage procedure with which it is not involved. All procedures in φ may call all other procedures in φ, but not the adversaries. Every procedure used in a game has a distinct state. The communications among the procedures is depicted in Figure 1.

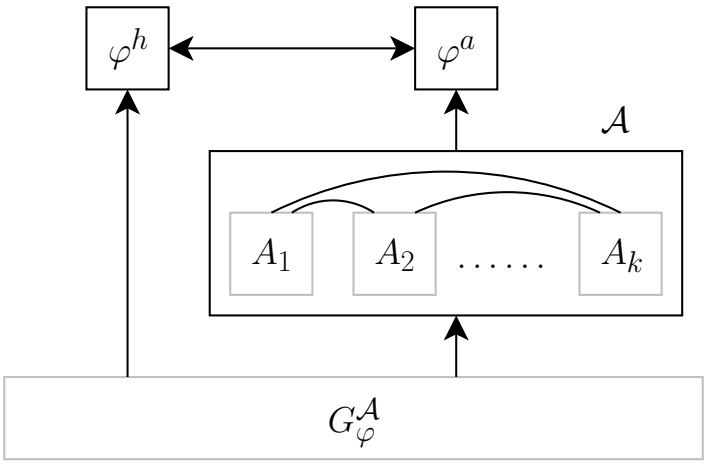

Fig. 1: Communications between the adversaries in $\mathcal{A}$ and the procedures in φ. Arrows denote access, and lines denote a shared storage procedure.

If we have a game G and adversaries $\mathcal{A}$ with an unspecified procedure sequence, and $\mathcal{B}$ is a procedure sequence which exports the same interface as the unspecified procedure sequence of $\mathcal{A}$, then we can define a game $H_{\varphi}^{\mathcal{B}} := G_{\varphi}^{\mathcal{A}\mathcal{B}}$, where the new main procedure is the main procedure of G along with the $\mathcal{A}$-calls absorbed, and the storage procedures of the new adversaries $\mathcal{B}$ are determined by looking at $\mathcal{A}\mathcal{B}$ and seeing how $\mathcal{A}$ allows $\mathcal{B}$ to communicate.

When we write $\mathbf{Pr}\left(G_{\varphi}^{\mathcal{A}} \to 1\right)$, we mean that the probability is taken over the random choices made by G, the adversaries in $\mathcal{A}$, and the functionality φ in order to determine how likely it is that the result of running the main procedure in $G_{\varphi}^{\mathcal{A}}$ results in an output of 1.

2.3 n-Games Versus Minimal n-Games

Although RSS [15] describe single-stage-games and multi-stage-games, and they discuss when games are equivalent, their discussion does not uniquely identify a single definition for the

concepts. A formal definition is however required for a good understanding of generalized notions of indifferentiability, and in this section we aim at such definition.

First, a minor detail, is a change from the use of the term *multi-stage-game* to *n-game*, where n is the number of adversaries.[1] In order to remove trivialities, we require that each of the n adversaries be called at least once by the n-game, since otherwise a game which only calls one adversary once could be looked at as a 2-game simply by adding an adversary which does nothing. In [15] RSS define equivalency of games as follows: for any fixed functionality φ and adversaries $\mathcal{A}$, two games are equivalent if

$$\mathbf{Pr}\left(G_{\varphi}^{\mathcal{A}} \to 1\right) = \mathbf{Pr}\left(H_{\varphi}^{\mathcal{A}} \to 1\right).$$

Note that $\mathcal{A}$ does not change, hence they do not discuss what happens if the number of adversaries changes. In fact, it is not obvious what is meant when an n-game is equivalent to an m-game for some $m < n$. For example, if one says that G and H are equivalent if there exist adversaries $\mathcal{A}$ and $\mathcal{B}$ such that $\mathbf{Pr}\left(G_{\varphi}^{\mathcal{A}} \to 1\right) = \mathbf{Pr}\left(H_{\varphi}^{\mathcal{B}} \to 1\right)$, then this misses the point of looking at equivalency of games in our case since G and H could be completely different, yet happen to output 1 with the same probability. There are other ways of changing n-games into "equivalent" m-games which are not useful for the discussion. For example, if you arbitrarily fix $n - m + 1$ adversaries in an n-game, then you have an m-game which is equivalent to the original n-game in some sense of the word. To unambiguously identify the amount of adversaries in a game, we explicitly formalize the notion of equivalency among games with respect to the amount of communication possible between the adversaries. Then we refer to a game as a minimal n-game if it is not equivalent to any m-game for $m < n$.

The basic idea behind our definition of a minimal n-game is that if you have a game with multiple adversaries and you allow some of the adversaries to share unlimited state, and this modification does not result in a change to the output of the game, then you can safely replace all of the adversaries with unlimited shared state by a single adversary, resulting in a game with fewer adversaries.

Let $n > 1$, let φ be a functionality and let G_{φ} be an n-game. Let s be a subset of $\{1, \ldots, n\}$, then we define H_{φ} to be the same as G_{φ}, except all storage procedures related to the pairs contained in s now have unlimited storage. Note that for all adversaries $\mathcal{A}$ it is the case that

$$\mathbf{Pr}\left(G_{\varphi}^{\mathcal{A}} \to 1\right) \leq \mathbf{Pr}\left(H_{\varphi}^{\mathcal{A}} \to 1\right).$$

Letting the pairs of adversaries at the positions specified in s have unlimited communication is the same as replacing all of them with one adversary, seeing as the only thing that separates adversaries is the amount of communication they may have. Hence if we define the game $\overline{G}_{\varphi}$ to be the same as G_{φ} except all adversaries with a position in s are replaced by a single adversary, we have that

$$\mathbf{Pr}\left(H_{\varphi}^{\mathcal{A}} \to 1\right) = \mathbf{Pr}\left(\overline{G}_{\varphi}^{\mathcal{B}} \to 1\right),$$

where $\mathcal{B}$ is a shortened version of $\mathcal{A}$ where all adversaries at the positions in s are replaced by a single adversary.

Definition 1. *Let $1 \leq m < n$. An n-game G_{φ} can be written as an m-game under φ if there exists a subset s of $\{1, \ldots, n\}$ such that $|s| = n - m + 1$ and for all sequences of adversaries $\mathcal{A}$ we have that*

$$\mathbf{Pr}\left(G_{\varphi}^{\mathcal{A}} \to 1\right) = \mathbf{Pr}\left(\overline{G}_{\varphi}^{\mathcal{B}} \to 1\right),$$

with $\overline{G}$ and $\mathcal{B}$ defined as above.

Let m be the smallest integer such that G_{φ} can be written as an m-game, then we say that G_{φ} is a minimal m-game.

[1] The authors believe that the word "stage" does not add anything meaningful to the description of the object.

Note that if an n-game G_φ is a minimal m-game, then G_φ can be written as an i-game for any $m \leq i \leq n$: one applies the above trick to $n - i + 1$ adversaries.

Example 1. Below we show an example of a two-game which was introduced in [15]. When we call an adversary, we write in superscript the procedures that the adversary has access to. The storage procedure between A_1 and A_2 is denoted by st_n and is limited to n bits. Let $\varphi := (\mathcal{H}, \mathcal{P})$.

procedure $CRP_\varphi^{A_1,A_2}(p, n, s)$
  $m \xleftarrow{\$} \{0,1\}^p$
  $A_1^{\mathcal{P},st_n}(m)$
  $c \xleftarrow{\$} \{0,1\}^s$
  $z \xleftarrow{\$} A_2^{\mathcal{P},st_n}(c)$
  return $z = \mathcal{H}(m\|c)$
end procedure

We consider the case where $\mathcal{H}$ is a domain extender and $\mathcal{P}$ is an ideal primitive. Note that when $n < p$ the game is really a *minimal* two-game. Say that A_1 is an adversary which simply stores m and A_2 is an adversary that calculates $\mathcal{H}(m'\|c)$ using $\mathcal{P}$, where m' is the value A_2 gets from querying st_n for m. If $n < p$ then the game outputs 1 with very small probability since the storage procedure truncates the message m, whereas if storage is unlimited, then the game outputs 1 with probability 1. Thus, in terms of Definition 1, we have $\mathbf{Pr}\left(G_\varphi^{\mathcal{A}} \to 1\right) < \mathbf{Pr}\left(\overline{G}_\varphi^{\mathcal{B}} \to 1\right)$ and the game is thus a minimal two-game.

If $n \geq p$ then we still cannot say it is a minimal one-game. The reason for this is that a particular choice of $\mathcal{A}$ may cause the strict inequality $\mathbf{Pr}\left(G_\varphi^{\mathcal{A}} \to 1\right) < \mathbf{Pr}\left(\overline{G}_\varphi^{\mathcal{B}} \to 1\right)$. For instance, A_1 could generate some random message m^* of length n and store $m^*\|m$ so that A_2 receives no useful information when the storage is limited, but can always win when the storage is unlimited.

3 A Different View of Indifferentiability

The goal of this section is to explain indifferentiability as just one way of achieving our original aim of composition. In the process we will prove the MRH theorem from [12], and indicate where exactly the application of the MRH theorem to the CRP game fails with certain domain extenders. Additionally, we will demonstrate that beyond the notions of regular indifferentiability and reset-indifferentiability, there is a whole spectrum of indifferentiability definitions that one could consider.

A functionality φ_1 is at least as secure as a functionality φ_2 if for all games G and adversaries $\mathcal{A}$, there exists a game $\mathcal{H}$ and adversary $\mathcal{B}$ such that the quantity

$$\mathbf{Pr}\left(G_{\varphi_1}^{\mathcal{A}} \to 1\right) - \mathbf{Pr}\left(H_{\varphi_2}^{\mathcal{B}} \to 1\right)$$

is small. Note that this definition differs from the one given by MRH in [12] in that we do not require that $H = G$. This definition remains sufficient in order to achieve composition, since as long as $\mathbf{Pr}\left(H_{\varphi_2}^{\mathcal{B}} \to 1\right)$ is small, $\mathbf{Pr}\left(G_{\varphi_1}^{\mathcal{A}} \to 1\right)$ will be small.

So given G and $\mathcal{A}$, the definition searches for an H and $\mathcal{B}$. If we define $H := G$, create $\mathcal{S}$, define $\mathcal{B} := \mathcal{A}\mathcal{S}$, and set $D := G^{\mathcal{A}}$, then we get

$$\mathbf{Pr}\left(G_{\varphi_1}^{\mathcal{A}} \to 1\right) - \mathbf{Pr}\left(H_{\varphi_2}^{\mathcal{B}} \to 1\right) = \mathbf{Pr}\left(D_{\varphi_1}^{\mathcal{L}} \to 1\right) - \mathbf{Pr}\left(D_{\varphi_2}^{\mathcal{S}} \to 1\right),$$

where $\mathcal{L}$ is a procedure which just exposes the adversarial interface to φ_1. This is exactly what indifferentiability is, and we have proved that indifferentiability gives us that φ_1 is at least

as secure as φ_2. Note that indifferentiability does not place any requirements on the games or procedures. In fact, in this point of view indifferentiability is mostly a rewording of the definition of "φ_1 is at least as secure as φ_2". Therefore it is not possible that indifferentiability nor the MRH theorem do not apply to the CRP game.

3.1 $\mathcal{G}$-Indifferentiability

What has occurred is that all existing indifferentiability proofs actually limit the types of games that are covered by limiting the powers of the distinguisher. Since the distinguishers are taken to be one-games, composition holds at best for minimal one-games. In fact, the types of games covered will only be as powerful as the distinguishers. For example, a one-game distinguisher cannot model all two-games, yet a minimal two-game distinguisher would be able to do such a thing.

One can look at different types of indifferentiability where the distinguishers are given varying degrees of power. Let $\mathcal{G}$ be a class of games; $\mathcal{G}$ could for example be all minimal one-games, all minimal two-games, or all minimal one-games along with CRP. What we are really interested in is when we may reduce functionality φ_1 to functionality φ_2 in the class $\mathcal{G}$, or in other words, when is φ_1 $\mathcal{G}$-indifferentiable from φ_2? So we pick a class of distinguishers, and then we need to show that every game in $\mathcal{G}$ can be written as a distinguisher as we have defined it. For example, if we take distinguishers as they have always been used in regular indifferentiability proofs, then we notice that they are all one-games: the adversaries (simulators) are allowed unlimited communication. Conversely, if we take an arbitrary one-game $G_{\varphi}^{\mathcal{AL}}$ (with respect to the adversaries $\mathcal{L}$) and we write it as allowing all adversaries in $\mathcal{L}$ to have unlimited communication, then we have that $G_{\varphi}^{\mathcal{AL}}$ is a one-game distinguisher. Hence we have shown that $\mathcal{G}$-indifferentiability where $\mathcal{G}$ is the class of one-games, corresponds exactly to regular indifferentiability.

So proving $\mathcal{G}$-indifferentiability for an arbitrary class $\mathcal{G}$ comes down to characterizing what the games in $\mathcal{G}$ are allowed to do with the functionalities and adversaries. In fact, without this characterization, it is not immediately clear for what class of games one would be proving indifferentiability. For example, reset-indifferentiability allows the distinguisher to reset as many times as it wants, yet the attack on online computable domain extenders in [15] only uses one reset, so it is interesting to see what games we may cover if we restrict the distinguisher to use only one reset, i.e. single-reset-indifferentiability. We know that single-reset-indifferentiability cannot cover any minimal three-games or higher, and we know that a one-game distinguisher already covers all one-games, hence we focus our attention on minimal two-games. Consider the following minimal two-game, where A_1 and A_2 may communicate up to n bits:

procedure $G_{\varphi}^{A_1,A_2}(p,n)$
  $m_1 \| m_2 \| m_3 \xleftarrow{\$} \{0,1\}^{3p}$
  $A_1^{\mathcal{P},st_n}(m_1)$
  $A_2^{\mathcal{P},st_n}(m_2)$
  $z \xleftarrow{\$} A_1^{\mathcal{P},st_n}(m_3)$
  return $z = \mathcal{H}(m_1 \| m_2 \| m_3)$
end procedure

Here the storage procedure is denoted by st_n. Any distinguisher which has to model the above game must be able to reset at least twice if each A_1 and A_2 call contains an $\mathcal{H}$ call. A distinguisher must reset once after $A_1(m_1)$ and once after $A_2(m_2)$, otherwise it would violate the fact that A_1 and A_2 do not have unlimited communication. Hence we have an example of a minimal two-game which cannot be modeled by a single-reset distinguisher, and so single-reset-indifferentiability does not cover all minimal two-games. In fact, using similar reasoning,

we see that a single-reset distinguisher is only able to mimic minimal two-games where all calls to the first adversary happen before all calls to the second adversary occur. So it is important to understand what types of games a given distinguisher is able to cover.

4 Impossibility Of Reset-Indifferentiability

In this section we show that it is impossible to create a domain extender which is reset-indifferentiable from a random oracle. This is not surprising as the existence of a domain extender which is reset-indifferentiable from a random oracle would mean that such a domain extender would have to survive being reset an arbitrary number of times. The result in this section is in fact a generalization of the attack of RSS [15], and is included as a simple illustration.

The attack uses the notions of min-entropy H_∞ and average min-entropy $\tilde{H}_\infty$. Readers not familiar with these two notions can find definitions and basic facts in Appendix B.

Let $F : \{0,1\}^M \to \{0,1\}^H$ be a domain extender that uses the sequence of ideal primitives $\pi = (\pi_1, \ldots, \pi_k)$; let $\varphi_1 := (F, \pi)$ be the corresponding functionality. For an input message m of length M, we denote the sequence of primitive calls made by F for the evaluation of m with $(p_1, p_2, \ldots, p_n)$, where

$$p_i : U_i \to V_i \,.$$

Note that n and the sequence in which the π_i are called could depend on the input message, and that two different primitive calls could possibly be the same primitive. For example, Liskov's zipper hash [11] uses two primitives, π_1 and π_2: p_1 through $p_{n/2}$ would be π_1-calls whereas $p_{n/2+1}$ to p_n would be π_2-calls.

Theorem 1. *Let $\varphi_2 := (RO, RO)$ be a functionality where both the adversarial and honest interfaces expose the random oracle RO with range $\{0,1\}^H$. Then there exists an n-game distinguisher D such that for all simulators $\mathcal{S} = (S_1, \ldots, S_n)$,*

$$\left|\mathbf{Pr}\left(D^{\mathcal{L}}_{\varphi_1} \to 1\right) - \mathbf{Pr}\left(D^{\mathcal{S}}_{\varphi_2} \to 1\right)\right| \geq 1 - \left(\sum_{i=1}^{n} \frac{q_i}{2^{M-\log|U_i|}} + \frac{1}{2^H}\right),$$

where $\mathcal{S}_i$ makes q_i queries to RO, $\sum_{i=1}^n q_i \leq q$ and $\mathcal{L}$ is a sequence of procedures where $L \in \mathcal{L}$ returns the value given by $\eta(L)$.

We stress that q here denotes the number of queries made by $\mathcal{S}$; the distinguisher D makes $n+1$ queries. In order to illustrate the bound, if $p_1 = \ldots = p_n = \pi : \{0,1\}^a \to \{0,1\}^b$ with $H \leq b$, as is the case with many existing (narrow- and wide-pipe) domain extenders, then we get

$$\left|\mathbf{Pr}\left(D^{\mathcal{L}}_{\varphi_1} \to 1\right) - \mathbf{Pr}\left(D^{\mathcal{S}}_{\varphi_2} \to 1\right)\right| \geq 1 - \left(\frac{q}{2^{M-a}} + \frac{1}{2^H}\right),$$

which can be made arbitrarily close to $1 - 1/2^H$, as M can be increased.

Notice that we specify that the distinguisher can be a multi-game and does not have to be a one-game. This is equivalent to saying that the distinguisher may reset an arbitrary number of times.

Proof. The distinguisher D requires a functionality $\varphi = (\mathcal{H}, \mathcal{P})$, where $\mathcal{H}$ exports the same interface as F and $\mathcal{P}$ exports the same interface as $(\pi_1, \ldots, \pi_k)$. Furthermore D requires a sequence $\mathcal{A}$ of n adversaries, with n being the number of primitive calls, and if $A_i \in \mathcal{A}$ then $\eta(A_i) = p_i$. We use the fact that the distinguisher may be a multi-game by letting all storage procedures have zero storage.

First the distinguisher selects a message m uniformly at random from $\{0,1\}^M$. Then D copies what F does, except a call to p_i is replaced by a call to A_i. The distinguisher outputs its result z which it then compares with $\mathcal{H}(m)$: D returns 1 when z equals $\mathcal{H}(m)$, and 0 otherwise. Note that

$$\mathbf{Pr}\left(D^{\mathcal{L}}_{(F,\pi)} \to 1\right) = 1\,.$$

Now we look at $D^{\mathcal{S}}_{(RO,RO)}$. If $\mathcal{H}(m)$ is never called by S_i for all i then z is independent of $\mathcal{H}(m)$, hence the chance that z equals $\mathcal{H}(m)$ is at most $1/2^H$. Let E_i denote the event that S_i queries $\mathcal{H}(m)$, then

$$\begin{aligned}\mathbf{Pr}\left(D^{\mathcal{S}}_{(RO,RO)} \to 1\right) &\leq \mathbf{Pr}\left(\cup_i E_i\right) + \mathbf{Pr}\left(D^{\mathcal{S}}_{(RO,RO)} \to 1 \mid \neg(\cup_i E_i)\right)\\ &\leq \sum_{i=1}^{n} \mathbf{Pr}\left(E_i\right) + \frac{1}{2^H}\,.\end{aligned}$$

Each input given to S_i, u_i, can be considered a random variable over U_i. At call i all the information that the simulator knows is u_i since no communication is allowed between the S_i and each S_i is only called once. This means that the probability that S_i can compute m is bounded above by $2^{-\tilde{H}_\infty(m|u_i)}$. We know that $\tilde{H}_\infty(m \mid u_i) \geq H_\infty(m) - \log|U_i| = M - \log|U_i|$ (cf. Appendix B), therefore

$$\mathbf{Pr}\left(E_i\right) \leq \frac{q_i}{2^{M-\log|U_i|}}\,,$$

and we have our desired result. □

5 Impossibility Of Single-Reset-Indifferentiability

We present the main impossibility result of this work, namely that there are no "meaningful" domain extenders which are single-reset-indifferentiable from a random oracle. What we mean by "meaningful" will be explained below.

Just as the result of Section 4, the attack of this section uses the notions of min-entropy H_∞ and average min-entropy $\tilde{H}_\infty$. Readers not familiar with these two notions can find definitions and basic facts in Appendix B.

Let $F : \{0,1\}^M \to \{0,1\}^H$ be a domain extender which uses some primitives $\pi = (\pi_1, \ldots, \pi_k)$. On a given input $m \in \{0,1\}^M$, the domain extender calls its primitives in a certain order: $p_1, p_2, \ldots, p_n$. Write $p_i : U_i \to V_i$. Note that the p_i are called primitives, not the primitives themselves, hence different p_i could be the same primitive. Let φ_1 denote the functionality representing F and its primitives.

We require that the internal state of F does not exceed N bits. Concretely this means that for all j, no more than N bits of the outputs of $p_1, \ldots, p_j$ are used in the inputs to $p_{j+1}, \ldots, p_n$. For all $j < n$ we can talk of the bits of m used in the inputs to $p_1, \ldots, p_j$, represented by l_j, and the bits of m used in the inputs to $p_{j+1}, \ldots, p_n$, or r_j. Let a_j be the bits of overlap between the two sides, so that $|l_j| + |r_j| - |a_j| = M$ (here we assume that all of m is used in the inputs, otherwise the domain extender would be easily differentiable from a random oracle).

We additionally restrict F such that modifying a bit of the input message m during computation returns $F(m)$ with low probability. Concretely, let ε be such that for all messages m, and for all j, if a bit chosen at random is flipped from a_j, and the result is computed as z, then $\mathbf{Pr}\left(F(m) = z\right)) < \varepsilon$. Our restriction is that ε is sufficiently close to 0. Note that this restriction is highly related to the well-known "avalanche effect", a basic property that practical hash functions are required to satisfy.

Theorem 2. *Let $\varphi_2 := (RO, RO)$ be a functionality where both the adversarial and honest interfaces expose the random oracle RO with range $\{0,1\}^H$. Then there exists a single-reset distinguisher D such that for all simulators,*

$$\left|\mathbf{Pr}\left(D^{\mathcal{L}}_{\varphi_1} \to 1\right) - \mathbf{Pr}\left(D^{\mathcal{S}}_{\varphi_2} \to 1\right)\right| \geq \min\left\{1 - \frac{1}{2^H} - \frac{q}{2^{M/4-2}}, \frac{3}{8} - \frac{3}{M} - \frac{\varepsilon}{2}\right\},$$

where q is the total number of queries that $\mathcal{S}$ makes to RO and $\mathcal{L}$ is a sequence of procedures with $L \in \mathcal{L}$ returning the value given by $\eta(L)$.

Proof. Our distinguisher D is a combination of two distinguishers P and Q which take as input a message m and an index j. Intuitively, D runs P if the domain extender is roughly single-pass, and it uses Q for domain extender which re-use a significant part of the message. We will describe the distinguishers in terms of the unspecified functionality $\varphi = (\mathcal{H}, \mathcal{P})$. The functionality $\varphi_1 = (F, \pi)$ represents the so-called "real world", and the functionality $\varphi_2 = (RO, RO)$ the "simulated world".

Distinguisher P. The distinguisher P goes through the process of computing with m just as the domain extender does: it performs all of the primitive calls $p_1, \ldots, p_n$. The only difference is that it resets after p_j. Eventually it arrives at some final result z and it outputs 1 if z is equal to $\mathcal{H}(m)$ and 0 otherwise.

Note that this distinguisher always outputs one in the real world. In the simulated world we have that z is independent of $\mathcal{H}(m)$ unless the simulator queries $\mathcal{H}(m)$. Let E be the event that the simulator queries $\mathcal{H}(m)$. Then

$$\begin{aligned}\mathbf{Pr}\left(P^{\mathcal{S}}_{\varphi_2} \to 1\right) &\leq \mathbf{Pr}\left(P^{\mathcal{S}}_{\varphi_2} \to 1 \mid \neg E\right) + \mathbf{Pr}\left(E\right) \\ &\leq \frac{1}{2^H} + \mathbf{Pr}\left(E\right) .\end{aligned}$$

By basic properties of entropy (cf. Appendix B), the probability that $\mathcal{S}$ queries $\mathcal{H}(m)$ before the reset is bounded above by $q/2^{H_\infty(m)-|l_j|}$ and the probability that $\mathcal{S}$ queries $\mathcal{H}(m)$ after the reset is bounded above by $q/2^{H_\infty(m)-N-|r_j|}$, where q is the total number of queries that the simulator makes to RO. Hence we get

$$\mathbf{Pr}\left(P^{\mathcal{S}}_{\varphi_2} \to 1\right) \leq \frac{1}{2^H} + \frac{q}{2^{H_\infty(m)-|l_j|}} + \frac{q}{2^{H_\infty(m)-N-|r_j|}} .$$

Distinguisher Q. Just like P, the distinguisher Q goes through the process of computing m like the domain extender and resets after p_j. The main difference is that if $|a_j| > 0$, then with probability one half it will flip one of the overlapping bits of m and continue the computation after the reset with the modified m'. If $|a_j| = 0$, we define Q to simply return 0 (as becomes clear later, D will by construction run Q only if $|a_j| > 0$). More formally, Q will take $b \xleftarrow{\$} \{0,1\}$, and if $b = 1$, it also takes $c \xleftarrow{\$} \{1, \ldots, |a_j|\}$. If $b = 1$, m' equals m with the cth bit flipped, whereas if $b = 0$, $m' = m$. It proceeds the second half of the computation with the modified m' and eventually arrives at some final result z. Now, Q outputs 1 if $b = 0 \wedge z = \mathcal{H}(m)$ or if $b = 1 \wedge z \neq \mathcal{H}(m)$.

Denote by B the event that $b = 1$ (or equivalently, that a bit flip occurred).

In the real world, $\neg B$, z equals $\mathcal{H}(m)$. On the other hand given B, $z \neq \mathcal{H}(m)$ with probability at least $1 - \varepsilon$ (see introduction). We find

$$\mathbf{Pr}\left(Q^{\mathcal{L}}_{\varphi_1} \to 1\right) \geq \frac{1 + 1 - \varepsilon}{2} = 1 - \frac{\varepsilon}{2} .$$

Let U denote the event that simulator after the reset learns c. Then, given that B does not occur then U occurs with probability zero. If B occurs, U occurs with probability at most $\frac{N}{|a_j|}$: indeed, the simulator learns the adjusted a_j, and at most N bits of information of the original one. We can write

$$\begin{aligned}\mathbf{Pr}\left(Q^{\mathcal{S}}_{\varphi_2} \to 1\right) &\leq \mathbf{Pr}\left(U\right) + \mathbf{Pr}\left(Q^{\mathcal{S}}_{\varphi_2} \to 1 \mid \neg U\right)\left(1 - \mathbf{Pr}\left(U\right)\right) \\ &\leq \frac{N}{2\left|a_j\right|} + \mathbf{Pr}\left(Q^{\mathcal{S}}_{\varphi_2} \to 1 \mid \neg U\right)\left(1 - \frac{N}{2\left|a_j\right|}\right),\end{aligned}$$

where we use that if $x \leq z$ and $y \leq 1$, $x + y(1-x) \leq z + y(1-z)$.

Let E denote the event that the simulator learns a_j. We have

$$\mathbf{Pr}\left(Q^{\mathcal{S}}_{\varphi_2} \to 1 \mid \neg U\right) \leq \mathbf{Pr}\left(E \mid \neg U\right) + \mathbf{Pr}\left(Q^{\mathcal{S}}_{\varphi_2} \to 1 \mid \neg E \wedge \neg U\right)\left(1 - \mathbf{Pr}\left(E \mid \neg U\right)\right).$$

Given that U does not occur, the probability of E is bounded above by $\frac{1}{|a_j|-N}$: indeed, the simulator learns the adjusted a_j, and at most N bits of information of the original one, which due to $\neg U$ rules out N possible values from which c is distributed. If the simulator does not learn a_j and U does not occur, then the output of the simulator must be independent of B, therefore $\mathbf{Pr}\left(Q^{\mathcal{S}}_{\varphi_2} \to 1 \mid \neg E \wedge \neg U\right) = \frac{1}{2}$. Putting all of the results together, we get

$$\mathbf{Pr}\left(Q^{\mathcal{S}}_{\varphi_2} \to 1\right) \leq \frac{1}{2}\left(1 + \frac{1}{|a_j| - N}\right) + \frac{N}{4\left|a_j\right|}\left(1 - \frac{1}{|a_j| - N}\right).$$

Distinguisher D. Now, we define a distinguisher D which picks a message m uniformly at random from $\{0,1\}^M$, where M is taken such that $M \geq 8N$ and $M \geq 8\max_i U_i$.

If there exists a $j \in \{1, \ldots, n\}$ such that $|a_j| > \frac{1}{4}M$, then D runs Q on input of j and m. In this case, the advantage of D is lower bounded by

$$\frac{3}{8} - \frac{3}{M} - \frac{\varepsilon}{2}.$$

Otherwise, suppose for all j we have $|a_j| \leq \frac{1}{4}M$. Let j be maximal such that $|l_j| \leq \frac{3}{4}M$. We claim that also $|r_j| + N \leq \frac{3}{4}M$. Indeed, as j is maximal, we have $\frac{3}{4}M \leq |l_{j+1}| \leq |l_j| + \max_i U_i$, and thus

$$|r_j| + N = M - |l_j| + |a_j| + N \leq M - \frac{3}{4}M + \max_i U_i + \frac{1}{4}M + N \leq \frac{3}{4}M.$$

Hence the claim. Now, D runs P on input of j and m. Note that the entropy of m could reduce by 1 bit due to the fact that the distinguisher is chosen based on m. By virtue of the distinguishers P, the advantage of D is lower bounded by

$$1 - \frac{1}{2^H} - \frac{q}{2^{M/4-2}},$$

which completes the proof. □

6 Conclusions and Future Work

As we have seen, the indifferentiability framework comes with a range of subtleties which make it difficult to establish composition for a wide variety of games. This was already clear since the work of Ristenpart et al. [15], who introduced the generalized definition of reset-indifferentiability. But, as we have shown, there does not exist any domain extender that is reset-indifferentiable, and even if one opts for single-reset-indifferentiability, there is no hope

to find a secure domain extender. These dead-ends in terms of domain extenders and random oracles, however, do not lead us to question the usefulness of indifferentiability, rather it points out the disconnect between what distinguishers are and the classes of games one is considering. Take for example, the fact that a game G and its adversaries $\mathcal{A}_1$ are merged to form a single distinguisher. This creates the unnatural effect that G and $\mathcal{A}_1$ are actually cooperating in differentiating one functionality from another, whereas usually G and $\mathcal{A}_1$ are designed with opposing goals (hence the name "adversary"). If we look at the CRP game defined earlier on, we see that the task of the adversaries is complicated by the fact that they may only communicate a finite amount of bits, and as a result neither adversary gets to see the full message. Generalizing this, one could consider all games which do not provide their adversaries with a complete message and limited communication, and see where this type of indifferentiability would lead.

Ultimately the important part is having an understanding of the natural classes of games that exist and are easy to describe. Finally, an alternative approach is to consider the same classes of games, except with different functionalities other than random oracles, as is done in [13].

Acknowledgments. This work has been funded in part by the IAP Program P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II. The second author is supported by a Ph.D. Postdoctoral Fellowship from the Flemish Research Foundation (FWO-Vlaanderen). The third author is supported by a Ph.D. Fellowship from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

[1] Andreeva, E., Mennink, B., Preneel, B.: On the indifferentiability of the Grøstl hash function. In: Security and Cryptography for Networks 2010. Lecture Notes in Computer Science, vol. 6280, pp. 88–105. Springer-Verlag, Berlin (2010)

[2] Bellare, M., Ristenpart, T.: Multi-property-preserving hash domain extension and the EMD transform. In: Advances in Cryptology - ASIACRYPT 2006. Lecture Notes in Computer Science, vol. 4284, pp. 299–314. Springer-Verlag, Berlin (2006)

[3] Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM Conference on Computer and Communications Security. pp. 62–73. ACM, New York (1993)

[4] Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: Advances in Cryptology - EUROCRYPT '94. Lecture Notes in Computer Science, vol. 839, pp. 92–111. Springer-Verlag, Berlin (1994)

[5] Bellare, M., Rogaway, P.: The exact security of digital signatures - how to sign with RSA and Rabin. In: Advances in Cryptology - EUROCRYPT '96. Lecture Notes in Computer Science, vol. 1109, pp. 399–416. Springer-Verlag, Berlin (1996)

[6] Bertoni, G., Daemen, J., Peeters, M., Assche, G.: On the indifferentiability of the sponge construction. In: Advances in Cryptology - EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 181–197. Springer-Verlag, Berlin (2008)

[7] Bhattacharyya, R., Mandal, A., Nandi, M.: Security analysis of the mode of JH hash function. In: Fast Software Encryption 2010. Lecture Notes in Computer Science, vol. 6147, pp. 168–191. Springer-Verlag, Berlin (2010)

[8] Coron, J., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård revisited: How to construct a hash function. In: Advances in Cryptology - CRYPTO 2005. Lecture Notes in Computer Science, vol. 3621, pp. 430–448. Springer-Verlag, Berlin (2005)

[9] Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging Merkle-Damgård for practical applications. In: Advances in Cryptology - EUROCRYPT 2009. Lecture Notes in Computer Science, vol. 5479, pp. 371–388. Springer, Heidelberg (2009)

[10] Hirose, S., Park, J., Yun, A.: A simple variant of the Merkle-Damgård scheme with a permutation. In: Advances in Cryptology - ASIACRYPT 2007. Lecture Notes in Computer Science, vol. 4833, pp. 113–129. Springer-Verlag, Berlin (2007)

[11] Liskov, M.: Constructing an ideal hash function from weak ideal compression functions. In: Selected Areas in Cryptography 2006. Lecture Notes in Computer Science, vol. 4356, pp. 358–375. Springer-Verlag, Berlin (2007)

[12] Maurer, U., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Theory of Cryptography Conference 2004. Lecture Notes in Computer Science, vol. 2951, pp. 21–39. Springer-Verlag, Berlin (2004)
[13] Naito, Y.: On the indifferentiable hash functions in the multi-stage security games. Cryptology ePrint Archive, Report 2012/014 (2012)
[14] Naito, Y., Yoneyama, K., Wang, L., Ohta, K.: How to confirm cryptosystems security: The original Merkle-Damgård is still alive! In: Advances in Cryptology - ASIACRYPT 2009. vol. 5912, pp. 382–398. Springer-Verlag, Berlin (2009)
[15] Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with composition: Limitations of the indifferentiability framework. In: Advances in Cryptology - EUROCRYPT 2011. Lecture Notes in Computer Science, vol. 6632, pp. 487–506. Springer-Verlag, Berlin (2011)
[16] Yoneyama, K., Miyagawa, S., Ohta, K.: Leaky random oracle (extended abstract). In: Provable Security 2008. Lecture Notes in Computer Science, vol. 5324, pp. 226–240. Springer-Verlag, Berlin (2008)

A Translation Between MRH and RSS Terminology

Table 1: Translation between MRH and RSS terminology.

MRH	**RSS**
Random System	Procedure
Environment	Game
Cryptosystem/Resource	Functionality
Public interface	Adversarial interface
Private interface	Honest interface

B Basic Properties of Entropy

For the purpose of the impossibility results of Sections 4 and 5, we present some basic properties of entropy.

Definition 2. *The min-entropy of a random variable x is $H_\infty(x) = -\log(\max_{x'} \mathbf{Pr}\left(x = x'\right))$.*

Definition 3. *When x and y are two (possibly correlated) random variables the average min-entropy is given by*

$$\tilde{H}_\infty(x \mid y) = -\log\left(\sum_{y'}\left(\max_{x'} \mathbf{Pr}\left(x = x' \mid y = y'\right)\right)\mathbf{Pr}\left(y = y'\right)\right).$$

The probability that an adversary guesses the value of x given y is bounded above by $2^{-\tilde{H}_\infty(x|y)}$. Furthermore, if a random variable y can take on n possible values, then

$$\begin{aligned}\sum_{y'}\left(\max_{x'} \mathbf{Pr}\left(x = x' \mid y = y'\right)\right)\mathbf{Pr}\left(y = y'\right) &= \sum_{y'}\max_{x'} \mathbf{Pr}\left(x = x' \cap y = y'\right) \\ &\leq n \max_{x'} \mathbf{Pr}\left(x = x'\right),\end{aligned}$$

and so $\tilde{H}_\infty(x \mid y) \geq H_\infty(x) - \log n$.