Resource-Restricted Indifferentiability

Grégory Demay[1], Peter Gaži[1,2], Martin Hirt[1], and Ueli Maurer[1]

[1] Department of Computer Science, ETH Zurich, Switzerland
{gregory.demay,peter.gazi,hirt,maurer}@inf.ethz.ch
[2] Department of Computer Science, Comenius University, Bratislava, Slovakia

Abstract. The notion of indifferentiability was introduced in [27] and in [14] it was tailored for security analysis of hash function constructions, making indifferentiability from a random oracle the desired property for any hash function design. However, the widely accepted view that a construction enjoying such a proof with an underlying ideal compression function can replace the random oracle in any application without compromising security is not justified in certain settings, as pointed out recently in [29].
In this paper we argue that one general reason for such a failure is the inflexibility of the indifferentiability notion with respect to more complex restrictions on resources (such as memory, randomness) available to the attacker: Typically, the distinguisher and the simulator in an indifferentiability statement are only required to be PPT algorithms, implicitly posing a polynomial restriction also on the resources available to them. We argue that this is not sufficient in certain scenarios and explain why this is the problem underlying the security breakdown described in [29]. We present a systematic treatment of such settings by proposing a more fine-grained notion of *memory-aware reducibility* that is necessary in contexts where memory is the resource that requires a more detailed quantification.
We employ this new formalism to prove a lower bound on the memory required by any simulator in a domain extension construction of a public random function. Our results imply that if we restrict to simulators without memory, even domain extension by a single bit becomes impossible. On the other hand, for the infinite extension from an ideal compression function to a random oracle, a memory roughly linear in the total sum of the lengths of all queries is required. This solves an open problem given in [29].
Finally, it follows from our results that for any multi-party setting where one cannot assume the existence of a central adversary and hence it requires to be modeled using an independent local simulator for each party, it is impossible to securely construct a public random oracle from a public ideal compression function.

1 Introduction

1.1 The Indifferentiability Framework

Certain cryptographic constructions are used (and hence have to be analyzed) in settings where some access to their internal state is available publicly, making it also within reach of any potential attacker. The notion of *indifferentiability* was introduced in [27] to serve as a generalization of indistinguishability tailored for analyzing exactly this type of constructions. It also comes with a composition theorem formalizing the important property that an ideal primitive can be replaced by an indifferentiable construction in any context.

The indifferentiability framework found its most important application in the analysis of hash function constructions. Many existing cryptographic constructions were proven to be secure in the random oracle model (ROM) [5]; however, it is well-known that random oracles do not exist [10]. Therefore, one has to instantiate the random oracle by an existing cryptographic hash function H and the proof in the ROM can then only be taken as a heuristic argument towards the security of the overall construction. However, if one uses a hash function construction H^f that was proven indifferentiable from a random oracle when using an ideal compression function f, this excludes any possible attacks exploiting the structure of H and reduces the security of the construction to the security of the underlying compression function f [14]. This is a much more compact object and is comparatively simpler to analyze. As a consequence, an indifferentiability proof in the setting with an ideal compression function is generally considered an important argument towards the security of a practical hash function design and many

of the SHA-3 candidates (including the winner Keccak [7]) enjoy such a proof (see e.g. [8,13,16,17,3]). Moreover, the indifferentiability framework offers a precise formalism for studying the security of reductions of the random oracle to the ideal compression function, which is an interesting theoretical question by itself.

1.2 Limitations of Indifferentiability

Indifferentiability as well as indistinguishability comes in two flavours, namely *information-theoretic* and *computational*. In the information theoretic variant, there are no restrictions posed on the resources available to the distinguisher and the simulator, while in the computational variant, they are both bound to be efficient. The typical specification of an efficient algorithm is by probabilistic polynomial time (PPT), which implies also polynomial memory and randomness limitations. This way of restricting adversarial resources is also present in other simulation-based security notions such as universal composability [12] and reactive simulatability [4].

These two degrees of resource restrictions are sufficient for most natural settings and hence indifferentiability results have a wide scope of applicability. However, there are practical settings where this rough approach is not sufficient and a more fine-grained analysis is needed. One such scenario was recently exhibited in [29] in the context of auditable storage. Before we introduce our contributions, let us briefly review this example.

Storage-Auditing Scenario from [29]. The central motivating example in [29] is a two-party protocol for verification of storage. Its goal is to allow the first party (the user) to verify that the second party (the server – e.g. a storage service) is properly storing a certain piece of data that the user has provided earlier. The protocol is using a hash function and as long as it is modeled as a random oracle, it is clearly impossible for a malicious server to pass the verification without actually storing the user's data. However, it was observed in [29] that this is no longer true if the random oracle is replaced by a particular iterative construction with an underlying ideal compression function, even though this construction is known to be indifferentiable from a random oracle. This puts in question the meaning of an indifferentiability proof as a security argument relevant in all possible contexts.

We argue that the best way to understand this example is to consider in greater detail the memory requirements of the simulator used in the indifferentiability proof in question. The simulator is modeled as a PPT algorithm, guaranteeing that the real implementation is at least as good as the ideal random oracle as long as the attacker is capable of performing the tasks modeled by the simulator, in particular has polynomial amount of memory available. However, this is an unacceptable assumption if we want to investigate whether the server can pass the verification procedure *without* allocating all the memory required to store the user's message. As a side contribution, we give a more detailed explanation of this problem in Appendix A, illustrating why one cannot expect the indifferentiability statement to apply to this setting.

1.3 Contributions of this Paper

It this paper we show a general approach to treating indifferentiability scenarios comprising resource restrictions such as the one given in [29]. Choosing memory as the particular resource to consider, we model indifferentiability statements with greater focus on memory requirements of the simulators.

Memory-Aware Reducibility. In Section 3 we introduce the notion of *memory-aware reducibility* that is derived from reducibility in the classical indifferentiability setting as given in [27,26], but does not allow the memory requirements of the simulator to be "swept under the rug", requiring only that they are polynomial. It requires any memory necessary for the simulator to be explicitly modeled as a part of the ideal primitive; with the intuitive meaning that the real construction is provably as good as

the ideal one as long as we assume that the adversary has the necessary amount of memory available. We also prove a composition theorem for this new notion.

Note that this does not render the original notion of indifferentiability incorrect or obsolete. A (computational) indifferentiability statement never aspired to give any guarantees on the memory requirements of the simulator beyond its efficiency (and hence polynomial memory). Therefore, in situations where such a guarantee is not sufficient (as in the scenario from [29]), one has to resort to a more detailed modeling of the resources involved, as we demonstrate here.

An independent approach to analyzing the complexity of the simulator in an indifferentiability statement appeared recently in [18], where the authors focus on the number of queries the simulator issues per one invocation. To the best of our knowledge, our work is the first one pointing out the importance of the simulator's memory requirements.

Simulator Memory for Domain Extension. In Section 4 we look at the most important application of indifferentiability: the question of *domain extension* for public random functions. More precisely, we consider constructions that can be used to obtain an arbitrary input-length random oracle $\mathbf{R}^{*,n}\colon \{0,1\}^* \to \{0,1\}^n$ from an ideal compression function $\mathbf{R}^{m,r}\colon \{0,1\}^m \to \{0,1\}^r$ in an indifferentiable way, such as the various variants of the Merkle-Damgård construction proposed in [14]. We also consider the question of finite domain extension, i.e., constructing $\mathbf{R}^{\ell,r}$ from $\mathbf{R}^{m,r}$ for $\ell > m$.

The formalism of memory-aware reducibility allows us to investigate the minimal necessary memory requirements of the simulator for *any* such domain-extension construction. We prove two lower bounds on the memory required by the simulator, with the following consequences (see Section 4 for the precise bounds):

1. With stateless simulators (i.e., without any memory) even domain extension by a single bit (i.e., $\ell = m + 1$) is impossible.
2. For a natural class of simulators issuing at most one query to the ideal resource per invocation, any simulator for a domain extension by d bits (i.e., $\ell - m = d$) requires at least d bits of memory.

These bounds hold for both the information-theoretic and the computational setting. They naturally imply analogous impossibility results for constructing an arbitrary input-length random oracle, with the obvious transition of ℓ denoting the sum of the lengths of all queries issued to the random oracle. This answers negatively the open question of the existence of such a construction using no simulator memory asked in [29]. To appreciate the relevance of the memory requirements of a simulator in a reduction statement, we again point to Appendix A for an example with a more detailed explanation.

As another consequence, we also obtain the irreducibility of the random oracle to the ideal cipher with respect to stateless simulators, in contrast to the equivalence of these two ideal primitives with respect to classical indifferentiability [14,15,22].

Random Oracles Used by Multiple Parties. The impossibility results described above have some intriguing consequences for the setting where a random oracle is being used in a protocol by multiple parties, if we consider that several of these parties might deviate from the prescribed protocol in a potentially non-coordinated way (for example due to conflicting goals). According to the abstract cryptography framework [26], a security notion for such a situation has to involve local simulators for each of the parties that deviate from the protocol. Clearly, if a distinguisher is allowed to access two such simulators (for two of the parties) in the ideal world, these have to be stateless as otherwise they would produce inconsistent results when brought to different states. On the other hand, our results described above imply that also for this setting, no stateless simulator can exist. Hence, roughly speaking, one can conclude that for settings where one cannot assume a central adversary coordinating all the actions of the misbehaving parties, no secure construction of a random oracle from an ideal compression function exists. This might be relevant in the contexts of *rational cryptography* [20], *incoercible computation* [9],

receipt-free voting [6] or *collusion-free computation* [23,1] and its recent composable variants [2,11]. We formalize the above argument in Section 5 as an illustration of the impact of our results.

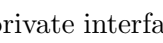
2 Preliminaries

Basic Notation. We denote sets by calligraphic letters or capital greek letters (e.g. $\mathcal{X}, \Sigma$) and their cardinalities by $|\mathcal{X}|, |\Sigma|$. For a superset clear from the context, we denote the complement of a set $\mathcal{X}$ by $\overline{\mathcal{X}}$. Throughout the paper all logarithms considered are to the base 2. The notation $\lceil \cdot \rceil$ corresponds to the usual ceiling function.

We denote random variables and concrete values they can take on by upper-case letters $X, Y, \ldots$ and lower-case letters $x, y, \ldots$, respectively. For random variables U and V with ranges $\mathcal{U}$ and $\mathcal{V}$, respectively, we let $\mathsf{P}_{U|V}$ be the corresponding conditional probability distribution, seen as a (partial) function $\mathcal{U} \times \mathcal{V} \to [0, 1]$. Here the value $\mathsf{P}_{U|V}(u, v) = \mathsf{P}[U = u|V = v]$ is well defined for all $u \in \mathcal{U}$ and $v \in \mathcal{V}$ such that $\mathsf{P}_V(v) > 0$ and undefined otherwise. For a discrete random variable X with range $\mathcal{X}$ we denote by $H(X)$ the Shannon entropy of X, i.e., $H(X) = \sum_{x \in \mathcal{X}} -\mathsf{P}_X(x) \log \mathsf{P}_X(x)$ where $\mathsf{P}_X(x)$ denotes the probability that X takes on the value $x \in \mathcal{X}$. Moreover, we denote by $H(Y|X)$ the usual notion of conditional entropy of Y given X, satisfying the chain rule $H(Y|X) = H(XY) - H(X)$. For a probability $p \in [0, 1]$ we also use the notion of binary entropy denoted $h(p)$ that is defined as the Shannon entropy of the binary random variable taking on the two possible values with probabilities p and $1 - p$.

Resources, Converters and Distinguishers. To formulate our results we use the language of abstract systems [26,25] to which we give here a self-contained introduction, partly following the exposition given in [28]. At the highest level of abstraction, a system is an object with interfaces via which it interacts with its environment (consisting of other systems). Two systems can be composed by connecting one interface of each system, and the composed object is again a system. Also, every two different systems are mutually independent.

We consider three distinct types of systems: *resources*, *converters* and *distinguishers*. Resources[3] are denoted by upper-case boldface letters such as $\mathbf{S}, \mathbf{T}$. In this paper we mostly (but not always) consider resources with two interfaces, hence our exposition here will only cover this case. In the indifferentiability setting these interfaces are referred to as private and public (for reasons explained below). Examples of resources discussed below are a fixed input-length random oracle with input length m and output length r denoted $\mathbf{R}^{m,r}$; an arbitrary input-length random oracle with output length n denoted $\mathbf{R}^{*,n}$; and an ideal block cipher with key length k and block length n denoted $\mathbf{E}^{k,n}$. Unless indicated otherwise, we see these as 2-interface resources providing access to the same random function at each interface.

Converters are systems having one *inner* and one *outer* interface and are denoted by small Greek letters such as ϕ, π, σ. The set of all converters considered is denoted as Σ. A converter ϕ can be composed with a resource $\mathbf{S}$ by attaching the inner interface of ϕ to one of the interfaces of $\mathbf{S}$. For example, if ϕ is attached to the private interface of $\mathbf{S}$ this can be depicted as 

. Note that the composed system is again a 2-interface resource that exposes the outer interface of ϕ instead of the interface of $\mathbf{S}$ to which ϕ was connected, together with the other interface of $\mathbf{S}$.

To describe the composition of resources and converters algebraically, we can take advantage of the restriction to 2-interface resources: We will understand the left and the right side of the symbol $\mathbf{S}$ as representing the private and the public interface of the system $\mathbf{S}$, respectively. Hence, attaching a converter π to the left (private) interface of a resource $\mathbf{S}$ results in a resource $\pi\mathbf{S}$ while attaching a converter σ to the right (public) interface of a resource $\mathbf{T}$ results in a resource $\mathbf{T}\sigma$. If two 2-interface resources $\mathbf{S}$ and $\mathbf{T}$ are used in parallel, this is denoted as $\mathbf{S}\|\mathbf{T}$ and is again a 2-interface resource; each of

[3] In this paper we sometimes also use the term "resources" in a more informal way to refer to computational power, memory, etc. The intended meaning should always be clear from the context.



Fig. 1: The real (left) and the ideal (right) setting considered in the definition of indifferentiability.

the interfaces of $\mathbf{S}\|\mathbf{T}$ allows to access the corresponding interface of both subsystems $\mathbf{S}$ and $\mathbf{T}$. The term id refers to the "identity converter" that forwards all inputs and outputs. For a 2-interface system $\mathbf{S}$ we sometimes denote by $[\mathbf{S}]_L(x)$ (resp. $[\mathbf{S}]_R(x)$) its response to a query x on its left (resp. right) interface.

We instantiate the general concept of abstract systems given above by considering (probabilistic) systems that communicate by passing messages from discrete sets and within discrete time steps. These can be formalized by the notion of random systems [24], i.e., conditional distributions of the outputs of the system (as random variables) given all previous inputs and outputs, where each input or output is associated to a specific interface. Since being sufficient for our setting, we only consider resources that only produce output in response to an input and on the same interface where the input was received. For a converter we assume that it is always invoked by a query at the outer interface, it then issues zero or more queries to the resource attached to its inner interface and finally produces an output at the outer interface. Under these assumptions, the behavior of composed systems is determined in the natural way, with the parallel composition of two resources defined asynchronously: each input at an interface of $\mathbf{S}\|\mathbf{T}$ explicitly specifies one of the subsystems, and this subsystem is invoked with the input.

A *distinguisher* $\mathbf{D}$ is a system that connects to all interfaces of a resource $\mathbf{T}$ and outputs (at a separate interface) a single bit denoted B. The complete interaction of $\mathbf{D}$ and $\mathbf{T}$ defines a random experiment and the probability that the bit B is 1 in this experiment is written as $\mathsf{P}^{\mathbf{DT}}(B=1)$. The *distinguishing advantage of* $\mathbf{D}$ *for the systems* $\mathbf{S}$ *and* $\mathbf{T}$ is then defined as $\Delta^{\mathbf{D}}(\mathbf{S},\mathbf{T}) := \left|\mathsf{P}^{\mathbf{DS}}(B=1) - \mathsf{P}^{\mathbf{DT}}(B=1)\right|$. We denote by $\mathcal{D}$ the set of all distinguishers considered and define $\Delta^{\mathcal{D}}(\mathbf{S},\mathbf{T}) := \sup_{\mathbf{D}\in\mathcal{D}} \Delta^{\mathbf{D}}(\mathbf{S},\mathbf{T})$.

Classical (weak) Indifferentiability. In the classical indifferentiability defined in [27] one restricts only to resources having two interfaces. The first one, referred to as *private*, is meant to model the access to the resource by all honest parties. On the other hand, the second interface is called *public* and is present to model the adversarial access to the internal state of the resource.

Let $\mathbf{S}$ and $\mathbf{T}$ be such 2-interface resources. We define $\mathbf{T}$ *being ε-reducible to* $\mathbf{S}$ *in the sense of weak indifferentiability* (denoted $\mathbf{S} \xrightarrow[\text{wi}]{\varepsilon} \mathbf{T}$) as

$$\mathbf{S} \xrightarrow[\text{wi}]{\varepsilon} \mathbf{T} \quad :\Leftrightarrow \quad (\exists \pi \in \Sigma)(\forall \mathbf{D} \in \mathcal{D})(\exists \sigma \in \Sigma) : \Delta^{\mathbf{D}}(\pi\mathbf{S}, \mathbf{T}\sigma) \leq \varepsilon$$

and refer to the converters π and σ as the protocol and the simulator, respectively. Usually we call $\mathbf{S}$ the real and $\mathbf{T}$ the ideal resource; hence also the random experiment of $\mathbf{D}$ interacting with $\pi\mathbf{S}$ (resp. $\mathbf{T}\sigma$) is called the real (resp. ideal) experiment.[4] The two settings distinguished are depicted in Fig. 1. Note that by choosing the sets of converters Σ and distinguishers $\mathcal{D}$ considered, this definition covers both information-theoretic and computational indifferentiability (as is true for all other definitions of reducibility presented below).

Strong Indifferentiability. We define $\mathbf{T}$ *being ε-reducible to* $\mathbf{S}$ *in the sense of strong indifferentiability* (denoted $\mathbf{S} \xrightarrow[\text{si}]{\varepsilon} \mathbf{T}$) as

$$\mathbf{S} \xrightarrow[\text{si}]{\varepsilon} \mathbf{T} \quad :\Leftrightarrow \quad (\exists \pi, \sigma \in \Sigma)(\forall \mathbf{D} \in \mathcal{D}) : \Delta^{\mathbf{D}}(\pi\mathbf{S}, \mathbf{T}\sigma) \leq \varepsilon.$$

Clearly reducibility under strong indifferentiability implies reducibility under the weak one and moreover, positive indifferentiability results (such as those in [14] showing security of MD-variants) typically prove this stronger type of statement by exhibiting a simulator that does not depend on the distinguisher. A

[4] One can also give an asymptotic definition, stating that (a family of resources) $\mathbf{T}_k$ is reducible to (a family of resources) $\mathbf{S}_k$ (denoted as $\mathbf{S}_k \xrightarrow[\text{wi}]{} \mathbf{T}_k$) if $(\exists \pi \in \Sigma)(\forall \mathbf{D} \in \mathcal{D})(\exists \sigma \in \Sigma) : \Delta^{\mathbf{D}}(\pi\mathbf{S}_k, \mathbf{T}_k\sigma) \leq \varepsilon(k)$ for a negligible function ε.

detailed discussion of the relationship between these two forms of simulatability in various formalisms can be found in [21,12], here we only remark that both notions are composable in the spirit of Theorem 1 (see below).

Domain Extension for Hash Functions. Finally, we briefly introduce the domain extension construction chop-MD from [14] that will serve us as a useful example throughout the paper. Let $f\colon \{0,1\}^{r+d} \to \{0,1\}^r$ be a compression function. The function $\mathsf{chop\text{-}MD}^f\colon \{0,1\}^* \to \{0,1\}^{r/2}$ is as defined in the box on the right. The role of the function Pad is to append the length of the message and a padding in a decodable way to obtain m' with length being a multiple of d bits. It will not be relevant for our discussion.

function $\mathsf{chop\text{-}MD}^f(m)$
  $m' \leftarrow \mathsf{Pad}(m)$
  parse m' **as** $m_1\|\cdots\|m_b$ **for** $m_i \in \{0,1\}^d$
  $y_0 \leftarrow 0^r$ (or any fixed initialization vector)
  for $i = 1$ **to** b **do** $y_i \leftarrow f(m_i\|y_{i-1})$
  return first $r/2$ **bits of** y_b

3 Memory-Aware Reducibility

Stateless simulators. To formally define memory-aware reducibility, we need to consider the class of *stateless converters* in the following sense. A stateless converter is a converter that uses no memory between answering outer queries, i.e., its (possibly randomized) behavior for a particular query depends only on the query itself and the ongoing interaction at the inner interface, not on previous outer queries and the transcript of the interaction during their evaluation. A formal definition follows.

Definition 1. *A converter ϕ is stateless if there exists a sequence of conditional probability distributions $\mathsf{p}^{\phi}_{IX_{j+1}|X_1\ldots X_jY_1\ldots Y_jQ}$ for $j \geq 0$ such that whenever ϕ received a query q at the outer interface and has then issued the sequence of queries $x_1,\ldots,x_j$ to the inner interface, obtaining responses $y_1,\ldots,y_j$, then $\mathsf{P}^{\phi}_{IX_{j+1}|X_1\ldots X_jY_1\ldots Y_jQ}(i,x_{j+1},x_1,\ldots,x_j,y_1,\ldots,y_j,q)$ determines the probability that its next action will be to output the value x_{j+1} at interface $i \in \{\mathsf{inner},\mathsf{outer}\}$. For a set of converters Σ we denote by Σ_{s} the set of all stateless converters from Σ.*

For example, the converter accessing an ideal compression function and realizing a Merkle-Damgård construction on top of it would be stateless according to the above definition.

Quantifying the Memory Requirements of the Simulator. Let $\mathbf{M}_s$ denote a resource that provides a dummy private interface and at the public (adversarial) interface, it provides the functionality of s-bit memory, i.e., allows efficient storage and retrieval of arbitrary information such that its size is in every point in time upper-bounded by s bits.

To quantify the memory requirements of the simulator in a reducibility statement we shall use the following formalism, which we broadly denote as *memory-aware reducibility*.

Definition 2. *We define* $\mathbf{T}$ being ε-reducible to $\mathbf{S}$ in the presence of s bits of adversarial memory *(denoted* $\mathbf{S}\xrightarrow[\mathrm{m}]{\varepsilon,s}\mathbf{T}$*) as*

$$\mathbf{S}\xrightarrow[\mathrm{m}]{\varepsilon,s}\mathbf{T} \quad :\Leftrightarrow \quad (\exists\pi\in\Sigma)(\forall\mathbf{D}\in\mathcal{D})(\exists\sigma\in\Sigma_{\mathsf{s}}) : \Delta^{\mathbf{D}}(\pi\mathbf{S},[\mathbf{T}||\mathbf{M}_s]\sigma) \leq \varepsilon.$$

Informally speaking, the statement $\mathbf{S}\xrightarrow[\mathrm{m}]{\varepsilon,s}\mathbf{T}$ indicates that $\mathbf{T}$ can be constructed securely from $\mathbf{S}$ within error ε in an environment where the adversary has s bits of memory available. In other words, whatever the adversary can achieve in the real world she could also achieve in the ideal world, but it might need up to s more bits of memory to do so. Evaluating whether this is acceptable depends on the context in which we want to use $\mathbf{S}$ instead of $\mathbf{T}$.

As before, by specifying the sets of converters and distinguishers to be considered, this definition covers both computational and information-theoretic memory-aware reducibility. As in the case of classical indifferentiability, the transition to an asymptotic definition would be trivial and we omit it.

In the special case of no memory (i.e., $s = 0$) the notion of memory-aware reducibility $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon,0} \mathbf{T}$ collapses to the notion of reducibility with stateless simulators. If we refer to this situation, we usually omit the 0 and simply write $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon} \mathbf{T}$. Technically, this special case is equivalent to the notion of reset indifferentiability introduced in [29]: First, if the simulator is stateless, then it can be used also in the scenario with resets with the same outcome. On the other hand, any simulator that satisfies the requirements of reset indifferentiability must be able to simulate successfully even in presence of an adversary that resets it before every query, hence there also exists an equivalent stateless simulator. However, our motivation to introduce stateless simulators is completely different. We do not put it forward as a security notion by itself, but only as a tool for modeling the memory requirements of the simulator explicitly.

Composability. An important reason to model the memory necessary for the simulation as a part of the ideal resource is that this allows us to make statements that are composable. Here we only state the respective composition theorem informally, its full statement and proof are given in Appendix C.

Theorem 1 (Informal). *Let* $\mathbf{R}$, $\mathbf{S}$ *and* $\mathbf{T}$ *be resources such that* $\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S}$ *and* $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon_2,s_2} \mathbf{T}$. *Assuming certain closure properties of the sets of converters and distinguishers considered, we have:*

1. *For any resource* $\mathbf{U}$, $\mathbf{R}\|\mathbf{U} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S}\|\mathbf{U}$ *and* $\mathbf{U}\|\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{U}\|\mathbf{S}$,
2. $\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1+\varepsilon_2,s_1+s_2} \mathbf{T}$.

4 Lower Bounds on Simulator Memory for Any Domain-Extending Construction

We now investigate the amounts of memory that we must assume to be available to the adversary in order to be able to conclude the security of classical domain extension constructions for hash functions.

4.1 Fixed Input-Length Random Oracles

The following theorem upper-bounds the achievable domain extension for fixed input-length random oracles, given a bound on the memory available to the simulator. In particular, it implies that without simulator memory, even domain extension by a single bit becomes impossible. This solves an open problem introduced in [29].

Theorem 2. *Any reduction* $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{\ell,r}$ *with*[5] $r \geq 2$ *and* $\varepsilon \leq 0.04$ *satisfies*[6]

$$\ell - m \leq s + \lceil \log(\min\{s,t\}) \rceil \tag{1}$$

where $t \geq 1$ *denotes an upper bound on the number of queries the simulator issues to the ideal primitive* $\mathbf{R}^{\ell,r}$ *to answer a single query.*

Proof. Recalling Def. 2, let us denote by π the protocol performing the reduction from the statement and let us consider a distinguisher $\mathbf{D}$ interacting with either $\pi\mathbf{R}^{m,r}$ or $[\mathbf{R}^{\ell,r}\|\mathbf{M}_s]\sigma$, where σ is the stateless simulator corresponding to $\mathbf{D}$.

Overview. In our proof we only consider the trivial distinguisher $\mathbf{D}$ given in Fig. 2 that chooses a random input $X \in \{0,1\}^\ell$ and then evaluates the $\{0,1\}^\ell$-domain function on the input X in two different ways. First it queries the private (left) interface for the whole input X; second it simulates the protocol π on X on its own and uses the public (right) interface to answer all the $\{0,1\}^m$-queries issued by π. Finally, $\mathbf{D}$ outputs 1 if and only if the two values obtained from these evaluations are equal. The distinguisher $\mathbf{D}$ participating in both the real and the ideal setting is depicted in Fig. 3. Note that we tacitly assume that this trivial distinguisher $\mathbf{D}$ is present in any reasonable distinguisher class that could be considered.

Distinguisher $\mathbf{D}(\mathbf{S})$ where $\mathbf{S} \in \{\pi\mathbf{R}^{m,r}, [\mathbf{R}^{\ell,r}\|\mathbf{M}_s]\sigma\}$

$X \xleftarrow{\$} \{0,1\}^\ell$
$Y \leftarrow [\mathbf{S}]_L(X)$
simulate π **to evaluate** $\hat{Y} \leftarrow [\mathbf{S}\pi]_R(X)$
if $Y = \hat{Y}$ **return** 1 **else return** 0

Fig. 2: The distinguisher $\mathbf{D}$ for the proof of Theorem 2. For a 2-interface system $\mathbf{S}$ we sometimes denote by $[\mathbf{S}]_L(x)$ (resp. $[\mathbf{S}]_R(x)$) its response to a query x on its left (resp. right) interface.

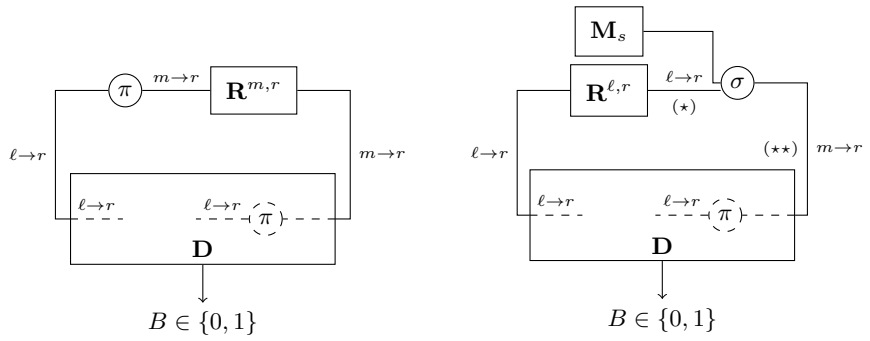


Fig. 3: The real and the ideal setting for the proof of Theorem 2. The notation $i \to o$ describes an interface that accepts queries from $\{0,1\}^i$ and responds with elements from $\{0,1\}^o$.

Clearly if $\mathbf{D}$ interacts with $\pi\mathbf{R}^{m,r}$ it always outputs 1. It remains to analyze the probability of $\mathbf{D}$ outputting 1 when interacting with $[\mathbf{R}^{\ell,r}\|\mathbf{M}_s]\sigma$. To this end, we consider the ideal setting depicted on the right-hand side of Fig. 3 and upper-bound the probability that the output of the protocol π simulated by $\mathbf{D}$ will be the correct value $\mathbf{R}^{\ell,r}(X)$. Informally speaking, we do this by upper-bounding the amount of useful information that this π can obtain about the actual values of $\mathbf{R}^{\ell,r}$ and show that it is not enough to recover $\mathbf{R}^{\ell,r}(X)$ with sufficient probability.

We use two separate approaches to bound this amount, each proving the above claim for one of the values in the minimum term in (1). In the first approach, we upper-bound the number of distinct queries the simulator σ is able to issue to $\mathbf{R}^{\ell,r}$ in any of its possible configurations (determined by the query it is answering and the state of its memory), thus using the channel denoted $(\star)$ in Fig. 3 as the "bottle-neck" to be considered. On the other hand, in the second approach we upper-bound the information provided by σ to π, this time the channel $(\star\star)$ acting as the "bottle-neck". We now give the details of both approaches.

First approach: the channel $(\star)$. Let us denote by $f(w, z, r_\sigma, r_{\mathbf{R}}) \subseteq \{0,1\}^\ell$ the set of all queries that the (stateless) simulator σ issues to the random oracle $\mathbf{R}^{\ell,r}$ while evaluating an outer query $w \in \{0,1\}^m$ with the available memory $\mathbf{M}_s$ containing value $z \in \{0,1\}^s$, using randomness r_σ while the responses from $\mathbf{R}^{\ell,r}$ are determined by its randomness $r_{\mathbf{R}}$. Since the random variables R_σ and $R_{\mathbf{R}}$ represent the only sources of randomness in this interaction, f is a well-defined deterministic mapping and by our assumption we have $|f(w, z, r_\sigma, r_{\mathbf{R}})| \leq t$ for all possible inputs. Let us define $\mathcal{S}_{r_\sigma, r_{\mathbf{R}}}$ to be the set of all

[5] The bound degrades gracefully for smaller r and bigger ε. In particular, for the same ε and $r = 1$ with no memory ($s = 0$) domain extension by a single bit is still impossible.

[6] To avoid handling the special case $s = 0$ separately we use the notational convention $\log 0 = 0$ throughout this section.

possible queries under all inputs (w, z) for a fixed randomness $(r_\sigma, r_{\mathbf{R}})$, i.e.,

$$\mathcal{S}_{r_\sigma, r_{\mathbf{R}}} := \bigcup_{\substack{w \in \{0,1\}^m \\ z \in \{0,1\}^s}} f(w, z, r_\sigma, r_{\mathbf{R}}),$$

then we have $|\mathcal{S}_{r_\sigma, r_{\mathbf{R}}}| \leq 2^{m+s+\log t}$ for any fixed randomness $(r_\sigma, r_{\mathbf{R}})$. Since $X \in \{0,1\}^\ell$ was chosen at random and independently from $\mathcal{S}_{R_\sigma, R_{\mathbf{R}}}$, we obtain $\mathsf{P}(X \in \mathcal{S}_{R_\sigma, R_{\mathbf{R}}}) \leq 2^{m+s+\log t}/2^\ell = 2^{m+s+\log t-\ell}$. Hence, if $\ell - m > s + \lceil \log t \rceil$ then $X \notin \mathcal{S}_{R_\sigma, R_{\mathbf{R}}}$ with probability at least $1/2$. However, if $X \notin \mathcal{S}_{R_\sigma, R_{\mathbf{R}}}$ then π has no information about $\mathbf{R}^{\ell,r}(X)$ and hence can only guess it successfully with negligible probability. Therefore, any proper simulation requires $\ell - m \leq s + \lceil \log t \rceil$.

SECOND APPROACH: THE CHANNEL ($\star\star$). In this case, let us denote by $\sigma_z(w)$ the response of σ to a query $w \in \{0,1\}^m$ with its memory set to the value $z \in \{0,1\}^s$ and let us denote by $Z'(w, z) \in \{0,1\}^s$ the new contents of the memory after this invocation of σ. Note that since σ is stateless, both $\sigma_z(w)$ and $Z'(w, z)$ are random variables fully determined by the function table of $\mathbf{R}^{\ell,r}$ and the internal randomness of σ. We can now define T to be the table containing the values $\sigma_z(w)$ and $Z'(w, z)$ for all possible w and z, formally $T := \{(\sigma_z(w), Z'(w, z)) \mid (w, z) \in \{0,1\}^m \times \{0,1\}^s\}$ .We then have $|T| = (r + s) \cdot 2^{m+s}$ and T can again be seen as a random variable fully determined by the function table of $\mathbf{R}^{\ell,r}$ and the internal randomness of σ.

We now consider a different protocol ρ instead of π which we allow to be stateful, but we only provide it with access to T, not σ (which we denote by ρ^T). We claim that the probability of the best such ρ in reconstructing $\mathbf{R}^{\ell,r}(X)$ given access to T is not smaller than the same probability for π given access to the right interface of $[\mathbf{R}^{\ell,r} \| \mathbf{M}_s]\sigma$, i.e., we have

$$\max_\rho \mathsf{P}[\rho^T(X) = \mathbf{R}^{\ell,r}(X)] \geq \mathsf{P}\left[[[\mathbf{R}^{\ell,r} \| \mathbf{M}_s]\sigma\pi]_R(X) = \mathbf{R}^{\ell,r}(X)\right]. \tag{2}$$

This is because one possible ρ to be considered on the left side of (2) is the following: it simulates π and answers each of π's queries to σ using the respective value from T instead. It also keeps track of the memory contents in its own state, updating it after each answered query according to the value given in T. This ρ clearly achieves equality in (2).

Now, since any ρ as described above only has access to T, we can use a corollary of the well-known Fano's inequality [19] to upper-bound the probability of ρ successfully reconstructing $\mathbf{R}^{\ell,r}(X)$ based on T. To simplify the notation, we shall denote by F the whole function table of $\mathbf{R}^{\ell,r}$ seen as a random variable (uniformly distributed over $\{0,1\}^{r2^\ell}$). The value X is chosen independently at random, hence we can apply Corollary 4 from Appendix B to obtain a lower bound on the probability $\bar{p}_e$ of error in a randomly chosen bit of $\mathbf{R}^{\ell,r}(X)$:

$$\begin{aligned} h(\bar{p}_e) &\geq \frac{1}{r2^\ell} H(F|T) = \frac{1}{r2^\ell} \left(H(FT) - H(T)\right) \geq \frac{1}{r2^\ell} \left(H(F) - H(T)\right) \\ &\geq \frac{1}{r2^\ell} \left(r2^\ell - (r+s)2^{m+s}\right) = 1 - 2^{m+s-\ell} - \left(\frac{s}{r}\right) 2^{m+s-\ell} \end{aligned}$$

If $\ell - m > s + \lceil \log s \rceil$ then since m, s, ℓ are integers we get $2^{m+s-\ell} \leq 1/2$ and $(s/r) \cdot 2^{m+s-\ell} \leq 1/2r$, hence $h(\bar{p}_e) \geq 1/2 - 1/2r$, resulting in $\bar{p}_e \geq 0.04$ for $r \geq 2$. Therefore any simulator successful beyond 96% has to satisfy $\ell - m \leq s + \lceil \log s \rceil$ as desired. □

Before we apply our result also to other contexts, note that our argument above is completely information-theoretic and hence the bound applies to both information-theoretic *and* computational memory-aware reducibility.

4.2 Arbitrary Input-Length Random Oracles

Seen from a different perspective, the above theorem also imposes a lower bound on the required simulator memory for any reduction of an arbitrary-input length random oracle to a fixed-input length random oracle (i.e., an ideal compression function) as a function of the sum of the lengths of hashed messages. We now investigate this setting.

Let $\mathcal{D}_\ell$ denote the class of distinguishers such that the total sum of the lengths of all queries asked to the private interface by any $\mathbf{D} \in \mathcal{D}_\ell$ is upper-bounded by ℓ.

Corollary 1. *If we only consider distinguishers $\mathbf{D} \in \mathcal{D}_\ell$ then any reduction $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{*,r}$ with $r \geq 2$ and $\varepsilon \leq 0.04$ satisfies*

$$s \geq \ell - m - \lceil \log(\min\{s,t\}) \rceil$$

where $t \geq 1$ denotes an upper bound on the number of queries the simulator itself issues to the ideal primitive to answer a single query. More generally, for the same set of distinguishers $\mathcal{D}_\ell$ and any reduction $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{*,n}$ with $\varepsilon \leq 0.04$ we still have $s \geq \ell - m - \lceil \log t \rceil$.

Proof. The distinguisher $\mathbf{D}$ from the proof of Theorem 2 hashing a single message of length ℓ clearly satisfies the restriction on the total query length, hence we can apply the same reasoning as in the proof of Theorem 2. The second claim follows from the fact that the analysis of the first case ($\star$) in the proof of Theorem 2 does not depend on the range of the ideal resource. □

To illustrate the meaning of the above statement, let us consider the domain extension construction chop-MD described in Section 2. The simulator presented in [14] to show that chop-MD using an ideal compression function is indifferentiable from a random oracle requires (without any optimizations) $\Omega((m+r)q)$ memory to simulate q queries. Hence, for a distinguisher hashing a single message of length ℓ using both interfaces (as in the proof of Theorem 2) the simulator needs roughly $\Omega((1+r/m)\cdot\ell)$ bits of memory. Moreover, it always asks at most one query to the ideal primitive to answer a single query itself — we will refer to such simulators as *single-query simulators*.

Our result implies that for any indifferentiable domain extension construction, if the respective simulator is single-query then it needs memory at least linear in ℓ. Since in a typical setting it is reasonable to assume $\ell \gg m, r, n$; our result implies that the simulator given in [14] for chop-MD has asymptotically optimal memory requirements within the class of single-query simulators.

4.3 Random Oracle vs. Ideal Cipher

Our proof of Theorem 2 relies on information-theoretic arguments that remain valid also after introducing additional permutation structure into the real resource. Hence, as a side result, we also obtain the impossibility of reducing an arbitrary input-length random oracle to an ideal cipher with respect to stateless simulators. This is in contrast to the results of [14] that demonstrate the possibility of such reduction with respect to stateful simulators.

Corollary 2. *The random oracle $\mathbf{R}^{*,r}$ cannot be reduced to the ideal cipher $\mathbf{E}^{k,n}$ with respect to stateless simulators, i.e., $\mathbf{E}^{k,n} \not\xrightarrow[\mathrm{m}]{} \mathbf{R}^{*,r}$.*

Note that for simplicity we use the asymptotic formulation here and only consider stateless simulators. One could also give a more fine-grained statement quantifying a lower bound on the memory requirements for any simulator used in a reduction of $\mathbf{R}^{*,r}$ to $\mathbf{E}^{k,n}$, in the spirit of Corollary 1.

5 Domain Extension is Impossible in a General Multi-Party Setting

As a particular application of our results, in this section we present some surprising consequences of the lower bound on simulator memory for domain extension of random functions established in the previous section.

The approach taken in any indifferentiability analysis is to model the system in question as having two interfaces: the private one and the public one, as described in Section 2. However, we often consider the constructed primitives to be used in an environment or protocol involving multiple parties. For example, a random oracle is typically understood to be available to all entities participating in a protocol (or possibly many concurrent protocols) that use it. The generic translation of an indifferentiability result into a security guarantee for such a setting is then tacitly assumed. Namely, we view *all* the honest parties as accessing identical copies of the private interface of the real primitive, each party running a local copy of the protocol π realizing the reduction. On the other hand, *all* the misbehaving parties are allowed to access the internals of the construction via identical copies of the public interface.

This implicit reasoning step imposes some requirements on the reduction used. First of all, the protocol π used by all the honest parties must be stateless in the sense of Definition 1. This is intuitively easy to see, since an inherently stateful protocol could lead to inconsistent behavior observed by different honest parties. For example, when constructing a random oracle from an ideal compression function, in the ideal world the resource (a random oracle) is stateful, with a single state (its function table) accessible to all honest parties. If in the real world a part of this state was stored by the protocol, different parties running different instances of the protocol could obtain different function values for the same query. Naturally, typical protocols constructing a random oracle from an ideal compression function such as the variants of the Merkle-Damgård construction proposed in [14] are indeed designed to be stateless.

It turns out that for a generic transition from an indifferentiability statement to a security guarantee in a setting with multiple parties, only using stateless protocols is by itself *not* sufficient. However, before we can formally approach this question, we first have to describe how we formulate security requirements in the multi-party setting. For this task we use the approach of abstract cryptography (AC) of Maurer and Renner.

AC Reducibility. Here we only give a very brief introduction to the AC framework required for our exposition, further details and the justification of the framework are given in [26]. The framework introduces a strong notion of isomorphism given at a very abstract level that, when applied to the particular setting of abstract systems, gives rise to the security notion described below. Its main technical difference compared to other simulation-based security definitions (e.g. [12,4]) relevant for our discussion is that it requires the existence of a *local* simulator for each of the parties.

From now on, we will be discussing more general resources having n interfaces labeled $1, \ldots, n$, hence we also have to extend our notation. If $\hat{\phi} = (\phi_1, \ldots, \phi_n)$ is an n-tuple of converters and $\mathbf{S}$ is an n-interface resource, we write $\hat{\phi}\mathbf{S}$ to denote the resource $\mathbf{S}$ with the converter ϕ_i applied to its i-th interface for all $i \in \{1, \ldots, n\}$. For a subset $\mathcal{P} \subseteq \{1, \ldots, n\}$ and an n-tuple of converters $\hat{\phi} = (\phi_1, \ldots, \phi_n)$ let us denote by $\hat{\phi}_{\mathcal{P}}$ the n-tuple of converters that is obtained from $\hat{\phi}$ by replacing all converters on positions *not* in $\mathcal{P}$ by the identity converter id. Hence, for two n-interface resources $\mathbf{S}$ and $\mathbf{T}$, the notation $\hat{\pi}_{\mathcal{P}}\mathbf{S}$ below denotes the system $\mathbf{S}$ with a protocol from $\hat{\pi}$ connected to every interface in $\mathcal{P}$ while $\hat{\sigma}_{\overline{\mathcal{P}}}\mathbf{T}$ denotes $\mathbf{T}$ with a simulator from $\hat{\sigma}$ connected to every interface *not* in $\mathcal{P}$.

Let $\mathbf{S}$ and $\mathbf{T}$ be n-interface resources. We say that $\mathbf{T}$ is ε-reducible to $\mathbf{S}$ in the sense of AC (denoted $\mathbf{S} \xrightarrow[\text{AC}]{\varepsilon} \mathbf{T}$) if there exist two n-tuples of converters $\hat{\pi} = (\pi_1, \ldots, \pi_n)$ and $\hat{\sigma} = (\sigma_1, \ldots, \sigma_n)$ such that for every subset $\mathcal{P}$ of indices $\{1, \ldots, n\}$ and every distinguisher $\mathbf{D} \in \mathcal{D}$ we have $\Delta^{\mathbf{D}}(\hat{\pi}_{\mathcal{P}}\mathbf{S}, \hat{\sigma}_{\overline{\mathcal{P}}}\mathbf{T}) \leq \varepsilon$, i.e.:

$$\mathbf{S} \xrightarrow[\text{AC}]{\varepsilon} \mathbf{T} \quad :\Leftrightarrow \quad (\exists \hat{\pi}, \hat{\sigma} \in \Sigma^n)(\forall \mathcal{P} \subseteq \{1, \ldots, n\})(\forall \mathbf{D} \in \mathcal{D}) : \Delta^{\mathbf{D}}(\hat{\pi}_{\mathcal{P}}\mathbf{S}, \hat{\sigma}_{\overline{\mathcal{P}}}\mathbf{T}) \leq \varepsilon. \tag{3}$$

For a 1-interface resource $\mathbf{S}$, let us denote by $\hat{\mathbf{S}}_n$ the n-interface resource that provides access to the same internal copy of $\mathbf{S}$ on each of its interfaces. For $\mathcal{P} \subseteq \{1, \ldots, n\}$ and a distinguisher $\mathbf{D}$ from the class $\mathcal{D}$ let $\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})$ denote a new distinguisher for the 2-interface indifferentiability setting that works exactly as $\mathbf{D}$ does but asks all $\mathbf{D}$'s queries to interfaces in $\mathcal{P}$ at the private interface instead and all $\mathbf{D}$'s queries to interfaces in $\overline{\mathcal{P}}$ at the public interface instead. Moreover, let $\mathsf{Proj}_{\mathcal{P}}(\mathcal{D}) := \{\mathsf{Proj}_{\mathcal{P}}(\mathbf{D}) \mid \mathbf{D} \in \mathcal{D}\}$.

Generic Transition to n-Party Setting. Now we are ready to state a theorem that formalizes the above-mentioned generic transition from any indifferentiability statement to a more meaningful statement in the multi-party AC setting under certain assumptions. Since the isomorphism notion introduced in the AC framework requires us to make statements where the simulators are chosen independently of the distinguisher (such as in (3)), to relate indifferentiability to AC we make use of its strong version described in Section 2.

Theorem 3. *Let $\mathbf{S}$, $\mathbf{T}$ be 1-interface resources and let $n \in \mathbb{N}$. If $\hat{\mathbf{S}}_2 \xrightarrow{\varepsilon}_{\mathrm{si}} \hat{\mathbf{T}}_2$ for a class of distinguishers $\mathcal{D}$ and both the protocol π and the simulator σ used in this reduction are stateless then we have $\hat{\mathbf{S}}_n \xrightarrow{\varepsilon}_{\mathrm{AC}} \hat{\mathbf{T}}_n$ for any class of distinguishers $\mathcal{D}'$ such that $\mathsf{Proj}_{\mathcal{P}}(\mathcal{D}') \subseteq \mathcal{D}$ for all $\mathcal{P} \subseteq \{1, \ldots, n\}$.*

Proof (sketch). To obtain the n-tuples of protocols and simulators (denoted $\hat{\pi}$ and $\hat{\sigma}$, respectively) required to prove the AC statement, one can simply use n independent copies of the protocol π and the simulator σ, respectively. Let us consider a distinguisher $\mathbf{D} \in \mathcal{D}$ for some fixed set $P \subseteq \{1, \ldots, n\}$ interacting with either $\hat{\pi}_{\mathcal{P}}\hat{\mathbf{S}}_n$ or $\hat{\sigma}_{\overline{\mathcal{P}}}\hat{\mathbf{T}}_n$ and relate it to the distinguisher $\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})$ interacting with either $\pi\hat{\mathbf{S}}_2$ or $\hat{\mathbf{T}}_2\sigma$. Since both π and σ are stateless, clearly whenever $\mathbf{D}$ issues a query to an interface in $\mathcal{P}$, the distribution of the answer will be the same as if $\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})$ issued the same query to the private interface. Analogously, any query issued by $\mathbf{D}$ to an interface in $\overline{\mathcal{P}}$ corresponds in the same way to a query issued by $\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})$ to the public interface. Hence $\Delta^{\mathbf{D}}(\hat{\pi}_{\mathcal{P}}\hat{\mathbf{S}}_n, \hat{\sigma}_{\overline{\mathcal{P}}}\hat{\mathbf{T}}_n) = \Delta^{\mathsf{Proj}_{\mathcal{P}}(\mathbf{D})}(\pi\hat{\mathbf{S}}_2, \hat{\mathbf{T}}_2\sigma)$ and the theorem follows. □

Impossibility of Domain Extension. Let us now consider the specific case of reducing the random oracle[7] $\hat{\mathbf{R}}_n^{*,r}$ to the ideal compression function $\hat{\mathbf{R}}_n^{m,r}$ in the n-party case. One can easily see that in this case requiring the simulators σ_i to be stateless is also necessary, for the same reason as described for the protocols π_i.

Lemma 1. *For any $n \geq 2$ if a reduction $\hat{\mathbf{R}}_n^{m,r} \xrightarrow{\varepsilon}_{\mathrm{AC}} \hat{\mathbf{R}}_n^{*,r}$ exists then there also exists a reduction $\hat{\mathbf{R}}_n^{m,r} \xrightarrow{2\varepsilon}_{\mathrm{AC}} \hat{\mathbf{R}}_n^{*,r}$ using a simulator $\hat{\sigma} = (\sigma_1, \ldots, \sigma_n)$ where at least one of the local simulators σ_i is stateless.*

Proof (sketch). Let us consider the case in equation (3) where $\{1, 2\} \subseteq \overline{\mathcal{P}}$, hence any distinguisher $\mathbf{D}$ would in the ideal world interact with two local simulators σ_1 and σ_2 on the interfaces 1 and 2. Since σ_1 and σ_2 have to produce consistent responses to all possible sequences of queries from $\{0,1\}^m$, they must be ε-close to realizing a function that only depends on the function table of the ideal resource $\hat{\mathbf{R}}_n^{*,r}$. Hence, if we replace say σ_1 by the stateless converter that realizes this function mapping with probability 1, the distinguishing advantage of any distinguisher can only increase by at most ε. □

On the other hand, the lower bound on simulator memory proved in Section 4 also applies to the more general AC setting. Indeed, if there existed a reduction $\hat{\mathbf{R}}_n^{m,r} \xrightarrow{}_{\mathrm{AC}} \hat{\mathbf{R}}_n^{*,r}$ where at least one of the local simulators σ_i was stateless, this σ_i could also be used in the two-interface (weak) indifferentiability setting, contradicting Corollary 1. Hence we arrive at the following observation.

Lemma 2. *For any $n \geq 2$ if there exists a reduction $\hat{\mathbf{R}}_n^{m,r} \xrightarrow{}_{\mathrm{AC}} \hat{\mathbf{R}}_n^{*,r}$ using a protocol $\hat{\pi} = (\pi_1, \ldots, \pi_n)$ and a simulator $\hat{\sigma} = (\sigma_1, \ldots, \sigma_n)$ then none of the local simulators σ_i is stateless.*

[7] In the rest of the section we will use symbols such as $\mathbf{R}^{*,r}$ as to refer to the single interface resource and use the introduced notation to explicitly state the number of interfaces we want to consider.

Combining the last two statements we get the following impossibility result.

Corollary 3. *For any $n \geq 2$ the random oracle $\hat{\mathbf{R}}_n^{*,r}$ cannot be reduced to the ideal compression function $\hat{\mathbf{R}}_n^{m,r}$ in the AC setting, i.e., $\hat{\mathbf{R}}_n^{m,r} \not\xrightarrow{\text{AC}} \hat{\mathbf{R}}_n^{*,r}$.*

Note that by invoking Theorem 2 instead of Corollary 1 one could even obtain the impossibility of domain extension by a single bit in the AC-setting.

Hence by the above result it is impossible to construct a public random oracle from an ideal compression function in a multi-party environment where the parties must be modeled as possibly having conflicting goals or deviating from the protocol in an uncoordinated manner (or, technically speaking, in any scenario where a proper modeling requires the use of local simulators). This is in contrast to the two-party indifferentiability setting (with several constructions that achieve the desired transformation) where one implicitly makes the assumption that all dishonest parties are coordinated by a hypothetical central adversary. This seems to be a very strong assumption in particular for random oracles that are typically thought of as being used by many different parties in many different applications. Of course, a particular use of a construction proven secure in the 2-party scenario within a multi-party setting as discussed above might still be secure under some additional assumptions, however our result indicates that such use should always be explicitly justified.

6 Conclusion

In this paper we have introduced a systematic approach that replaces classical indifferentiability arguments in settings where a more fine-grained quantification of a certain resource is necessary.

Choosing memory as the resource to focus on, this allows us to explain from a different perspective the unexpected security failure of the protocol given in [29] when used with the construction chop-MD. We proceed by giving lower bounds on the required simulator memory for the reduction of a public random oracle to a public random function, showing that memory roughly linear in the total length of all queries is necessary and that with no memory even domain extension by a single bit becomes impossible.

Finally, we apply our result to the setting where the random oracle is used by multiple parties with no central adversary to coordinate potential misbehavior. We argue that special care must be taken in such settings when replacing the random oracle by a construction using an ideal compression function. We show that a construction that would be secure in every such setting does not exist and hence any use of a domain-extending construction in a particular setting of this type should be explicitly justified.

Acknowledgements. We would like to thank Thomas Holenstein and Robin Künzler for valuable input and fruitful discussions at the early stages of this work. In particular, they independently gave a proof of the impossibility of single-bit domain extension for the case of simulators without memory.

Our research was in part supported by the grants SNF 200020-132794, UK/23/2012 and by the Zurich Information Security and Privacy Center (ZISC).

References

1. Joël Alwen, Jonathan Katz, Yehuda Lindell, Giuseppe Persiano, Abhi Shelat, and Ivan Visconti. Collusion-free multiparty computation in the mediated model. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 524–540. Springer, 2009.
2. Joël Alwen, Jonathan Katz, Ueli Maurer, and Vassilis Zikas. Collusion-preserving computation. In Safavi-Naini and Canetti [30], pages 124–143.
3. Elena Andreeva, Bart Mennink, and Bart Preneel. On the Indifferentiability of the Grostl Hash Function. In Juan Garay and Roberto De Prisco, editors, *Security and Cryptography for Networks*, volume 6280 of *Lecture Notes in Computer Science*, pages 88–105. Springer Berlin / Heidelberg, 2010.

4. Michael Backes, Birgit Pfitzmann, and Michael Waidner. A general composition theorem for secure reactive systems. In Moni Naor, editor, *Theory of Cryptography*, volume 2951 of *Lecture Notes in Computer Science*, pages 336–354. Springer Berlin / Heidelberg, 2004.
5. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
6. Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In *Proceedings of the 26th annual ACM symposium on Theory of computing*, STOC '94, pages 544–553, New York, NY, USA, 1994. ACM.
7. Guido Bertoni, Joan Daemen, Michal Peeters, and Gilles Van Assche. Keccak specifications. Submission to NIST (Round 1), 2008.
8. Guido Bertoni, Joan Daemen, Michal Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel Smart, editor, *Advances in Cryptology EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer Berlin / Heidelberg, 2008.
9. Ran Canetti and Rosario Gennaro. Incoercible multiparty computation. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96*, pages 504–513. IEEE Computer Society, 1996.
10. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 209–218. ACM, 1998.
11. Ran Canetti and Margarita Vald. Universally composable security with local adversaries. In Ivan Visconti and Roberto De Prisco, editors, *SCN*, volume 7485 of *Lecture Notes in Computer Science*, pages 281–301. Springer, 2012.
12. Ron Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *42nd IEEE Symposium on Foundations of Computer Science, Proceedings of FOCS 2001*, pages 136–145. IEEE Computer Society Press, Oct. 2001. Full version at `http://eprint.iacr.org/2000/067`.
13. Donghoon Chang and Mridul Nandi. Improved Indifferentiability Security Analysis of chopMD Hash Function. In Kaisa Nyberg, editor, *Fast Software Encryption*, volume 5086 of *Lecture Notes in Computer Science*, pages 429–443. Springer Berlin / Heidelberg, 2008.
14. 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. Springer Berlin / Heidelberg, 2005.
15. Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In *CRYPTO 2008: Proceedings of the 28th Annual conference on Cryptology*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer-Verlag, 2008.
16. Yevgeniy Dodis, Leonid Reyzin, Ronald Rivest, and Emily Shen. Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to md6. In Orr Dunkelman, editor, *Fast Software Encryption*, volume 5665 of *Lecture Notes in Computer Science*, pages 104–121. Springer Berlin / Heidelberg, 2009.
17. Yevgeniy Dodis, Thomas Ristenpart, and Thomas Shrimpton. Salvaging merkle-damgrd for practical applications. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 371–388. Springer Berlin / Heidelberg, 2009.
18. Yevgeniy Dodis, Thomas Ristenpart, John P. Steinberger, and Stefano Tessaro. To Hash or Not to Hash Again? (In)Differentiability Results for H 2 and HMAC. In Safavi-Naini and Canetti [30], pages 348–366.
19. Robert Fano. *Transmission of Information: A Statistical Theory of Communications*. The MIT Press, Cambridge, MA, 1961.
20. Joseph Halpern and Vanessa Teague. Rational secret sharing and multiparty computation. In *Proceedings of the 36th annual ACM symposium on Theory of computing*, STOC '04, pages 623–632, New York, NY, USA, 2004. ACM.
21. Dennis Hofheinz and Dominique Unruh. Comparing two notions of simulatability. In Joe Kilian, editor, *Theory of Cryptography*, volume 3378 of *Lecture Notes in Computer Science*, pages 86–103. Springer Berlin / Heidelberg, 2005.
22. Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 89–98, New York, NY, USA, 2011. ACM.
23. Matt Lepinksi, Silvio Micali, and abhi shelat. Collusion-free protocols. In *Proceedings of the 37th annual ACM symposium on Theory of computing*, STOC '05, pages 543–552, New York, NY, USA, 2005. ACM.
24. Ueli Maurer. Indistinguishability of random systems. In Lars Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 110–132. Springer-Verlag, May 2002.
25. Ueli Maurer. Constructive cryptography – a new paradigm for security definitions and proofs. In S. Moedersheim and C. Palamidessi, editors, *Theory of Security and Applications (TOSCA 2011)*, volume 6993 of *Lecture Notes in Computer Science*, pages 33–56. Springer-Verlag, April 2011.
26. Ueli Maurer and Renato Renner. Abstract cryptography. In Bernard Chazelle, editor, *The Second Symposium on Innovations in Computer Science ICS 2011*, pages 1–21. Tsinghua University Press, January 2011.
27. Ueli Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *Theory of Cryptography Conference — TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer-Verlag, February 2004.

28. Ueli Maurer, Andreas Rüedlinger, and Björn Tackmann. Confidentiality and integrity: A constructive perspective. In Ronald Cramer, editor, *Theory of Cryptography — TCC 2012*, volume 7194 of *Lecture Notes in Computer Science*, pages 209–229. IACR, Springer, 2012.
29. Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In Kenneth Paterson, editor, *Advances in Cryptology — EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506. Springer Berlin / Heidelberg, 2011.
30. Reihaneh Safavi-Naini and Ran Canetti, editors. *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*. Springer, 2012.

A The Storage-Auditing Scenario from [29]

In [29], the scenario of auditable storage is put forward: a user storing some data M on a remote server wants to verify in an efficient way that his data is still present unchanged on the server. The challenge-response protocol for this task that is analyzed in [29] works as follows: the user sends the server a random challenge c and the server is required to respond with the value $H(M||c)$ for a cryptographically secure hash function H.

As argued in [29], this protocol is clearly secure in the random oracle model. However, if the hash function is instantiated by the Merkle-Damgård-chop construction $\mathsf{chop\text{-}MD}^f$ (described in Section 2) using an ideal compression function f, the protocol becomes completely insecure. A malicious server can simply parse the data M into blocks $m_1, \ldots, m_\ell$ and compute the value

$$r := f(f(\cdots f(f(IV, m_1), m_2) \cdots), m_\ell).$$

Now it can discard the data M and store only the value r, allowing it to respond correctly to any challenge c by returning the first half of the bits of $f(r, c)$. [8]

Since the construction chop-MD was shown indifferentiable from a random oracle in [14], this example contradicts the common understanding that an ideal primitive can be replaced by an indifferentiable construction in any context without compromising the security. The goal of this section is to explain this seemingly inconsistent situation.

In our view, the best way to understand the example described above is by explicitly taking into account the memory requirements of the simulator that is used to prove indifferentiability of the chop-MD construction from a random oracle.

First, let us see why this is a relevant aspect. The role of a simulator in an indifferentiability reduction statement is to capture the fact that anything that the adversary would be able to obtain from its interaction with the real resource, she could also obtain when interacting with the ideal one. This is since she could perform all the tasks embodied in the simulator on her own. However, such argumentation is only valid if it is feasible for the adversary to incorporate the simulator into itself, not violating its own complexity limitations.

This is clearly not the case in the scenario described above. If the goal of the adversary is to pass the challenge without actually storing the data M, it cannot afford to perform the job of the simulator itself. This is because the simulator proving indifferentiability of chop-MD from a random oracle presented in [14] actually remembers all the queried values during the interaction. Hence if we take the adversary that is able to cheat to pass the verification protocol with chop-MD without remembering M and try to modify it to work in the random oracle setting, it would additionally need to perform the simulator's job, remembering the whole data M. This invalidates the reduction proof.

[8] For simplicity, we assumed that the length of M is divisible by the block length and the challenge consists of a single block.

B Fano's Inequality

In our proof we use the following theorem given by Fano in [19].

Theorem 4 (Fano). *Let X and Y be random variables. For the error probability p_e of any algorithm reconstructing X given Y we have*

$$h(p_e) + p_e \cdot \log(|\mathcal{X}| - 1) \geq H(X|Y)$$

where $\mathcal{X}$ is the support of X.

If we assume X to be distributed over bitstrings of a certain length, one can easily derive a corollary of the above theorem that lower-bounds the error probability of reconstruction of a randomly chosen bit of X.

Corollary 4. *Let X, Y be random variables such that $X \in \{0,1\}^n$, i.e., $X = (X_1, X_2, \ldots, X_n)$ for $X_i \in \{0,1\}$. Moreover, let us consider any algorithm reconstructing X given Y and denote by $\tilde{X}_i$ the reconstruction of X_i. Then for the probability of error in a uniformly chosen random bit $\bar{p}_e := \frac{1}{n}\sum_{i=1}^{n} \mathsf{P}[\tilde{X}_i \neq X_i]$ we have*

$$h(\bar{p}_e) \geq \frac{1}{n} H(X|Y).$$

Proof. We have

$$h(\bar{p}_e) = h\left(\frac{1}{n}\sum_{i=1}^{n} \mathsf{P}[\tilde{X}_i \neq X_i]\right) \geq \frac{1}{n}\sum_{i=1}^{n} h(\mathsf{P}[\tilde{X}_i \neq X_i]) \geq \frac{1}{n}\sum_{i=1}^{n} H(X_i|Y) \geq \frac{1}{n} H(X|Y)$$

where we applied the Jensen inequality, the Fano inequality for the case $|\mathcal{X}| = 2$ and the basic properties of the entropy function. □

C Composition for Memory-Aware Reducibility

As a supporting material, we present in this section a proof of the composition theorem for memory-aware reducibility of 2-interface resources. In order to state this theorem, we need to make explicit the closure properties assumed on the set of 2-interface resources Φ_2, the set of converters Σ, and the set of distinguishers $\mathcal{D}$. Those closure properties correspond to the notion of a cryptographic algebra [26, Def. 14] and of a compatible distinguisher class [26, Def. 16].

C.1 Algebra of Resources

Let Φ_2 denote the set of 2-interface resources whose interfaces are labelled by an element of the set $\{\mathsf{left}, \mathsf{right}\}$. The parallel composition of resources $\|$ corresponds to a binary operation $\| : \Phi_2 \times \Phi_2 \to \Phi_2$, and we denote by $< \Phi_2, \| >$ the algebra formed by the set of 2-interface resources and the corresponding parallel composition operation.

C.2 Algebra of Converters

As described in Section 2, we only consider converters of the following type: a converter is invoked by a query at the outer interface, it then issues zero or more queries to its inner interface and finally produces an output at the outer interface. Let Σ be a subset of the set of such converters. The algebra of converters considered is comprised of Σ, a sequential composition operation $\circ : \Sigma \times \Sigma \to \Sigma$, a parallel composition

operation $\mid : \Sigma \times \Sigma \to \Sigma$, an identity converter[9] id, and is denoted $< \Sigma, \circ, \mid, \mathsf{id} >$. We assume that both binary operations $\circ$ and $\mid$ satisfy the following equalities. For all resources $\mathbf{S}$ and $\mathbf{T}$, and for all converters α and β,

$$(\alpha \circ \beta)\mathbf{S} = \alpha(\beta\mathbf{S}),$$
$$(\alpha \mid \beta)(\mathbf{S} \| \mathbf{T}) = (\alpha\mathbf{S}) \| (\beta\mathbf{T}),$$

and similarly at the right interface.

The following lemma, stated without proof, shows some simple properties on the set of stateless converters defined in Section 3.

Lemma 3. *Let Σ_{s} be the set of stateless converters in Σ. Then, $< \Sigma_{\mathsf{s}}, \circ, \mid, \mathsf{id} >$ is a sub-algebra of $< \Sigma, \circ, \mid, \mathsf{id} >$, i.e.,*

1. $\Sigma_{\mathsf{s}} \circ \Sigma_{\mathsf{s}} \subseteq \Sigma_{\mathsf{s}}$,
2. $\Sigma_{\mathsf{s}} \mid \Sigma_{\mathsf{s}} \subseteq \Sigma_{\mathsf{s}}$,
3. $\mathsf{id} \in \Sigma_{\mathsf{s}}$.

C.3 Distinguishers

Recall that a distinguisher $\mathbf{D}$ is a system that connects to all interfaces of a resource $\mathbf{R}$ and outputs at a separate interface a single bit B. Thus, for a 2-interface resource $\mathbf{R}$, $\mathbf{DR}$ is a random experiment with a single bit output. For a converter α, consider the random experiment[10] $\mathbf{D}(\alpha^i\mathbf{R})$, where $i \in \{\mathsf{left}, \mathsf{right}\}$. In order to see the system formed by the distinguisher $\mathbf{D}$ which applies the converter α at the i interface of the connected system as a new distinguisher interacting with $\mathbf{R}$, we need the assumption that $\mathcal{D}$ is closed under emulation of a converter, that is

$$(\forall \mathbf{D} \in \mathcal{D})(\forall \alpha \in \Sigma)(\exists \mathbf{D}' \in \mathcal{D})(\forall \mathbf{R} \in \Phi_2) : \ \mathsf{P}^{\mathbf{D}(\alpha^i\mathbf{R})}(B = 1) = \mathsf{P}^{\mathbf{D}'\mathbf{R}}(B = 1).$$

Note that such $\mathbf{D}'$ can only depend on $\mathbf{D}$ and α^i, and will be denoted $\mathbf{D}\alpha^i$ from now on.

Similarly, for two 2-interface resources $\mathbf{S}$ and $\mathbf{T}$, consider the random experiment $\mathbf{D}(\mathbf{S}\|\mathbf{T})$. In order to consider the system formed by $\mathbf{D}$ which appends in parallel the resource $\mathbf{T}$ to the connected system as a valid distinguisher for $\mathbf{S}$, we need that $\mathcal{D}$ is closed under emulation of a resource, i.e,

$$(\forall \mathbf{D} \in \mathcal{D})(\forall \mathbf{T} \in \Phi_2)(\exists \mathbf{D}' \in \mathcal{D})(\forall \mathbf{S} \in \Phi_2) : \ \mathsf{P}^{\mathbf{D}(\mathbf{S}\|\mathbf{T})}(B = 1) = \mathsf{P}^{\mathbf{D}'\mathbf{S}}(B = 1).$$

Such $\mathbf{D}'$ can only depend on $\mathbf{D}$ and on $\mathbf{T}$ and will be denoted by $\mathbf{D}(\cdot\|\mathbf{T})$ from hereafter. Since $\|$ is not necessarily symmetric, we also have the following assumption on $\mathcal{D}$,

$$(\forall \mathbf{D} \in \mathcal{D})(\forall \mathbf{S} \in \Phi_2)(\exists \mathbf{D}' \in \mathcal{D})(\forall \mathbf{T} \in \Phi_2) : \ \mathsf{P}^{\mathbf{D}(\mathbf{S}\|\mathbf{T})}(B = 1) = \mathsf{P}^{\mathbf{D}'\mathbf{T}}(B = 1),$$

where such $\mathbf{D}'$ will be denoted by $\mathbf{D}(\mathbf{S}\|\cdot)$.

C.4 Memory-Aware Reducibility is Composable

We now state and prove the composition theorem for memory-aware reducibility which was only informally stated in Theorem 1.

Theorem 5. *Consider an algebra of resources $< \Phi_2, \| >$, an algebra of converters $< \Sigma, \circ, \mid, \mathsf{id} >$, and a class of distinguishers $\mathcal{D}$ closed under emulation of a converter and of a resource, as described above. Let $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{T}$ be resources such that $\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1, s_1} \mathbf{S}$ and $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon_2, s_2} \mathbf{T}$. Then we have:*

[9] id forwards all inputs and outputs

[10] We are using the following notation: $\mathbf{D}(\alpha^{\mathsf{left}}\mathbf{R}) := \mathbf{D}(\alpha\mathbf{R})$ and $\mathbf{D}(\alpha^{\mathsf{right}}\mathbf{R}) := \mathbf{D}(\mathbf{R}\alpha)$.

1. *For any resource* $\mathbf{U}$, $\mathbf{R}\|\mathbf{U} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S}\|\mathbf{U}$ *and* $\mathbf{U}\|\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{U}\|\mathbf{S}$,
2. $\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1+\varepsilon_2,s_1+s_2} \mathbf{T}$.

Proof. We first prove that

$$\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S} \implies \mathbf{R}\|\mathbf{U} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S}\|\mathbf{U}$$

for any resource $\mathbf{U}$. Recalling Definition 2, let us assume that $\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S}$, i.e.,

$$(\exists \pi \in \Sigma)(\forall \mathbf{D} \in \mathcal{D})(\exists \sigma \in \Sigma_{\mathsf{s}}) : \Delta^{\mathbf{D}}(\pi\mathbf{R}, [\mathbf{S}||\mathbf{M}_{s_1}]\sigma) \leq \varepsilon_1. \tag{4}$$

Let $\pi' := (\pi \mid \mathsf{id}) \in \Sigma$ and consider an arbitrary distinguisher $\mathbf{D}' \in \mathcal{D}$. Let $\mathbf{D} := \mathbf{D}'(\cdot\|\mathbf{U}) \in \mathcal{D}$. By (4), there exists a stateless simulator $\sigma \in \Sigma_{\mathsf{s}}$ such that $\Delta^{\mathbf{D}}(\pi\mathbf{R}, [\mathbf{S}||\mathbf{M}_{s_1}]\sigma) \leq \varepsilon_1$. Let $\sigma' := \mu \circ (\sigma \mid \mathsf{id})$, where μ is a converter relabelling sub-interfaces represented in Figure 4. Trivially $\mu \in \Sigma_{\mathsf{s}}$ and Lemma 3 implies that $\sigma' \in \Sigma_{\mathsf{s}}$. Then,

$$\begin{aligned}
\Delta^{\mathbf{D}'}(\pi'(\mathbf{R}||\mathbf{U}),\ [(\mathbf{S}||\mathbf{U})||\mathbf{M}_{s_1}]\sigma') &= \Delta^{\mathbf{D}'}((\pi \mid \mathsf{id})(\mathbf{R}||\mathbf{U}),\ [(\mathbf{S}||\mathbf{U})\|\mathbf{M}_{s_1}]\mu(\sigma \mid \mathsf{id})) \\
&= \Delta^{\mathbf{D}'}((\pi\mathbf{R}\|\mathbf{U}),\ ([\mathbf{S}\|\mathbf{M}_{s_1}]\|\mathbf{U})(\sigma \mid \mathsf{id})) \\
&= \Delta^{\mathbf{D}'}((\pi\mathbf{R}\|\mathbf{U}),\ [\mathbf{S}\|\mathbf{M}_{s_1}]\sigma\|\mathbf{U}) \\
&= \Delta^{\mathbf{D}'(\cdot\|\mathbf{U})}(\pi\mathbf{R},\ [\mathbf{S}||\mathbf{M}_{s_1}]\sigma) \\
&= \Delta^{\mathbf{D}}(\pi\mathbf{R},\ [\mathbf{S}||\mathbf{M}_{s_1}]\sigma) \\
&\leq \varepsilon_1.
\end{aligned}$$

Thus, we have

$$(\exists \pi' \in \Sigma)(\forall \mathbf{D}' \in \mathcal{D})(\exists \sigma' \in \Sigma_{\mathsf{s}}) : \Delta^{\mathbf{D}'}(\pi'(\mathbf{R}||\mathbf{U}), [(\mathbf{S}||\mathbf{U})||\mathbf{M}_{s_1}]\sigma') \leq \varepsilon_1,$$

that is, $\mathbf{R}\|\mathbf{U} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S}\|\mathbf{U}$, which concludes the proof.

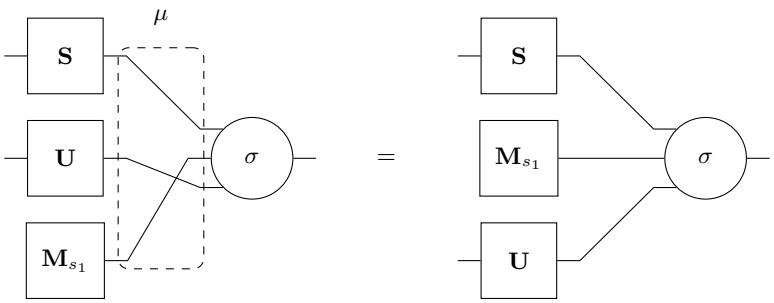


Fig. 4: Relabelling of sub-interfaces by the converter μ

The proof of $\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S} \implies \mathbf{U}\|\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{U}\|\mathbf{S}$ is symmetric and therefore omitted.

We now prove that

$$\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S} \wedge \mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon_2,s_2} \mathbf{T} \implies \mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_3,s_3} \mathbf{T}$$

where $\varepsilon_3 := \varepsilon_1 + \varepsilon_2$ and $s_3 := s_1 + s_2$. To that end, let us assume that both $\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{S}$ and $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon_2,s_2} \mathbf{T}$ hold, i.e.,

$$(\exists \pi_1 \in \Sigma)(\forall \mathbf{D}_1 \in \mathcal{D})(\exists \sigma_1 \in \Sigma_{\mathsf{s}}) : \Delta^{\mathbf{D}_1}(\pi_1\mathbf{R}, [\mathbf{S}||\mathbf{M}_{s_1}]\sigma_1) \leq \varepsilon_1 \tag{5}$$

and

$$(\exists \pi_2 \in \Sigma)(\forall \mathbf{D}_2 \in \mathcal{D})(\exists \sigma_2 \in \Sigma_{\mathsf{s}}) : \Delta^{\mathbf{D}_2}(\pi_2\mathbf{S}, [\mathbf{T}||\mathbf{M}_{s_2}]\sigma_2) \leq \varepsilon_2. \tag{6}$$

Let $\pi_3 := (\pi_2 \circ \pi_1) \in \Sigma$, and consider an arbitrary distinguisher $\mathbf{D}_3 \in \mathcal{D}$. Let $\mathbf{D}_1 := \mathbf{D}_3\pi_2^{\mathsf{left}} \in \mathcal{D}$. By (5), there exists a stateless simulator $\sigma_1 \in \Sigma_{\mathsf{s}}$ such that $\Delta^{\mathbf{D}_1}(\pi_1\mathbf{R}, [\mathbf{S}||\mathbf{M}_{s_1}]\sigma_1) \leq \varepsilon_1$. Let $\mathbf{D}_2 := \mathbf{D}_3[(\cdot\|\mathbf{M}_{s_1})\sigma_1] \in \mathcal{D}$. Similarly by (6), there exists a stateless simulator $\sigma_2 \in \Sigma_{\mathsf{s}}$ such that $\Delta^{\mathbf{D}_2}(\pi_2\mathbf{S}, [\mathbf{T}||\mathbf{M}_{s_2}]\sigma_2) \leq \varepsilon_2$. Let $\sigma_3 := (\sigma_2 \mid \mathsf{id}) \circ \sigma_1$, and note that by Lemma 3, $\sigma_3 \in \Sigma_{\mathsf{s}}$. Then,

$$
\begin{aligned}
\Delta^{\mathbf{D}_3}\left(\pi_3\mathbf{R},\ [\mathbf{T}\|\mathbf{M}_{s_3}]\sigma_3\right) &\overset{(a)}{\leq} \Delta^{\mathbf{D}_3}\left(\pi_3\mathbf{R},\ \pi_2[\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1\right) + \Delta^{\mathbf{D}_3}\left(\pi_2[\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1,\ [\mathbf{T}\|\mathbf{M}_{s_3}]\sigma_3\right) \\
&\overset{(b)}{=} \Delta^{\mathbf{D}_3}\left(\pi_2\pi_1\mathbf{R},\ \pi_2[\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1\right) + \Delta^{\mathbf{D}_3}\left([\pi_2\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1,\ [\mathbf{T}\|\mathbf{M}_{s_3}]\sigma_3\right) \\
&\overset{(c)}{=} \Delta^{\mathbf{D}_3\pi_2^{\mathsf{left}}}\left(\pi_1\mathbf{R},\ [\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1\right) + \Delta^{\mathbf{D}_3}\left([\pi_2\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1,\ [\mathbf{T}\|[\mathbf{M}_{s_2}\|\mathbf{M}_{s_1}]]\sigma_3\right) \\
&\overset{(d)}{=} \Delta^{\mathbf{D}_1}\left(\pi_1\mathbf{R},\ [\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1\right) + \Delta^{\mathbf{D}_3}\left([\pi_2\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1,\ [[\mathbf{T}\|\mathbf{M}_{s_2}]\|\mathbf{M}_{s_1}]\sigma_3\right) \\
&= \Delta^{\mathbf{D}_1}\left(\pi_1\mathbf{R},\ [\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1\right) + \Delta^{\mathbf{D}_3}\left([\pi_2\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1,\ [[\mathbf{T}\|\mathbf{M}_{s_2}]\|\mathbf{M}_{s_1}](\sigma_2 \mid \mathsf{id}) \circ \sigma_1\right) \\
&= \Delta^{\mathbf{D}_1}\left(\pi_1\mathbf{R},\ [\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1\right) + \Delta^{\mathbf{D}_3}\left([\pi_2\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1,\ [[\mathbf{T}\|\mathbf{M}_{s_2}]\sigma_2\|\mathbf{M}_{s_1}]\sigma_1\right) \\
&= \Delta^{\mathbf{D}_1}\left(\pi_1\mathbf{R},\ [\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1\right) + \Delta^{\mathbf{D}_3[(\cdot\|\mathbf{M}_{s_1})\sigma_1]}\left(\pi_2\mathbf{S},\ [\mathbf{T}\|\mathbf{M}_{s_2}]\sigma_2\right) \\
&= \Delta^{\mathbf{D}_1}\left(\pi_1\mathbf{R},\ [\mathbf{S}\|\mathbf{M}_{s_1}]\sigma_1\right) + \Delta^{\mathbf{D}_2}\left(\pi_2\mathbf{S},\ [\mathbf{T}\|\mathbf{M}_{s_2}]\sigma_2\right) \\
&\leq \varepsilon_1 + \varepsilon_2,
\end{aligned}
$$

where (a) follows from the triangle inequality; (b) follows from the fact that $\mathbf{M}_{s_1}$ has a dummy left interface, and thus $\pi_2[\mathbf{S}\|\mathbf{M}_{s_1}] = [\pi_2\mathbf{S}\|\mathbf{M}_{s_1}]$; (c) follows from the fact that a memory of s_3 bits can be divided into two independent memories of s_1 and s_2 bits, i.e., $\mathbf{M}_{s_3} = \mathbf{M}_{s_2}\|\mathbf{M}_{s_1}$; and (d) follows from the associativity of $\|$. Note that if $\|$ were not associative (which in full generality could be the case), (d) would still hold upon a trivial modification of σ_2 (this could be done by considering a converter $\sigma_2' := \mu'\sigma_2$, where μ', similarly to μ in the previous part of the proof, interleaves the sub-interfaces in such a way that associativity is guaranteed), we avoid this technicality here.

Thus, we have

$$(\exists\pi_3 \in \Sigma)(\forall\mathbf{D}_3 \in \mathcal{D})(\exists\sigma_3 \in \Sigma_{\mathsf{s}}) : \Delta^{\mathbf{D}_3}(\pi_3\mathbf{R}, [\mathbf{T}||\mathbf{M}_{s_3}]\sigma_3) \leq \varepsilon_3,$$

that is, $\mathbf{R} \xrightarrow[\mathrm{m}]{\varepsilon_3, s_3} \mathbf{T}$, which concludes the proof.

□