Resource-Restricted Indifferentiability*

Grégory Demay, Peter Gaži, Martin Hirt, and Ueli Maurer

Department of Computer Science, ETH Zurich, Switzerland
{gregory.demay,peter.gazi,hirt,maurer}@inf.ethz.ch

Abstract. A major general paradigm in cryptography is the following argument: Whatever an adversary could do in the real world, it could just as well do in the ideal world. The standard interpretation of "just as well" is that the translation from the real to the ideal world, usually called a simulator, is achieved by a probabilistic polynomial-time algorithm. This means that a polynomial blow-up of the adversary's time and memory requirements is considered acceptable.

In certain contexts this interpretation of "just as well" is inadequate, for example if the concrete amount of memory used by the adversary is relevant. The example of Ristenpart et al. (Eurocrypt 2011), for which the original indifferentiability notion introduced by Maurer et al. (Eurocrypt 2004) is shown to be insufficient, turns out to be exactly of this type. It requires a fine-grained statement about the adversary's memory capacity, calling for a generalized treatment of indifferentiability where specific resource requirements can be taken into account by modeling them explicitly.

We provide such treatment and employ the new indifferentiability notion to prove lower bounds on the memory required by any simulator in a domain extension construction of a public random function. In particular, for simulators without memory, even domain extension by a single bit turns out to be impossible. Moreover, for the construction of a random oracle from an ideal compression function, memory roughly linear in the length of the longest query is required. This also implies the impossibility of such domain extension in any multi-party setting with potential individual misbehavior by parties (i.e., no central adversary).

* A preliminary version of this paper appears in the proceedings of EUROCRYPT 2013. This is the full version.

1 Introduction

1.1 Simulation-Based Security

The so-called "real world – ideal world" paradigm is underlying all current cryptographic frameworks aiming for composable security statements. Using the language of [MRH04,MR11], the purpose of a protocol is to construct an "ideal" resource (which is secure by definition) from "real" resources assumed to be available. The security of such a construction is then argued by showing that if some misbehaving entity (adversary) deviates from the prescribed protocol in the real world, it cannot achieve anything more than what would also be possible in the ideal world. Since the ideal resource is considered secure by definition, any such action is seen as harmless, thus implying the security of the protocol using the real resources.

The translation of the adversarial actions from the real world to the ideal world is described by exhibiting an algorithm performing it, called a *simulator*. The above argument for the real construction being as secure as the ideal resource is then valid as long as we assume that the adversary can, in addition to executing its attack, also translate it into the ideal-world setting by performing the job of the simulator itself. Simulators are typically modeled as probabilistic polynomial-time (PPT) Turing machines, which implies also polynomial memory (the range of the tape that can be accessed within this time) and randomness limitations. This potentially leads to a polynomial blow-up of the attack's resource requirements when translated from the real to the ideal world.

The implicit step of considering this overhead acceptable is hard-coded into most of the existing frameworks, such as the universal composability [Can01], indifferentiability [MRH04] and reactive simulatability [BPW04]. It is appropriate in most natural settings and hence the results in the above-mentioned frameworks 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 [RSS11] in the context of indifferentiability, considering the setting of auditable storage. Before we introduce our contributions, let us briefly review both the indifferentiability notion and the example from [RSS11].

1.2 The Case of Indifferentiability

Indifferentiability was introduced in [MRH04] as a generalization of indistinguishability for settings where some access to the internal state of the considered resources is available publicly, within reach of any potential attacker. The framework comes with a composition theorem loosely interpreted as saying that an ideal resource 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 [CDMP05]. Many existing cryptographic constructions are proven secure in the random oracle model [BR93], but once we instantiate the random oracle (RO) by an existing cryptographic hash function H, such a proof can be seen at most as a heuristic argument towards the security of the construction [CGH98]. However, if one uses a hash function construction H^f that was proven indifferentiable from a RO 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, a more compact object that is 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 [BDPVA08a]) enjoy such a proof (see e.g. [BDPVA08b,CN08,DRRS09,DRS09,AMP10]).

Storage-Auditing Scenario from [RSS11]. Re-examining the guarantees provided by indifferentiability, in [RSS11] the authors present an example of a two-party protocol for storage verification. 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 RO, it is clearly impossible for a malicious server to pass the verification without actually storing the user's data. However, as observed in [RSS11] this is no longer true if the RO is replaced by a particular iterative construction with an underlying ideal compression function, even though this construction is known to be indifferentiable from a RO. This puts in question the meaning of an indifferentiability proof as a security argument relevant in all possible contexts.

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 RO 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

Our contributions are three-fold. First, we introduce a new formalism based on abstract cryptography (AC, [MR11]), allowing a fine-grained modelling of resource requirements, necessary to capture problems such as the one described above. Second, we apply this new formalism to the problem of domain extension of public random functions and prove lower bounds on the memory needed by any simulator in this type of constructions. And finally, we investigate the consequences of these bounds for settings with multiple parties that may potentially deviate from the prescribed behavior in an uncoordinated manner. We proceed by a more detailed description of all three parts.

Memory-Aware Reducibility. In Section 3 we introduce the notion of *memory-aware reducibility* that is derived from reducibility[1] in the classical indifferentiability setting as given in [MRH04,MR11], but does not allow the memory requirements of the simulator to be "swept under the rug", requiring only that they are polynomial. In accordance with the spirit of the AC framework that is used to formalize it, our notion requires any memory necessary for the simulator to be explicitly modeled as a part of the ideal resource; with the intuitive meaning that the real construction is provably as good as the ideal resource as long as we assume that the adversary has the necessary amount of memory available. We also give a composition theorem for our new notion.

An independent approach to analyzing the complexity of the simulator in an indifferentiability statement appeared recently in [DRST12], where the authors focus on the number of queries the simulator issues per invocation. To the best of our knowledge, our work is the first one pointing out the importance of the simulator's memory requirements. However, we stress that the applicability of our approach goes beyond modeling memory, extending also to other resources such as computational power or randomness, would the investigated setting require it.

[1] The term "reducibility" is used in [MRH04] and, for consistency, also throughout this paper. It is to be understood in the same sense as the term "construction" used above, but the viewpoint is reversed. To construct S from R means the same as to reduce (the need for) S to (the need for) R.

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 RO $\mathbf{R}^{*,n}\colon \{0,1\}^* \to \{0,1\}^n$ 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 [CDMP05]. 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 the 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 RO, with the obvious transition of ℓ denoting the length of the longest query issued to the RO. This answers negatively the open question of the existence of such a construction using no simulator memory asked in [RSS11]. As another consequence, we also obtain the irreducibility of the RO to the ideal cipher with respect to stateless simulators, in contrast to the equivalence of these two ideal primitives with respect to classical indifferentiability [CDMP05,CPS08,HKT11].

Random Oracles Used by Multiple Parties. The impossibility results described above have some intriguing consequences for the setting where a RO 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 AC framework, 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 essentially 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, for settings where one cannot assume a central adversary coordinating all the actions of the misbehaving parties, no secure construction of a RO from an ideal compression function exists. This might be relevant in the contexts of *rational cryptography* [HT04], *incoercible computation* [CG96], *receipt-free voting* [BT94] or *collusion-free computation* [LMs05,AKL+09] and its recent composable variants [AKMZ12,CV12]. We formalize the above argument in Section 5 as an illustration of the impact of our results.

2 Preliminaries

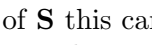
Basic Notation. We denote sets by calligraphic letters or capital greek letters (e.g. $\mathcal{X}, \Sigma$) and their cardinalities by $|\mathcal{X}|, |\Sigma|$. For a superset clear from the context, we denote the complement of a set $\mathcal{X}$ by $\overline{\mathcal{X}}$. Throughout the paper all 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 [MR11,Mau11] to which we give here a self-contained introduction, partly following the exposition given in [MRT12]. 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[2] 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 the interfaces of $\mathbf{S}\|\mathbf{T}$ allows to access the corresponding interface of both subsystems $\mathbf{S}$ and $\mathbf{T}$. Two converters ψ and ϕ can also be composed: either sequentially, obtaining a converter $\psi \circ \phi$ such that $(\psi \circ \phi)\mathbf{S} = \psi(\phi\mathbf{S})$; or in parallel, obtaining $\psi|\phi$ such that $(\psi|\phi)(\mathbf{S}\|\mathbf{T}) = (\psi\mathbf{S})\|(\phi\mathbf{T})$. The application of composed converters to the public interface works in an analogous way. The term id refers to the "identity converter" that forwards all inputs and outputs, we always assume $\mathsf{id} \in \Sigma$. 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.

[2] In this paper we sometimes also use the term "resources" in a more informal way to refer to computational power, memory, etc. This should cause no confusion, since these resources could also be formalized in the sense of the notion introduced above.

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

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 [Mau02], 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 restrict our considerations to 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 [MRH04] 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. For given sets Σ and $\mathcal{D}$ of converters and distinguishers, respectively, 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. The two settings distinguished are depicted in Fig. 1.

Note that by choosing the sets Σ and $\mathcal{D}$, this definition covers both information-theoretic and computational indifferentiability; moreover, one could also easily derive an asymptotic definition. These remarks are also true for all other reducibility notions presented below.

Strong Indifferentiability. For given sets Σ and $\mathcal{D}$ 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 [CDMP05] showing security of MD-variants) typically prove this stronger type of statement by exhibiting a simulator that does not

depend on the distinguisher. A detailed discussion of the relationship between these two forms of simulatability in various formalisms can be found in [HU05,Can01], 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 [CDMP05] 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. 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 *uses no memory between answering outer queries*, i.e., its 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. However, it might of course be randomized, using fresh randomness at every invocation. This is captured by the following formal definition.

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_{sl} 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 require it to be stateless and only use the memory provided by the ideal resource, leading to the following formalism (broadly denoted as *memory-aware reducibility*).

Definition 2. *For given sets Σ and $\mathcal{D}$ of converters and distinguishers, respectively, 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*

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

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 he 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; moreover, the transition to an asymptotic definition would be straightforward. Alongside the notion of reducibility, one could also explicitly define the underlying notion of *memory-aware indifferentiability* that would only consider the trivial protocol $\pi = \mathsf{id}$, leading to the same relationship between indifferentiability and the respective reducibility as in the classical case [MRH04]. Since our results make use of the reducibility notion, we omit this step.

In 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 $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon} \mathbf{T}$. In this special case, the underlying indifferentiability notion is technically equivalent to the notion of reset indifferentiability introduced in [RSS11]: 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 them forward to define a security notion by themselves, but only as a tool for modeling the memory requirements of the simulator explicitly.

Composability. The formalism of memory-aware reducibility given above leads to statements that are composable under some natural closure assumptions on the sets Σ and $\mathcal{D}$ of converters and distinguishers considered. Here we only state the respective composition theorem informally.

Theorem 1 (Informal). *Let Σ be closed under both sequential composition $\circ$ and parallel composition $|$ and let $\mathcal{D}$ be closed under the emulation of any converter and of any resource. Let $\mathbf{S}$, $\mathbf{T}$ and $\mathbf{V}$ be resources such that $\mathbf{S} \xrightarrow[\mathrm{m}]{\varepsilon_1,s_1} \mathbf{T}$ and $\mathbf{T} \xrightarrow[\mathrm{m}]{\varepsilon_2,s_2} \mathbf{V}$. Then we have:*

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

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

We now investigate the amount 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, thus solving an open problem introduced in [RSS11].

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

1: $X \xleftarrow{\$} \{0,1\}^\ell$
2: **query** $Y := [\mathbf{S}]_L(X)$
3: **simulate** π **to evaluate** $\hat{Y} := \pi(X)$
   answer new inner queries by querying $[\mathbf{S}]_R$
   answer repeated inner queries consistently
4: **if** $Y = \hat{Y}$ **then**
5: **return** 1
6: **return** 0

Fig. 2: The distinguisher $\mathbf{D}$ for the proof of Theorem 2.

Theorem 2. *Assume that for any $\pi \in \Sigma$, the distinguisher $\mathbf{D}$ constructed from π according to Fig. 2 is present in $\mathcal{D}$. Then any reduction $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{\ell,r}$ with[3] $r \geq 2$ and $\varepsilon \leq 0.04$ satisfies[4]*

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

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 $\mathbf{D}$.

Proof Overview. 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 the $\{0,1\}^m$-queries issued by π. Moreover, it never repeats a query to the right interface: in case the simulated protocol π would issue a repeated query, it is answered as before. We will refer to this modified (simulated) protocol π as π'; note that $\mathbf{D}$ is capable of this modification since it can keep the history of query-answer pairs in its state. 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 it is natural to expect that this simple distinguisher $\mathbf{D}$ is present in any reasonable distinguisher class.

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 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

[3] 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.

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

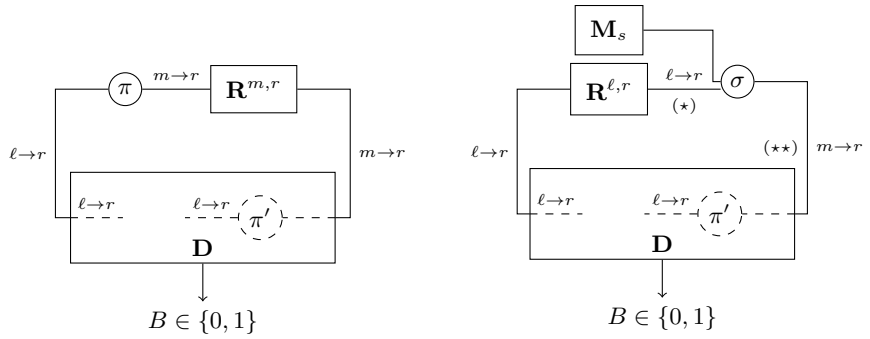


Fig. 3: The real and the ideal setting for the proof of Theorem 2. The notation $i \to o$ describes an interface that accepts queries from $\{0,1\}^i$ and responds with elements from $\{0,1\}^o$.

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 $(\star)$. To capture the randomness involved in the ideal distinguishing experiment, we denote by R_w the (fresh, independent) internal randomness used by σ when it is answering an outer query $w \in \{0,1\}^m$ *for the first time*[5] and let $R_\sigma := \{R_w\}_{w\in\{0,1\}^m}$. Moreover, let $R_{\mathbf{R}}$ denote the overall randomness of the ideal resource $\mathbf{R}^{\ell,r}$, i.e., its function table. For a fixed randomness $R_\sigma = r_\sigma$ and $R_{\mathbf{R}} = r_{\mathbf{R}}$ where $r_\sigma = \{r_w\}_{w\in\{0,1\}^m}$, let us denote by $f(w, z, r_w, r_{\mathbf{R}}) \subseteq \{0,1\}^\ell$ 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 with the available memory $\mathbf{M}_s$ containing value $z \in \{0,1\}^s$, using randomness r_w while the responses from $\mathbf{R}^{\ell,r}$ are determined by $r_{\mathbf{R}}$. Since the random variables R_w and $R_{\mathbf{R}}$ represent the only sources of randomness in this evaluation, f is a well-defined deterministic mapping and by our assumption $|f(w, z, r_w, r_{\mathbf{R}})| \le t$ for all possible inputs. Let us define $\mathcal{S}_{r_\sigma, r_{\mathbf{R}}}$ to be the set of all possible queries under all inputs (w, z) for this fixed randomness $(r_\sigma, r_{\mathbf{R}})$, i.e.,

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

then we have $|\mathcal{S}_{r_\sigma, r_{\mathbf{R}}}| \le 2^{m+s+\log t}$ for any $(r_\sigma, r_{\mathbf{R}})$. Since $X \in \{0,1\}^\ell$ was chosen at random and independently from $\mathcal{S}_{R_\sigma, R_{\mathbf{R}}}$, we obtain $\mathsf{P}(X \in \mathcal{S}_{R_\sigma, R_{\mathbf{R}}}) \le 2^{m+s+\log t}/2^\ell = 2^{m+s+\log t-\ell}$. 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 $\ell - m \le s + \lceil \log t \rceil$.

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 the available 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

[5] Formally, one can imagine σ being replaced by a stateful simulator that chooses all random variables R_w at the beginning and then uses it when the query w arrives for the first time. This view does not change the outcomes of the experiment.

$\mathbf{R}^{\ell,r}$ and the internal randomness of σ used during this invocation. We can now define T to be the table containing a sample of $\sigma_z(w)$ and $Z'(w,z)$ for all possible w and z, formally

$$T := \left\{ \left(\sigma_z(w), Z'(w,z)\right) \right\}_{(w,z)\in\{0,1\}^m\times\{0,1\}^s}.$$

Then T can be seen as a random variable distributed over $\{0,1\}^{(r+s)\cdot 2^{m+s}}$ and is again determined by the function table of $\mathbf{R}^{\ell,r}$ and the randomness used by 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 its queries to σ using the respective value from T instead (recall that π' asks each query at most once). 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 [Fan61] 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 $\mathbf{R}^{\ell,r}(X)$ as follows:

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

Now if $\ell - m > s + \lceil \log s \rceil$ then since m, s, ℓ are integers we get that $2^{m+s-\ell} \leq 1/2$ and also $(s/r)\cdot 2^{m+s-\ell} \leq 1/2r$. 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 lengths of hashed messages.

In the statement below we shall again consider the distinguisher given in Fig. 2, this time for the setting of the reduction $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{*,r}$. To emphasize that it chooses the value X from the set $\{0,1\}^\ell \subseteq \{0,1\}^*$, we shall denote it $\mathbf{D}_\ell$, note that it again implicitly depends on a protocol π. One could give a similar statement also for a distinguisher asking several private queries and using the public interface to evaluate the protocol π on the longest one.

Corollary 1. *If for every $\pi \in \Sigma$ the distinguisher $\mathbf{D}_\ell$ described above is present in $\mathcal{D}$ 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. For the more general case $\mathbf{R}^{m,r} \xrightarrow[\mathrm{m}]{\varepsilon,s} \mathbf{R}^{*,n}$ we still have $s \geq \ell - m - \lceil \log t \rceil$ under the same assumptions.

Proof. The argument is analogous to the proof of Theorem 2 with a trivial modification to account for $\mathbf{R}^{*,r}$ as the ideal resource: in part $(\star\star)$ the random variable F now only stands for the part of the function table of $\mathbf{R}^{*,r}$ corresponding to inputs of length ℓ. The second claim holds since the analysis of the 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 [CDMP05] to show its indifferentiability from a random oracle would use (without optimizations) roughly $(1 + r/m) \cdot \ell$ bits of memory to answer all queries of the distinguisher $\mathbf{D}_\ell$ considered in Corollary 1, while always asking at most one query to the ideal primitive to answer a single query itself. Our result implies that for any indifferentiable domain extension construction, if the respective simulator is of this single-query form then it needs at least $\ell - m$ bits of memory. Since typically $\ell \gg m$, this implies that the simulator given in [CDMP05] has essentially optimal memory requirements within this class (i.e., linear in ℓ).

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 [CDMP05] that demonstrate the possibility of such reduction with respect to stateful simulators.

For the proof of the following corollary, recall that $\mathbf{E}^{k,n}$ allows both encryption and decryption queries under an arbitrary key, hence taking an input of $k + n + 1$ bits. Taking this into account, the argument is analogous to the part $(\star\star)$ in the proof of Theorem 2 and is hence omitted.

Corollary 2. *If for every $\pi \in \Sigma$ and for $\ell = k + n + \lceil \log(n/r) \rceil + 2$ the distinguisher $\mathbf{D}_\ell$ considered in Corollary 1 is present in $\mathcal{D}$, then any reduction $\mathbf{E}^{k,n} \xrightarrow[\mathrm{m}]{\varepsilon} \mathbf{R}^{*,r}$ has to satisfy $\varepsilon \geq 0.1$.*

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

As a particular application of our results, in this section we present some interesting consequences of the lower bound on simulator memory for domain extension of public 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. For example, when constructing a random oracle from an ideal compression function, all honest parties should use the same protocol π and moreover, it should 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. In the ideal world the resource (a random oracle) is stateful, with the 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 [CDMP05] 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, using a stateless protocol is in general 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 [MR11]. 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. [Can01,BPW04]) 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. For some understood Σ and $\mathcal{D}$, 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 (including the same randomness). 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: it turns out that using stateless protocols *and* simulators is sufficient. 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* $\hat{\mathbf{S}}_2 \xrightarrow[\text{si}]{\varepsilon} \hat{\mathbf{T}}_2$ *for a class of converters* Σ *and distinguishers* $\mathcal{D}$*, and both the protocol* π *and the simulator* σ *used in this reduction are stateless, then we have* $\hat{\mathbf{S}}_n \xrightarrow[\text{AC}]{\varepsilon} \hat{\mathbf{T}}_n$ *for the class of converters* Σ *and 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 the domain extension for random functions in the n-party case[6] (i.e., the reduction $\hat{\mathbf{R}}_n^{m,r} \xrightarrow[\text{AC}]{\varepsilon} \hat{\mathbf{R}}_n^{\ell,r}$ with $\ell > m$). In this case using inherently stateful simulators σ_i would also lead to inconsistencies, for the same reason as described for the protocols π_i. Note that we cannot claim that such a reduction cannot be achieved using a stateful simulator, since its stateful behavior might not manifest in the distinguishing experiment. However, any such stateful simulator could be replaced by a stateless one without significant impact, as formalized in Lemma 1 below. Later we observe that the simulators cannot be stateless (for the same reason as in the indifferentiability case), leading to the impossibility result.

For the statement of Lemma 1, we will assume that the set of distinguishers $\mathcal{D}$ satisfies a simple closure property. For any $\mathbf{D} \in \mathcal{D}$ asking queries only to interfaces 1 and 2 let us consider a derived distinguisher $\mathbf{D}_{(i)}$ that proceeds in the same way as $\mathbf{D}$ but it also counts its queries to interface 2 and as soon as its i-th such query occurs, it asks the same query also to interface 3. At the end, $\mathbf{D}_{(i)}$ will output 1 if and only if the response to its i-th query to interface 2 was distinct from the response to the same query to interface 3. We assume $\mathbf{D}_{(i)} \in \mathcal{D}$ for all $\mathbf{D} \in \mathcal{D}$ and all $1 \leq i \leq q$ where q is an upper bound on the number of $\mathbf{D}$'s queries to interface 2.

Lemma 1. *Consider some fixed* $n \geq 3$, $\ell > m$ *and some fixed sets of converters* Σ *and distinguishers* $\mathcal{D}$ *satisfying the property given above. Assume that there exists a reduction* $\hat{\mathbf{R}}_n^{m,r} \xrightarrow[\text{AC}]{\varepsilon} \hat{\mathbf{R}}_n^{\ell,r}$ *via a tuple of protocols* $\hat{\pi} = (\pi_1, \ldots, \pi_n)$ *and simulators* $\hat{\sigma} = (\sigma_1, \ldots, \sigma_n)$. *Then there also exists a tuple of simulators* $\hat{\sigma}' = (\sigma_1, \sigma_2', \sigma_3, \ldots, \sigma_n)$ *such that* σ_2' *is stateless and for every distinguisher* $\mathbf{D} \in \mathcal{D}$ *accessing only interfaces* 1 *and* 2 *we have* $\Delta^{\mathbf{D}}(\hat{\pi}_{\{1\}}\hat{\mathbf{R}}_n^{m,r}, \hat{\sigma}'_{\{1\}}\hat{\mathbf{R}}_n^{\ell,r}) \leq (q+1)\varepsilon$ *where* q *is an upper bound on the number of its queries to interface* 2.

Proof. Let us consider the case in equation (3) where $\mathcal{P} = \{1\}$, hence any distinguisher would in the ideal experiment interact with a local simulator σ_i on each interface $i > 1$. Let $\mathbf{D}$ be a

[6] In the rest of the section we will use symbols such as $\mathbf{R}^{m,r}$ to refer to the single-interface resource and use the introduced notation to explicitly state the number of interfaces we want to consider (e.g., $\hat{\mathbf{R}}_n^{m,r}$).

distinguisher that only asks queries to interfaces 1 and 2. Informally, we show that if we replace σ_2 by a simulator σ_2' that treats each query in the same way as the simulator σ_3 would treat its *first* query, this change will most likely remain unnoticed by any such $\mathbf{D}$. Moreover, since the behavior of this σ_2' on a particular query does not depend on any previous interaction, it is stateless in the sense of Definition 1. Note that we do not assume that this σ_2' belongs to the set Σ of converters considered for our reductions, but this will not be required by the further use of Lemma 1.

For any $i \in \{1, \ldots, q\}$ let us consider the derived distinguisher $\mathbf{D}_{(i)}$ described above. It will never output 1 in the real experiment, since there both interfaces 2 and 3 provide access to the same function. Therefore, the advantage achieved by $\mathbf{D}_{(i)}$ is equal to the probability that the two responses it compares in the ideal experiment are distinct. Since $\mathbf{D}_{(i)} \in \mathcal{D}$ we know that its advantage is at most ε, hence in the ideal experiment this inconsistency can occur with probability at most ε. This means that for each $i \in \{1, \ldots, q\}$, if the i-th response of σ_2 was replaced by a response generated by σ_2' instead, the outcome of the whole distinguishing experiment would change with probability at most ε. Applying the union bound, we get that if we replace *all* responses of σ_2 by those of σ_2', the transcript of the whole distinguishing experiment will change with probability at most $q\varepsilon$. Finally, since the advantage achieved by $\mathbf{D}$ before this change was at most ε, it will be at most $(q+1)\varepsilon$ afterwards. □

Let us now denote by $\hat{\mathbf{D}}$ the distinguisher given in Fig. 2 (implicitly parametrized by a converter $\pi \in \Sigma$) modified into the n-interface setting as follows: it uses interface 1 for all its (originally) private-interface queries, while using interface 2 for all public-interface queries. If $\hat{\mathbf{D}} \in \mathcal{D}$ then the upper bound given in Lemma 1 applies to $\Delta^{\hat{\mathbf{D}}}(\hat{\pi}_{\{1\}}\hat{\mathbf{R}}_n^{m,r}, \hat{\sigma}'_{\overline{\{1\}}}\hat{\mathbf{R}}_n^{\ell,r})$. On the other hand, since $\ell > m$ and σ_2' uses no memory, following the proof of Theorem 2 we also obtain that $\Delta^{\hat{\mathbf{D}}}(\hat{\pi}_{\{1\}}\hat{\mathbf{R}}_n^{m,r}, \hat{\sigma}'_{\overline{\{1\}}}\hat{\mathbf{R}}_n^{\ell,r}) > 0.04$. Combining these observations we get the following corollary.

Corollary 3. *Consider some fixed $n \geq 3$, $r \geq 2$, $\ell > m$ and sets of converters Σ and distinguishers $\mathcal{D}$ satisfying the properties required in Lemma 1 and additionally such that for each $\pi \in \Sigma$ the respective $\hat{\mathbf{D}}$ is in $\mathcal{D}$. If there exists a reduction $\hat{\mathbf{R}}_n^{m,r} \xrightarrow[\text{AC}]{\varepsilon} \hat{\mathbf{R}}_n^{\ell,r}$ via a tuple of protocols $\hat{\pi} = (\pi_1, \ldots, \pi_n)$ then $\varepsilon > 0.04/(p+1)$ where p is an upper bound on the number of $\{0,1\}^m$-queries the protocol π_1 used for this reduction needs to evaluate on one $\{0,1\}^\ell$-input.*

Hence by the above result it is impossible to extend the domain of a public random function even by a single bit 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). The above result extends trivally also to the case of infinite domain extension, i.e., the construction of a public random oracle from an ideal compression function. This is in contrast to the two-party indifferentiability setting (with several constructions that achieve this 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 Conclusions

We have introduced a general way of treating simulation-based security in situations where a more fine-grained quantification of a certain resource is necessary. Focusing on indifferentiability as the

security notion in question and memory as the resource, this also allowed us to explain from a different perspective the unexpected security failure of the protocol given in [RSS11] when used with the construction chop-MD.

We proceeded by giving lower bounds on the required simulator memory for any reduction of a public random oracle to a public random function, showing that memory roughly linear in the length of the longest query is necessary, and that with no memory even domain extension by a single bit becomes impossible.

Finally, we applied our result to the setting where the random oracle is used by multiple parties with no central adversary to coordinate potential misbehavior. We showed that special care must be taken in such settings when replacing the random oracle by a construction using an ideal compression function, since no construction secure in *every* such setting exists.

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. We would also like to thank the anonymous reviewers for pointing out a bug in an earlier version of our proof.

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

[AKL+09] 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, *Advances in Cryptology — CRYPTO 2009*, volume 5677 of *LNCS*, pages 524–540. Springer Berlin Heidelberg, 2009.

[AKMZ12] Joël Alwen, Jonathan Katz, Ueli Maurer, and Vassilis Zikas. Collusion-preserving computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology — CRYPTO 2012*, volume 7417 of *LNCS*, pages 124–143. Springer Berlin Heidelberg, 2012.

[AMP10] 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 *LNCS*, pages 88–105. Springer Berlin Heidelberg, 2010.

[BDPVA08a] Guido Bertoni, Joan Daemen, Michal Peeters, and Gilles Van Assche. Keccak specifications. Submission to NIST (Round 1), 2008.

[BDPVA08b] 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 *LNCS*, pages 181–197. Springer Berlin Heidelberg, 2008.

[BPW04] Michael Backes, Birgit Pfitzmann, and Michael Waidner. A general composition theorem for secure reactive systems. In Moni Naor, editor, *Theory of Cryptography — TCC 2004*, volume 2951 of *LNCS*, pages 336–354. Springer Berlin Heidelberg, 2004.

[BR93] 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.

[BT94] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In *STOC '94: Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 544–553, New York, NY, USA, 1994. ACM.

[Can01] Ron Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *FOCS '01: Proceedings of the 42nd IEEE Annual Symposium on Foundations of Computer Science*, pages 136–145. IEEE Computer Society Press, Oct. 2001. Full version at http://eprint.iacr.org/2000/067.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård Revisited: How to Construct a Hash Function. In Victor Shoup, editor, *Advances in Cryptology — CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer Berlin Heidelberg, 2005.

[CG96] Ran Canetti and Rosario Gennaro. Incoercible multiparty computation. In *FOCS '96: Proceedings of the 37th IEEE Annual Symposium on Foundations of Computer Science*, pages 504–513. IEEE Computer Society, 1996.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *STOC '98: Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 209–218. ACM, 1998.

[CN08] Donghoon Chang and Mridul Nandi. Improved Indifferentiability Security Analysis of chopMD Hash Function. In Kaisa Nyberg, editor, *Fast Software Encryption*, volume 5086 of *LNCS*, pages 429–443. Springer Berlin Heidelberg, 2008.

[CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In *Advances in Cryptology — CRYPTO 2008*, volume 5157 of *LNCS*, pages 1–20. Springer Berlin Heidelberg, 2008.

[CV12] Ran Canetti and Margarita Vald. Universally composable security with local adversaries. In Ivan Visconti and Roberto De Prisco, editors, *SCN*, volume 7485 of *LNCS*, pages 281–301. Springer Berlin Heidelberg, 2012.

[DRRS09] 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 *LNCS*, pages 104–121. Springer Berlin Heidelberg, 2009.

[DRS09] Yevgeniy Dodis, Thomas Ristenpart, and Thomas Shrimpton. Salvaging Merkle-Damgård for Practical Applications. In Antoine Joux, editor, *Advances in Cryptology — EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 371–388. Springer Berlin Heidelberg, 2009.

[DRST12] Yevgeniy Dodis, Thomas Ristenpart, John Steinberger, and Stefano Tessaro. To Hash or Not to Hash Again? (In)Differentiability Results for H 2 and HMAC. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology — CRYPTO 2012*, volume 7417 of *LNCS*, pages 348–366. Springer Berlin Heidelberg, 2012.

[Fan61] Robert Fano. *Transmission of Information: A Statistical Theory of Communications*. The MIT Press, Cambridge, MA, 1961.

[HKT11] Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. STOC '11: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing, pages 89–98, New York, NY, USA, 2011. ACM.

[HT04] Joseph Halpern and Vanessa Teague. Rational secret sharing and multiparty computation. In *STOC '04: Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 623–632, New York, NY, USA, 2004. ACM.

[HU05] Dennis Hofheinz and Dominique Unruh. Comparing two notions of simulatability. In Joe Kilian, editor, *Theory of Cryptography — TCC 2005*, volume 3378 of *LNCS*, pages 86–103. Springer Berlin Heidelberg, 2005.

[LMs05] Matt Lepinksi, Silvio Micali, and abhi shelat. Collusion-free protocols. In *STOC '05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 543–552, New York, NY, USA, 2005. ACM.

[Mau02] Ueli Maurer. Indistinguishability of random systems. In Lars Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. Springer Berlin Heidelberg, May 2002.

[Mau11] 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 *LNCS*, pages 33–56. Springer Berlin Heidelberg, April 2011.

[MR11] 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.

[MRH04] 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 — TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer Berlin Heidelberg, February 2004.

[MRT12] 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 *LNCS*, pages 209–229. IACR, Springer Berlin Heidelberg, 2012.

[RSS11] 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 *LNCS*, pages 487–506. Springer Berlin Heidelberg, 2011.

A The Storage-Auditing Scenario from [RSS11]

In [RSS11], 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 [RSS11] 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 [RSS11], 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 $f(r, c)$.[7]

Since the construction chop-MD was shown indifferentiable from a random oracle in [CDMP05], 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.

As we sketched briefly already in Section 1, 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 construction chop-MD using an ideal compression function from a random oracle.

First, let us recall 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, it could also obtain when interacting with the ideal one. This is since it could perform all the tasks embodied in the simulator on its 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 [CDMP05] 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.

B Fano's Inequality

In our proof we use the following theorem given by Fano in [Fan61].

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.

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

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. □