From Indifferentiability to Constructive Cryptography (and Back)

Ueli Maurer[1(✉)] and Renato Renner[2]

[1] Department of Computer Science, ETH Zurich, Zurich, Switzerland
maurer@inf.ethz.ch
[2] Department of Physics, ETH Zurich, Zurich, Switzerland
renner@phys.ethz.ch

Abstract. The concept of indifferentiability of systems, a generalized form of indistinguishability, was proposed in 2004 to provide a simplified and generalized explanation of impossibility results like the non-instantiability of random oracles by hash functions due to Canetti, Goldreich, and Halevi (STOC 1998). But indifferentiability is actually a constructive notion, leading to possibility results. For example, Coron *et al.* (Crypto 2005) argued that the soundness of the construction $C(f)$ of a hash function from a compression function f can be demonstrated by proving that $C(R)$ is indifferentiable from a random oracle if R is an ideal random compression function.

The purpose of this short paper is to describe how the indifferentiability notion was a precursor to the theory of constructive cryptography and thereby to provide a simplified and generalized treatment of indifferentiability as a special type of constructive statement.

1 Introduction

An important abstraction in cryptography, introduced by Bellare *et al.* [4], is the so-called random oracle model (ROM). A random oracle is an idealized resource or system available to all involved parties, with parameters m and n, which behaves as if it contained a uniformly chosen function table $F : \{0,1\}^m \to \{0,1\}^n$ and, for every query $x \in \{0,1\}^m$ from any party, provides the function value $F(x)$ to that party. Other parties do not see the query x nor the reply $F(x)$. A random oracle can also be defined for the countably infinite domain $\{0,1\}^*$ of all finite-length input strings, the resource usually meant in cryptography by the term "random oracle".

The idea behind the ROM is a natural decomposition idea often arising in cryptographic reasoning. On one hand one tries to construct, at least approximately, a random oracle from weaker resources (e.g. a shared random string), and on the other hand one uses the idealized resource of a random oracle to design secure protocols. The rationale is that if a well-designed hash function can be assumed to behave like a random oracle, then a cryptographic protocol proved secure in the ROM remains secure when the random oracle is replaced

M. Hirt and A. Smith (Eds.): TCC 2016-B, Part I, LNCS 9985, pp. 3–24, 2016.
DOI: 10.1007/978-3-662-53641-4_1

by a hash function, thus composing two steps of reasoning. Analogous reasoning is, for example, applied if one proves a scheme secure assuming it has access to a uniformly random value (e.g., a shared secret key), and then argues that the random value can be replaced by a pseudo-random value without compromising security.

Two questions arise.

1. What exactly do we mean by composition of steps in the above reasoning and how can we make it mathematically sound? It turns out, as discussed in this paper, that the random oracle example requires a different and more sophisticated reasoning compared to the pseudo-randomness example.
2. Can a random oracle be constructed from a weaker resource, especially one that can realistically be assumed to be available in a given application context?

An important paper by Canetti *et al.* [6] showed that the random oracle model is not instantiable by any hash function. The approach taken in that paper was to devise a provably secure signature scheme S, which internally makes use of a secure signature scheme S' and has access to a random oracle, such that S is insecure if the random oracle is replaced by any hash function, even one devised in the future and in full knowledge of the random oracle. Intuitively, the reason for this impossibility is that the program code p for a hash function can not contain more entropy than the length of p and that therefore, if one accesses the random oracle for a number of arguments yielding more entropy than the length of p, then one can distinguish a black-box containing the random oracle from one containing the hash function.

This result raises some natural questions which were the starting point for the research leading to the paper [18] on indifferentiability.

1. How can this simple entropy argument be made precise, in view of the quite involved original proof of [6], and how can it be generalized?
2. What is a meaningful definition of the possibility (rather than impossibility) of such a construction, and which concrete constructions are indeed possible?
3. How can the construction notion be generalized to capture other cryptographic settings like encryption or message authentication?
4. How can one design complex cryptographic protocols such that their security proof follows simply from composition and the (generally simple) security proofs of the individual construction steps?

The answer to the second question turned out to be useful for the design of hash functions from a compression function (e.g. see [1,2,7,11,12]).

The third question asks for an understanding of the application of a cryptographic scheme like a symmetric or public-key encryption scheme, a message authentication scheme, or a digital signature scheme, as a construction of a resource from other resources. The question then is which resources one should consider and how cryptographic schemes can be understood as such constructions. Cryptographic resources provide a guarantee to honest parties in view

of potentially dishonest parties behaving arbitrarily. Such arbitrary or unspecified behavior is often called "malicious". For example, a secure communication channel guarantees to the honest parties (the sender and the receiver) that an adversary can learn at most the length of the message. Note that, in the sense of a specification discussed later, it is not guaranteed that the adversary learns the message length, only that she does not learn more. For example, symmetric encryption can be understood as constructing a secure channel from an authenticated channel and a shared secret key, and message authentication can be understood as constructing an authenticated channel from an insecure channel and a shared secret key [16,17,19,20]. Similarly, public-key encryption can be understood as constructing a confidential channel from an insecure channel and an authenticated channel in the other direction [8].

The above approach to cryptography was proposed in [17], motivated by earlier approaches to achieving composition in cryptography, most notably Canetti's UC framework [5] and the reactive simulatability framework of Backes, Pfitzmann, and Waidner [3].

The outline of the paper is as follows. In Sect. 2, the general construction paradigm and composability is discussed. In Sect. 3, we introduce the type of resources relevant in cryptography. In Sect. 4, the cryptographic construction notion is introduced and a few simple construction statements are proved. In Sect. 5, a few impossibility results are proved which imply considerably strengthened versions of the impossibility of constructing a random oracle. In Sect. 6, the positive construction result of Coron *et al.* [9] is discussed in view of the new treatment appearing in this paper. In Sect. 7, it is mentioned that the construction notion of this paper directly leads to construction statements involving several parties, some of which are honest and some of which are dishonest. In Sect. 8, the relation of this paper to the original indifferentiability paper [18] is explained.

A Word About Terminology. The title of the original paper [17] proposing constructive cryptography was "Abstract cryptography". Two main aspects of that paper were (1) the proposal to use top-down abstraction in the spirit of algebra in cryptography (and more generally in computer science), and (2) to use the construction paradigm (see Sect. 2) in cryptography. Therefore, depending on which aspect is stressed, both "abstract cryptography" and "constructive cryptography" have been used in the literature to refer to this theory. The term constructive cryptography, which was first used in [16], seems more natural and captures the goal of the theory better, and we propose to use it from now on to avoid confusion.

2 The Construction Paradigm

2.1 Specifications and Constructions

In almost every engineering discipline one considers, explicitly or implicitly, the concept of a *specification* of an object or *resource*. Examples include the specifi-

cation of a mechanical part (e.g. by lower and upper bounds on its dimensions, its weight, and material parameters) and the specification of a software module M (e.g. by defining the functions that M computes and possibly some accuracy guarantees and/or some timing guarantees).

A key task in such a discipline is to *construct*, from an object or resource satisfying a certain specification $\mathcal{R}$, an object or resource satisfying another (better or more valuable) specification $\mathcal{S}$. Such a construction is achieved by means of a *constructor* or recipe, say γ. One can then write

$$\mathcal{R} \xrightarrow{\gamma} \mathcal{S}.$$

For example, the designer of a software module N making use of the module M will provide a specification $\mathcal{S}$ which is guaranteed (and proved) to be satisfied by N, provided the underlying module M satisfies specification $\mathcal{R}$.

As another example, in communication theory and information theory, a binary symmetric channel (BSC) is a well-known resource specification characterized by a maximal probability p of flipping the transmitted bits (where the errors for all bits are independent). A good error-correcting code with 2^k codewords of length n can be understood as constructing, from an n-bit BSC with parameter p, an error-free k-bit communication channel. More precisely, one only achieves a specification of a channel which is ϵ-close to an error-free k-bit channel, for a small ϵ and a certain measure of closeness, i.e., for a metric on the set of channels, namely the worst-case (over messages) decoding error probability.

Typically one considers a certain set Γ of constructors, possibly restricted in terms of efficiency or implementation cost. One is then interested in constructibility and also in non-constructibility statements, where $\mathcal{S}$ is not constructible from $\mathcal{R}$, denoted $\mathcal{R} \not\longrightarrow \mathcal{S}$, if there exists no constructor γ for which $\mathcal{R} \xrightarrow{\gamma} \mathcal{S}$:

$$\mathcal{R} \not\longrightarrow \mathcal{S} :\Longleftrightarrow \neg\exists\, \gamma \in \Gamma :\ \mathcal{R} \xrightarrow{\gamma} \mathcal{S}.$$

One often wants to use several resources in a construction, i.e., one wants to consider a tuple of resources, for example a tuple of three resources satisfying specifications $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$, as a single resource. We denote such a combined resource specification as $[\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3]$.

2.2 Composition

If we assume that constructors can be composed, where the constructor resulting from applying γ and then γ' is denoted as $\gamma' \circ \gamma$, then a very desirable and natural property is that the corresponding construction statements can be composed. Formally, this means that

$$\mathcal{R} \xrightarrow{\gamma} \mathcal{S} \wedge \mathcal{S} \xrightarrow{\gamma'} \mathcal{T} \Longrightarrow \mathcal{R} \xrightarrow{\gamma' \circ \gamma} \mathcal{T}.$$

For example, any construction requiring an error-free channel and resulting in a yet more useful resource should also be (approximately) correct if, instead of

the error-free channel, the channel constructed by an error-correcting code from an error-prone channel is used. Whether or not this is indeed the case requires a formalization and a proof.

Another useful property of the construction notion is *context-insensitivity*: For any $\mathcal{U}$ and $\mathcal{V}$,

$$\mathcal{R} \xrightarrow{\gamma} \mathcal{S} \Longrightarrow [\mathcal{U}_1, \ldots, \mathcal{U}_k, \mathcal{R}, \mathcal{V}_1, \ldots, \mathcal{V}_\ell] \xrightarrow{\gamma} [\mathcal{U}_1, \ldots, \mathcal{U}_k, \mathcal{S}, \mathcal{V}_1, \ldots, \mathcal{V}_\ell]$$

for any $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{U}_1, \ldots, \mathcal{U}_k, \mathcal{V}_1, \ldots, \mathcal{V}_\ell$. The understanding here is that γ "knows" which resource it needs to access.[1]

We point out that these properties may or may not be satisfied by a construction notion under consideration, and when investigating a concrete such notion one needs to prove that they are satisfied.

2.3 Sets as Specifications

The notion of a specification is abstract, but often a specification is understood as the subset of a universe Φ of objects, namely those that satisfy the specification. For example the specification of a BSC corresponds to the set of all channels where the bit-flipping probability of each bit is upper bounded by p but otherwise arbitrary (and the flipping events are independent). As another example, a software specification may require only an approximative computation of certain results, and a concrete element of the specification is given by a fixed function that is within the accuracy bounds.

If a pseudo-metric d on Φ is defined, a particular type of specification by sets are ϵ-balls around a given object R, denoted

$$R^\epsilon = \{R' \mid R' \approx_\epsilon R\},$$

where we write $R' \approx_\epsilon R$ for $d(R, R') \leq \epsilon$. More generally,

$$\mathcal{R}^\epsilon = \{R' \mid \exists R \in \mathcal{R} : \ R' \approx_\epsilon R\} = \bigcup_{R \in \mathcal{R}} R^\epsilon,$$

A construction statement $\mathcal{R} \xrightarrow{\gamma} \mathcal{S}$ becomes stronger the larger the specification $\mathcal{R}$ (i.e., the less needs to be assumed about the given resource), and, analogously, the statement becomes stronger the smaller the specification $\mathcal{S}$, i.e., the more specific the guarantee about the constructed resource is. In other words, we have

$$\mathcal{R} \xrightarrow{\gamma} \mathcal{S} \Longrightarrow \mathcal{R}' \xrightarrow{\gamma} \mathcal{S}'$$

if $\mathcal{R}' \subseteq \mathcal{R}$ and $\mathcal{S} \subseteq \mathcal{S}'$.

[1] Formally, the constructor γ on the right side might involve some scheme for addressing the resource specified by $\mathcal{R}$ among all resources, and in this case it would have to be an adequately modified version of γ on the left side (i.e., in $\mathcal{R} \xrightarrow{\gamma} \mathcal{S}$).

The situation is dual for impossibility results, which are a focus of [6,18] and of this paper. Namely,

$$\mathcal{R} \not\longrightarrow \mathcal{S} \Longrightarrow \mathcal{R}' \not\longrightarrow \mathcal{S}'$$

if $\mathcal{R} \subseteq \mathcal{R}'$ and $\mathcal{S}' \subseteq \mathcal{S}$. In other words, the smaller $\mathcal{R}$ or the larger $\mathcal{S}$, the stronger is the impossibility statement. We will pay attention to trying to obtain strong possibility and impossibility results.

3 Cryptographic Resource Systems and Their Use

In this section we discuss the specific type of resource appearing in cryptographic statements.

3.1 Systems, Interfaces, Parties

Cryptographic resources can be modelled as systems with several interfaces. One can think of each interface as allowing one party to connect to the system and access the functionality provided by it, but this view is not strict. It is also possible that interfaces capture a more fine-grained capability and that several interfaces are assigned to the same party. Conversely, one could also consider several parties as accessing (sub-interfaces of) the same interface.

In a cryptographic context, one considers so-called "honest" and "dishonest" parties, where often all the dishonest parties are modeled as a single party, called "the adversary" or Eve.

For the purpose of this paper, it suffices to consider resources with two interfaces, where all honest parties (sometimes summarized as Alice) access the resource through the left interface and Eve accesses it from the right side.

More technically, in this paper we consider a specific type of system, namely discrete resource systems that can (possibly) take an input at any interface and provide an output at the same interface. Then a system can take another input at some interface and produce an output at that interface, etc. For this paper, we will not need a formalization of such discrete systems, but we refer to [15,22]. The metric on the set of discrete systems is naturally defined via the optimal distinguishing advantage of a certain class of distinguishers.

3.2 Example Resource Systems

An example of such a resource is a uniform random function (URF) $\{0,1\}^m \to \{0,1\}^n$, accessible to all involved parties, which can be specified by considering a uniformly chosen function table $F : \{0,1\}^m \to \{0,1\}^n$ that can be accessed by giving as input a value x and receiving as output the value $F(x)$.

When considering the above URF resource in a cryptographic context, even when restricted to a single honest party and a single adversary, the above specification is not adequate as it is on one hand too specific (it guarantees that the adversary can access the resource, while one does not want to give such a

guarantee), and it is on the other hand not sufficiently specific in that one would want to additionally specify lower and upper bounds on the number of allowed queries (see later), as well as what is guaranteed to be hidden from the adversary. There are a number of such specifications which are natural, and we list a few of them below.

1. Alice can access the URF and Eve has no access to it.
2. Alice can access the URF and Eve has no access to it, but she potentially sees whenever Alice makes a query.
3. As before, but Eve can potentially also learn the values queried and obtained by Alice.
4. Alice and Eve can both access the URF and Eve obtains no other information (e.g. about Alice's access).
5. As before, but Eve can potentially also learn the values queried and obtained by Alice.

The fourth example is what is often called a (fixed input-length) random oracle which is accessible to all parties, whether honest or not, here restricted to a single honest party. One can also consider such a random oracle resource with arbitrary input-length, i.e., which for each input in $\{0,1\}^*$ returns a random value in $\{0,1\}^n$. An important question is from which resources a random oracle can or cannot be constructed. The impossibility result of [6] can be interpreted as the statement that a random oracle cannot be constructed from a fixed bit-string (the hash program) which can be probabilistically chosen.

3.3 Converters

A party can use a resource $R \in \Phi$ by applying to it a so-called *converter*[2] α which is, for example, a (state-full) protocol engine. A converter can be thought of as a system, with an inside and an outside interface, which is attached to the resource system. Application of a converter at interface i transforms a resource R into another resource which we denote by $\alpha^i R$, with the same set of interfaces as R.

More formally, we consider a set Σ of objects, called converters. A converter α, when applied as an interface i of a resource, induces a function[3] $\Phi \to \Phi : R \mapsto \alpha^i R$. Moreover, Σ is equipped with a composition operation $\circ$ satisfying

$$(\beta \circ \alpha)^i R = \beta^i(\alpha^i R).$$

The set Σ also contains a special element, the *identity converter* $id \in \Sigma$, which induces the identity function $\Phi \to \Phi$ (for any interface i) and simply stands for using the resource "as is". It satisfies

$$id \circ \alpha = \alpha \circ id = \alpha.$$

[2] The term "converter" is used because its application at an interface converts the interface into an interface with a different behavior.

[3] In general, one could consider partial function where the application of a converter at an interface need not always be defined. For the purpose of this paper there is no need to consider partial functions.

The set Σ is closed under composition, i.e., $\Sigma \circ \Sigma = \Sigma$, where equality holds because $id \in \Sigma$.

For two-interface resources as used in this paper, if one (i.e., Alice) applies a converter α at the left interface of a resource R, the resulting resource is denoted as

$$\alpha R.$$

Similarly, if one (i.e., Eve) applies a converter β at the right interface of a resource R, the resulting resource is denoted as

$$R\beta.$$

A key property we require, and which is typically satisfied, is that application of converters at the left and the right interface commute, i.e.,

$$(\alpha R)\beta = \alpha(R\beta),$$

which justifies to write $\alpha R\beta$ for the resulting resource.

A *resource specification* is simply a subset of $\mathcal{R} \subseteq \Phi$ containing those resources satisfying the specification. When no confusion can arise, we will also use the term resource for a resource specification. An element of $R \in \Phi$ can be understood as a singleton specification, i.e., as $\{R\}$.

Applying a converter α to a resource specification $\mathcal{R}$ is naturally defined as

$$\alpha\mathcal{R} = \big\{\alpha R \big|\, R \in \mathcal{R}\big\},$$

and analogously for $\mathcal{R}\beta$ and $\alpha\mathcal{R}\beta$.

3.4 Some Relevant Resource Specification Relaxations

The purpose of this section is to introduce a few generic types of relaxations of a resource specification $\mathcal{R}$ and to state some simple facts. We have already discussed ϵ-balls $\mathcal{R}^\epsilon$.

The understanding is that a dishonest party can do something arbitrary, i.e., apply an arbitrary converter. For a specification $\mathcal{R}$, the specification capturing that it is unknown what happens at the right interface is

$$\mathcal{R}^* \;:=\; \mathcal{R}\Sigma = \big\{R\beta \big|\, R \in \mathcal{R}, \beta \in \Sigma\big\},$$

where the symbol $*$ stands for an arbitrary converter. One can prove that

$$\mathcal{R} \subseteq \mathcal{R}^* = (\mathcal{R}^*)^*. \tag{1}$$

One can consider a special converter $\dashv$ which blocks the right interface, i.e., the resource $R\dashv$ only has a left interface. More technically speaking, for a resource $R\dashv$, a distinguisher sees only the left interface and has no access to the right interface. A resource R is *right-outbound* if no converter attached to the right interface can have an effect at the left interface, i.e., if

$$R^*\dashv = R\dashv.$$

This means that no signalling from the right to the left interface of R is possible. In this paper we do not need the dual left-outbound property.

For a given resource specification $\mathcal{R}$ one can consider the set, denoted $\mathcal{R}[\![$, of right-outbound resources S compatible with (a resource in) $\mathcal{R}$ (only) at the left interface:

$$\mathcal{R}[\![:= \left\{ S \middle| \, S \text{ is right-outbound and } \; S\dashv \in \mathcal{R}\dashv \right\} = \left\{ S \middle| \, S^*\dashv = S\dashv \in \mathcal{R}\dashv \right\}.$$

For example, if $\mathcal{R}$ denotes the specification of a random oracle (which hides Alice's queries from Eve), then $\mathcal{R}[\![$ includes all resources that leak partial or all information about Alice's queries to Eve. An impossibility result stating that $\mathcal{R}[\![$ is not constructible is therefore a significantly stronger statement than that a standard random oracle is not constructible. One can prove that

$$\mathcal{R} \subseteq \mathcal{R}[\![= (\mathcal{R}[\![)[\![. \tag{2}$$

3.5 Modeling Aspects: Resources vs. Converters

The implementation of a converter requires computational resources such as computing power, memory, and randomness. On one hand, how many resources an implementation requires seems relevant, and it appears generally better if a converter can be more efficiently implemented. On the other hand, one often makes statements that involve a quantification over all converters (e.g. all simulators), and such a quantification only makes sense if, by definition, the actual choice is irrelevant.[4]

In almost every scientific consideration, one intentionally ignores certain aspects as irrelevant and focuses on the particular ones considered relevant in the given context. What is relevant or irrelevant is generally a conscious choice. For example, in a computer science (or more specifically a cryptographic) context, one may or may not care to model the exact computational power available to a party. In particular, one may use an asymptotic model and only require that the number of computational steps is polynomially bounded in a security parameter.

The general guiding principle in constructive cryptography is that everything that is considered relevant for the analysis one wants to perform is modeled as part of the resource. In contrast, the choice of a converter is, by definition, irrelevant with regard to the entailed cost or complexity. If, for instance, computing power, memory, or randomness needed for a cryptographic construction is considered to matter, then it has to be explicitly modeled as part of the resource. To illustrate this point, we explain a few possible such explicit choices. Each can be thought of as a particular security model (e.g. computational or information-theoretic).

1. The term *information-theoretic security* is usually used when computation (at least by the adversary) is irrelevant. In such a case the converter set includes all systems, regardless of the computational complexity of implementing them.

[4] For a logical predicate P, the purpose of a statement of the form $\exists x \; P(x)$ is precisely to ignore *which* x makes $P(x)$ true.

2. Even for information-theoretic security one may be interested in making nevertheless the memory requirements explicit (see [10]). In this case, memory is modeled as part of the resource and the converters are all systems that can compute arbitrary functions (regardless of the complexity) but cannot keep state between invocations.[5] Ristenpart et al. [23] pointed out an apparent problem with the indifferentiability notion of [18], but it was shown in [10] that this problem was only an artefact of the fact that Turing machines come, by definition, with an arbitrary amount of memory (the tape) and that therefore this model is not adequate in a setting, as that considered in [23], where memory is indeed a relevant resource.
3. If computing power is considered relevant, then one can consider converters that perform no computation by themselves but only connect systems and possibly input constants (for example a program). Any computational resource can be modeled as a (parallel) resource. Such a resource can either be a specific system with a certain behavior (e.g. a system encrypting messages), without reference to an implementation on a certain computational model. Alternatively, it could be a computer resource C in some computational model, with an upper bound on the available computing power (for example called complexity), and which can run an arbitrary program up to that complexity bound. In this case, the converter inputs a program to C, and we consider it irrelevant (from a resource viewpoint) which program is used. Possibly, the specification of C could involve an upper bound on the length of the program. In such a view, converters only route information, without performing computation.
4. If, for some notion of efficiency, efficient computing power is considered irrelevant, then one can consider Σ to be the set of efficiently implementable converter systems. Typically in cryptography, efficient is defined as some form of polynomial-time notion, which of course, and unfortunately, requires now the objects to be defined asymptotically in some way. A main reason for using polynomial-time is that this notion, if properly defined, is closed.[6] We point out that polynomial-time is a specific choice that has its merits but for many statements need not be fixed.

Clearly, one could consider different converter sets for honest parties and for dishonest parties. For example, it would be natural to consider a notion of efficiency and a different, larger notion of feasibility, where the converters of honest parties must be efficiently implementable and the converters of dishonest parties must only be feasibly implementable. It does not really seem well-justified to use the same polynomial-time notion for both, except by tradition and possibly by the set of results one can prove for this choice.

[5] In this model, the memory required for a function computation is assumed to be free. Of course, one could also model this memory as a resource.

[6] More formally, converters α and β from this particular set Σ can be composed to a new converter, say $\alpha \circ \beta$, and this composition is closed in the sense that the function $\Phi \to \Phi$ induced by $\alpha \circ \beta$ is contained in the class of functions induced by converters in Σ.

4 Cryptographic Constructions for a Fixed Adversary Interface

4.1 Definition of Constructions and Some Lemmas

If a resource satisfying specification $\mathcal{R}$ is available, Alice can apply a converter π to it, resulting in specification $\pi\mathcal{R}$. Often one wants to think about $\pi\mathcal{R}$ in a simpler way, namely in terms of a specification $\mathcal{S}$ such that $\pi\mathcal{R} \subseteq \mathcal{S}$. The guarantee given to Alice by the specification $\mathcal{S}$ is generally weaker than the specification $\pi\mathcal{R}$, but, in the usual sense of abstraction, this loss of information is accepted because $\mathcal{S}$ is a simpler (to use and work with) specification.

We can then say that a desired resource (specification) $\mathcal{S}$ *is constructed from* an assumed resource (specification) $\mathcal{R}$ by application of the converter $\pi \in \Sigma$ (which is the constructor). This is written as $\mathcal{R} \xrightarrow{\pi} \mathcal{S}$.

Definition 1. $\mathcal{R} \xrightarrow{\pi} \mathcal{S} :\Longleftrightarrow \pi\mathcal{R} \subseteq \mathcal{S}$.

Lemma 1. *This construction notion is composable:*

$$\mathcal{R} \xrightarrow{\pi} \mathcal{S} \wedge \mathcal{S} \xrightarrow{\pi'} \mathcal{T} \Longrightarrow \mathcal{R} \xrightarrow{\pi' \circ \pi} \mathcal{T}.$$

Proof. From the first condition $\pi\mathcal{R} \subseteq \mathcal{S}$ it follows that $\pi'\pi\mathcal{R} \subseteq \pi'\mathcal{S}$. Combining this with the second condition, $\pi'\mathcal{S} \subseteq \mathcal{T}$, we obtain $\pi'\pi\mathcal{R} \subseteq \mathcal{T}$, which was to be proved. □

The following lemmas assert that the three specification relaxations discussed in Sect. 3.4 are compatible with the construction notion.

Definition 2. *A metric d on Φ is called* non-expanding *if $d(\alpha R, \alpha S) \leq d(R, S)$ for all α and $d(R\beta, S\beta) \leq d(R, S)$ for all β.*

Lemma 2. *If the metric on Φ is non-expanding, then, for any $\epsilon > 0$,*

$$\mathcal{R} \xrightarrow{\pi} \mathcal{S} \Longrightarrow \mathcal{R}^{\epsilon} \xrightarrow{\pi} \mathcal{S}^{\epsilon}.$$

Proof. We need to show that if $R' \in \mathcal{R}^{\epsilon}$, i.e., $R' \approx_{\epsilon} R$ for some $R \in \mathcal{R}$, then $\pi R' \in \mathcal{S}^{\epsilon}$, i.e., $\pi R' \approx_{\epsilon} S$ for some $S \in \mathcal{S}$. The condition $\mathcal{R} \xrightarrow{\pi} \mathcal{S}$ guarantees that $\pi R = S$ for some $S \in \mathcal{S}$. For the same S we have $\pi R' \approx_{\epsilon} S$ since $\pi R' \approx_{\epsilon} \pi R = S$ (due to the non-expanding property). This completes the proof. □

The following lemmas are stated without proofs.

Lemma 3. $\mathcal{R} \xrightarrow{\pi} \mathcal{S} \quad \Longrightarrow \quad \mathcal{R}^{*} \xrightarrow{\pi} \mathcal{S}^{*}$.

Lemma 4. $\mathcal{R} \xrightarrow{\pi} \mathcal{S} \quad \Longrightarrow \quad \mathcal{R}[\![\xrightarrow{\pi} \mathcal{S}[\![$.

4.2 Proving Constructions by Simulators

A line of reasoning often arising in cryptography, including [18], can be captured by the following system equation (see also [17]):

$$\pi R \;\approx_\epsilon\; S\sigma, \tag{3}$$

where the converter σ is usually called a *simulator* (see discussion in Sect. 4.2). The usefulness of finding a simulator σ satisfying the equation is that it implies a construction statement:

Lemma 5. *If the metric is non-expanding, then*

$$\exists \sigma \in \Sigma : \; \pi R \approx_\epsilon S\sigma \Longrightarrow R \xrightarrow{\pi} (S^*)^\epsilon.$$

Proof. Since $\sigma \in \Sigma$ we have $S\sigma \in S\Sigma = S^*$. Hence $\pi R \approx_\epsilon S\sigma$ implies that $\pi R \subseteq (S\sigma)^\epsilon \subseteq (S^*)^\epsilon$, which is the definition of $R \xrightarrow{\pi} (S^*)^\epsilon$. □

In the literature, the converter σ in Eq. (3) is usually called a simulator. It is sometimes described as translating what an adversary could do in the real world (the left side of the equation), say β, into what she needs to do in the ideal world (the right side of the equation) to achieve the same (or something close to) what she would achieve in the real world, namely $\beta \circ \sigma$. Note that $\pi R\beta \approx_\epsilon S\sigma\beta$ due to the non-expanding property of the pseudo-metric.

We point out, however, that in contrast to most of the existing literature, the actual statement of interest (see Lemma 5) to us is not Eq. (3) itself, but the construction statement it implies. In particular, the simulator does not appear in the definition of a construction, and there can be interesting construction statements proved in different ways than by use of Lemma 5.

In view of Lemma 5, the notion of indifferentiability [18] can be understood as follows: T is indifferentiable from S, within ϵ, if $T \subseteq (S^*)^\epsilon$, where this is proved by demonstrating a simulator σ such that $T \approx_\epsilon S\sigma$. If $T = \pi R$, this corresponds to the construction statement $R \xrightarrow{\pi} (S^*)^\epsilon$.

4.3 Computational Considerations

Often in cryptography, Σ is the set of polynomial-time implementable converter systems. If the metric on Φ is chosen as the two-valued computational indistinguishability metric, then a polynomial-time converter can be absorbed into a poly-time distinguisher without leaving the distinguisher class, i.e., the metric is non-expanding.

In a concrete-security consideration, the efficiency loss of a reduction and therefore the concrete implementation complexity of σ matters. In other words, a statement of the form (3) becomes more useful for a more efficient σ. This, however, does at first not seem to be compatible with the idea that converters in Σ are considered free (of cost). Either a converter is free, or it is not. Let us explain how this contradiction is resolved in our approach.

More specifically, suppose we use model 3 described in Sect. 3.5, where Σ are the converters that perform no computation. Suppose furthermore that one has shown that equality $\pi R = S\beta$ holds for some system β that requires some computation, i.e., $\beta \notin \Sigma$. Then we can give the equation the following meaning. Let $\bar{\beta}$ be a system corresponding to the resource that behaves like β, with inside and outside interface both available to Eve (only at the right interface). Then one can rephrase the equation $\pi R = S\beta$ as

$$\pi R = [S, \bar{\beta}]\ \sigma,$$

where σ is the trivial converter that simply connects $\bar{\beta}$ to S, i.e., such that

$$[S, \bar{\beta}]\ \sigma = S\beta.$$

In other words, any equation of the type $\pi R = S\beta$ can be turned into a construction statement of the form

$$R \xrightarrow{\pi} \left([S, \bar{\beta}]\right)^*$$

which makes the computational resource required for the "simulation" explicit.

5 Impossibility of Constructing a Random Oracle

As an example for an impossibility result, we show that a random oracle cannot be constructed, even if a source of public randomness is available. To state this more precisely, we use the following specifications.

- PR^k is *public randomness* of size k. The resource chooses Z uniformly at random from the set $\{0,1\}^k$ of k-bit strings.[7] Any party can read Z.[8]
- $\mathrm{RO}^{m \to n}_{[q,q']}$ is a *random oracle* with input size m and output size n. The resource chooses F uniformly at random from the set of all functions from $\{0,1\}^m$ to $\{0,1\}^n$. Any party can submit queries $x \in \{0,1\}^m$ which are answered by $F(x)$. At least q and at most q' queries by any party are allowed.

As before, we assume that the set of resources is equipped with a (non-expanding) distance measure, d, defined as the maximum advantage of any distinguisher from a class $\mathcal{D}$.[9] The results derived below will be valid for any reasonable distinguisher class $\mathcal{D}$. The only requirement is that the execution of basic algorithms giving inputs and receiving outputs and performing equality checks, such as D_1 and D_2 below, are within the class $\mathcal{D}$.

We start with a basic impossibility result, which asserts that public randomness cannot be expanded.

[7] To keep the presentation simple we assume that the probability distribution of Z is uniform; a generalization to arbitrary probability distributions is straightforward. This includes the case where PR^k is a fixed hash function program of length k.

[8] One may impose the additional restriction that the string Z can only be read bit-wise, but this is not relevant for the considerations here.

[9] That is, $d(R,S) = \sup_{D \in \mathcal{D}} \Delta^D(R,S)$, where $\Delta^D(R,S)$ is the absolute value of the difference between the probability that D returns 0 when connected to R and the probability that it returns 0 when connected to S.

Lemma 6. *Let $k \in \mathbb{N}$ and $\epsilon < \frac{1}{4}$. Then*

$$\mathrm{PR}^k \not\rightarrow \mathrm{PR}^{k+1} [\![\, ^{\epsilon}.$$

Proof. As explained, we regard PR^k as a specification of a system with two interfaces (left and right), which model the access to the resource by the honest and the dishonest parties, respectively. It suffices to consider two honest parties, which we label by A and A', as well as one dishonest party, labelled by E. We recall that in this two-interface case, any constructor corresponds to a converter π for the left interface, which can be understood as a pair of converters π_A and $\pi_{A'}$ for the two honest parties.

We need to prove that

$$d(\pi \mathrm{PR}^k, \mathcal{R}) \geq \frac{1}{4}$$

for any converter π and for any right-outbound resource $\mathcal{R}$ with the property $\mathcal{R} \dashv \subseteq \mathrm{PR}^{k+1} \dashv$. Because d is non-expanding, it suffices to show that

$$d(\pi \mathrm{PR}^k \pi', \mathcal{R}\pi') \geq \frac{1}{4} \tag{4}$$

for some converter π'. We take π' to be $\pi_{A'}$. More precisely, π' answers a query by E in the same way as π would answer a query by A'. We then consider a distinguisher D_1 that executes the following simple algorithm and show that it can tell apart $\pi \mathrm{PR}^k \pi'$ and $\mathcal{R}\pi'$ with advantage at least $\frac{1}{4}$.

Distinguisher D_1

read the $(k+1)$-bit strings Z_A and $Z_{A'}$ from the left interface;
read the $(k+1)$-bit string Z_E from the right interface;
if $Z_A \neq Z_{A'}$ **then**
  return 0; **halt** ;
else if $Z_A \neq Z_E$ **then**
  return 1; **halt** ;
return 0

Suppose first that D_1 is connected to $\pi \mathrm{PR}^k \pi'$. It only returns 1 if $Z_A = Z_{A'} \neq Z_E$. By the definition of π', the strings $Z_{A'}$ and Z_E are generated by identical (possibly probabilistic, but independent) procedures. It follows that the probability of the event $Z_A = Z_{A'} \neq Z_E$ is upper bounded by

$$\Pr[Z_A = Z_{A'}] \Pr[Z_E \neq Z_{A'}] = \Pr[Z_A = Z_{A'}](1 - \Pr[Z_A = Z_{A'}]) \leq \frac{1}{4}$$

(since $\frac{1}{4}$ is the maximum of the function $x \mapsto x(1-x)$ for $0 \leq x \leq 1$). Hence D_1 returns 0 with probability at least $\frac{3}{4}$.

Conversely, in the case where D_1 is connected to $\mathcal{R}\pi'$, $Z_A = Z_{A'}$ holds by definition of $\mathcal{R}$, and Z_A is a uniformly random $(k+1)$-bit string, whereas

Z_E is a $(k+1)$-bit string computed by π'. Since π' behaves by definition like $\pi_{A'}$ and thus takes as input only a k-bit string, Z_E depends on a string W of length at most k. D_1 only returns 0 if $Z_A = Z_E$. The probability of this event is upper bounded by the min-entropy of Z_A conditioned on W, i.e., $\Pr[Z_A = Z_E] \leq 2^{-H_{\min}(Z_A|W)}$ (cf. Appendix). By (11), the chain rule for the min-entropy, we have $H_{\min}(Z_A|W) \geq H_{\min}(Z_A) - k = 1$, where we used that W consists of at most k bits. We conclude that $\Pr[Z_A = Z_E] \leq \frac{1}{2}$. Hence, when connected to $\mathcal{R}\pi'$, D_1 returns 0 with probability at most $\frac{1}{2}$. Combining this with the above shows that the distinguishing advantage is at least $\frac{1}{4}$, which implies (4). □

Lemma 6 states that public randomness cannot be expanded by a single bit, even if one would tolerate that Eve may learn something about what happens at the honest parties' interface (which is captured by "[["). This also suggests that one cannot construct a more powerful public randomness resource that allows to extract more than k bits:

Corollary 1. *Let $k \in \mathbb{N}$ and $\epsilon < \frac{1}{4}$. Then*

$$\mathrm{PR}^k \not\longrightarrow \mathrm{RO}^{m\to 1}_{[q,\infty]}[\![^{\epsilon}$$

unless $m < \log_2(k+1)$ or $q \leq k$.

Proof. Suppose that

$$\mathrm{PR}^k \xrightarrow{\pi} \mathrm{RO}^{m\to 1}_{[q,\infty]}[\![^{\epsilon} \tag{5}$$

holds for some constructor π. Let furthermore π' be a constructor that simply outputs the first $\min(q, 2^m)$ entries of the function table of the random oracle, and thus achieves

$$\mathrm{RO}^{m\to 1}_{[q,\infty]} \xrightarrow{\pi'} \mathrm{PR}^{\min(q,2^m)}[\![.$$

Using Lemma 4 as well as (2), this yields

$$\mathrm{RO}^{m\to 1}_{[q,\infty]}[\![\xrightarrow{\pi'} \mathrm{PR}^{\min(q,2^m)}[\![$$

and hence, using Lemma 2, also

$$\mathrm{RO}^{m\to 1}_{[q,\infty]}[\![^{\epsilon} \xrightarrow{\pi'} \mathrm{PR}^{\min(q,2^m)}[\![^{\epsilon}. \tag{6}$$

By Lemma 1, the composition of constructions (5) and (6) gives

$$\mathrm{PR}^k \xrightarrow{\pi'\circ\pi} \mathrm{PR}^{\min(q,2^m)}[\![^{\epsilon}.$$

Lemma 6 now implies that $\min(q, 2^m) < k+1$. □

We now proceed to a substantially stronger impossibility claim. Note that Corollary 1 only applies to cases where the total entropy that the honest parties can draw from the random oracle is strictly larger than the number k of public random bits that are available. Theorem 1 below shows that this is not necessary

for the impossibility result to hold. It asserts that even a weak random oracle that answers only a small number of queries (say, $q = 1024$), and thus only provides a small amount of entropy to the honest parties, cannot be constructed. In addition, the impossibility claim remains valid if one tolerates that the constructed random oracle leaks arbitrary information, e.g., about what happens at the honest parties' interface, to the adversary.

For simplicity, we restrict the statement to oracles with output size 1 (but it obviously implies a corresponding impossibility result for random oracles with larger output size).

Theorem 1. *For any* $k, m, q \in \mathbb{N}$ *and* $\epsilon \leq \frac{1}{2}$

$$\mathrm{PR}^k \not\rightarrow \mathrm{RO}^{m\to 1}_{[q,\infty]} [\![\, ^{\epsilon}$$

unless $m < \min(1 + \log_2 k, 10)$ *or* $q < 2^{10}$.

Proof. Set without loss of generality $q = 2^{10}$ and assume that $m \geq 1 + \log_2 k$ and $m \geq 10$. The proof proceeds analogously to that of Lemma 6, i.e., we show that

$$d(\pi \mathrm{PR}^k \pi', \mathcal{R}\pi') > \frac{1}{2}, \tag{7}$$

where $\mathcal{R}$ is a right-outbound resource such that $\mathcal{R} \dashv = \mathrm{RO}^{m\to 1}_{[q,\infty]} \dashv$, and where π' is again a converter that reproduces the behavior of π for one party. To establish this inequality we consider a distinguisher D_2 defined by the following simple algorithm and show that it can tell apart $\pi \mathrm{PR}^k \pi'$ and $\mathcal{R}\pi'$ with advantage strictly larger than $\frac{1}{2}$.

Distinguisher D_2

choose q different values $X_1, \ldots, X_q$ at random from the set $\{0,1\}^m$;
for $j \in \{1, \ldots, q\}$ **do**
  A and A' submit query X_j and record the answers $Z_{A,j}$ and $Z_{A',j}$;
  if $Z_{A,j} \neq Z_{A',j}$ **then return** 0; **halt** ;
end
for $j \in \{1, \ldots, q\}$ **do**
  E submits query X_j and records the answer $Z_{E,j}$;
  if $Z_{A,j} \neq Z_{E,j}$ **then return** 1; **halt** ;
end
return 0

We first treat the case where D_2 is connected to $\pi \mathrm{PR}^k \pi'$. D_2 only returns 1 if, for some $j \in \{1, \ldots, q\}$, $Z_{A,j} \neq Z_{E,j}$. Following the same reasoning as in the proof of Lemma 6, we can infer that the probability of this event is upper bounded by $\frac{1}{4}$. Hence, when connected to $\pi \mathrm{PR}^k \pi'$, D_2 returns 0 with probability at least $\frac{3}{4}$.

Conversely, if D_2 is connected to $\mathcal{R}\pi'$, the answers $Z_{A,j}$ and $Z_{A',j}$ received by the honest parties upon any query X_j will agree by definition of $\mathcal{R}$. The distinguisher thus returns 0 only if they also coincide with the answers $Z_{E,j}$ received by a dishonest party E. This latter event only occurs if the tuple of answers $Z = (Z_{A,1}, \ldots, Z_{A,q})$ to all queries $X_1, \ldots, X_q$ is reproduced by the output of the converter π'. Since π' carries out the same computation as π for one party, this output depends on a string W of length at most k. Because Z can be regarded as a subset of q bits chosen at random from $2^m \geq 2k$ uniform bits, Corollary 2 (see Appendix) asserts that $H_{\min}(Z|X_1 \cdots X_q W) > 2$. This implies that the success probability of any strategy for guessing Z from W is strictly smaller than $\frac{1}{4}$. Hence, if connected to $\mathcal{R}\pi'$, D_2 returns 0 with probability strictly smaller than $\frac{1}{4}$. Combining this with the above shows that D_2 has distinguishing advantage strictly larger than $\frac{1}{2}$, which establishes (7). □

6 Construction Results

Coron *et al.* [9] showed that a random oracle with arbitrary input length and fixed output length n can be constructed from a compression function with fixed input length and output length n. The latter is itself modelled as a random oracle. The following theorem is a variation of this result.[10]

Theorem 2. *For any $n, \kappa, \ell, q, q' \in \mathbb{N}$ and $\epsilon = 2^{-n+1}q'^2$ there is π such that*

$$\mathrm{RO}^{n+\kappa+\lceil \log_2 \ell \rceil \to n}_{[\ell q, q']} \xrightarrow{\pi} \left((\mathrm{RO}^{n+\ell\kappa \to n}_{[q,q']})^*\right)^{\ell\epsilon}. \tag{8}$$

We are going to provide a proof of Theorem 2 based on the following result.

Lemma 7. *For any $n, a, q, q' \in \mathbb{N}$ and $\epsilon = 2^{-n+1}q'^2$*

$$\left[\mathrm{RO}^{a\to n}_{[q,\infty]},\ \mathrm{RO}^{n+\kappa\to n}_{[q,q']}\right] \xrightarrow{\pi} \left((\mathrm{RO}^{a+\kappa\to n}_{[q,q']})^*\right)^{\epsilon},$$

where π is the constructor which answers queries $(x, y) \in \{0,1\}^a \times \{0,1\}^\kappa$ with $F^{n+\kappa\to n}(F^{a\to n}(x), y)$, where $F^{a\to n}$ and $F^{n+\kappa\to n}$ are the functions defined by the two random oracles.

Proof. As shown in [9]

$$d\big(\pi\left[\mathrm{RO}^{a\to n}_{[q,\infty]}, \mathrm{RO}^{n+\kappa\to n}_{[q,q']}\right], \mathrm{RO}^{a+\kappa\to n}_{[q,q']}\sigma\big) \leq \epsilon$$

holds for a simulator σ defined by the following algorithm.

[10] The result in [9] corresponding to Theorem 2 is weaker in that the error ϵ is multiplied with ℓ^2 rather than ℓ.

Simulator σ

if *query* $x \in \{0,1\}^a$ *to* $F^{a\to n}$ **then**
  return *random* $v \in \{0,1\}^n$;
else if *query* $(v', y) \in \{0,1\}^n \times \{0,1\}^\kappa$ *to* $F^{n+\kappa\to n}$ **then**
  if v' *equals output of* $F^{a\to n}$ *for some previously queried* x' **then**
    return *answer of the resource to query* (x', y)
  else
    return *random* $z \in \{0,1\}^n$

The claim of the lemma then follows from Lemma 5. □

Proof. (of Theorem 2). The construction that gives rise to (8) can be regarded as the concatenation of several more basic constructions. The first, π_0, a simple domain splitting step, constructs ℓ independent random oracles with identical domain from a single random oracle, whose input domain consists of $\lceil \log_2 \ell \rceil$ additional bits, i.e.,

$$\mathrm{RO}_{[\ell q,q']}^{n+\kappa+\lceil \log_2 \ell \rceil \to n} \xrightarrow{\pi_0} \big[\underbrace{\mathrm{RO}_{[q,q']}^{n+\kappa\to n}, \cdots, \mathrm{RO}_{[q,q']}^{n+\kappa\to n}}_{\ell \text{ times}}\big]^*. \tag{9}$$

This is achieved by converters which simply answer any query x to the jth constructed random oracle by submitting the concatenation of x and a binary encoding of j to the given random oracle and then forwarding its answer.

For the next step, we invoke Lemma 7 with $a = n + j\kappa$, for $j \in \{1, \ldots, \ell - 1\}$. This lemma, together with Lemmas 2 and 3, the fact that $(\mathcal{R}^\epsilon)^* \subseteq (\mathcal{R}^*)^\epsilon$, and (1), implies that there exists a constructor π_j such that

$$\big[\big((\mathrm{RO}_{[q,\infty]}^{n+j\kappa\to n})^*\big)^{(j-1)\epsilon}, \mathrm{RO}_{[q,q']}^{n+\kappa\to n}\big] \xrightarrow{\pi_j} \big((\mathrm{RO}_{[q,q']}^{n+(j+1)\kappa\to n})^*\big)^{j\epsilon}.$$

Recursive application of this construction gives

$$\big[(\mathrm{RO}_{[q,\infty]}^{n+\kappa\to n})^*, \underbrace{\mathrm{RO}_{[q,q']}^{n+\kappa\to n}, \cdots, \mathrm{RO}_{[q,q']}^{n+\kappa\to n}}_{\ell-1 \text{ times}}\big] \xrightarrow{\pi_{\ell-1}\circ\cdots\circ\pi_1} \big((\mathrm{RO}_{[q,q']}^{n+\ell\kappa\to n})^*\big)^{(\ell-1)\epsilon}.$$

Using $\mathrm{RO}_{[q,q']}^{n+\kappa\to n} \subseteq \mathrm{RO}_{[q,\infty]}^{n+\kappa\to n} \subseteq (\mathrm{RO}_{[q,\infty]}^{n+\kappa\to n})^*$ we can substitute the first term in the above construction statement to obtain

$$\big[\underbrace{\mathrm{RO}_{[q,q']}^{n+\kappa\to n}, \cdots, \mathrm{RO}_{[q,q']}^{n+\kappa\to n}}_{\ell \text{ times}}\big] \xrightarrow{\pi_{\ell-1}\circ\cdots\circ\pi_1} \big((\mathrm{RO}_{[q,q']}^{n+\ell\kappa\to n})^*\big)^{(\ell-1)\epsilon}.$$

Similarly to the above, this implies that

$$\big[\underbrace{\mathrm{RO}_{[q,q']}^{n+\kappa\to n}, \cdots, \mathrm{RO}_{[q,q']}^{n+\kappa\to n}}_{\ell \text{ times}}\big]^* \xrightarrow{\pi_{\ell-1}\circ\cdots\circ\pi_1} \big((\mathrm{RO}_{[q,q']}^{n+\ell\kappa\to n})^*\big)^{(\ell-1)\epsilon}. \tag{10}$$

Theorem 2 now follows by composing the constructions (9) and (10). □

7 Generalization to Many Parties

We briefly sketch how the construction notion described in Sect. 4 directly leads to a construction notion for resources with several honest parties and an adversary, simply by considering the left interface as consisting of a sub-interface for each honest party and by considering the special type of converter (for the combined interface) as corresponding to a list of converters, one for each sub-interface. A typical case is the so-called Alice-Bob-Eve setting as discussed in [16,17] with two honest parties Alice and Bob. This model allows to capture many core cryptographic constructions, including the construction of a shared secret key, of an authenticated channel, and of a secure channel.

One can also capture a setting where various parties could be dishonest. Usually the terminology used is that a central adversary corrupts some of the parties. In other words, any party can possibly be honest or dishonest. A *protocol* is a tuple of converters, one for each potentially honest party, where the idea is that an honest party is guaranteed to apply the designated converter (i.e., to "follow the protocol"). One can then make a collection of construction statements, for each set of dishonest parties that needs to be considered, where for each such statements the honest parties' interfaces can be thought of as being grouped at the left side and the dishonest parties' interfaces are grouped at the right side.

8 Conclusions

The goal of this paper was to cover the essential aspects of the original indifferentiability paper [18], but in a more general and more adequate manner, leading to a general construction notion. The paper [18] contained basic ideas of constructive cryptography [17], but this is perhaps not apparent since [18] was mostly written in the tradition of the cryptography literature at the time: The objects considered were usually asymptotic in a security parameter, and the usual polynomial-time efficiency notion and the usual negligibility notion were used. It should be clear from [17] and this paper that fixing such a particular model is unnecessary. Moreover, indifferentiability was presented in [18] as a generalized form of indistinguishability, appearing as an intermediate step needed to define constructions (actually called reductions in [18]).

In view of the general construction notion presented in this paper, the indifferentiability notion corresponds to a specific construction type, for the special type S^* of resource specifications, where, moreover, S is right-outbound. Then T is indifferentiable from S, within ϵ, if $T \subseteq (S^*)^\epsilon$, where this is proved by demonstrating a simulator σ (not called simulator in [18]) such that $T \approx_\epsilon S\sigma$. If $T = \pi R$, this corresponds to the construction statement $R \xrightarrow{\pi} (S^*)^\epsilon$. Demonstrating a simulator and applying Lemma 5 is only one of possibly several ways of proving construction statements, and simulators should therefore probably only appear in proofs, not in definitions.

Acknowledgments. We would like to thank the TCC Test-of-Time award committee for selecting our paper for the award of this instantiation of TCC. Very sadly, our

coauthor Clemens Holenstein passed away in 2012 and could neither receive the award nor contribute to this paper. Discussions with many people have contributed immensely to shaping our described viewpoint of cryptography. Of particular help were discussions with Joël Alwen, Christian Badertscher, Ran Canetti, Sandro Coretti, Grégory Demay, Yevgeniy Dodis, Peter Gaži, Martin Hirt, Dennis Hofheinz, Daniel Jost, Christian Matt, Christopher Portmann, Phil Rogaway, Gregor Seiler, Björn Tackmann, Stefano Tessaro, Daniel Tschudi, Daniele Venturi, Stefan Wolf, and Vassilis Zikas.

Appendix: Min-entropy sampling

The min-entropy of a random variable X conditioned on another random variable Y, $H_{\min}(X|Y)$, is defined as (see, e.g., [14])

$$H_{\min}(X|Y) = -\log_2 \max_f \Pr[X = f(Y)],$$

where the maximum ranges over all functions f from the alphabet $\mathcal{Y}$ of Y to the alphabet $\mathcal{X}$ of X. Note that the expression in the logarithm on the right hand side can be interpreted as the maximum probability of correctly guessing X from Y. The min-entropy has several natural properties analogous to the Shannon entropy. Among them is a chain rule, which implies

$$H_{\min}(X|Y) \geq H_{\min}(X) - \log_2 |\mathcal{Y}|. \tag{11}$$

The min-entropy of a sample chosen at random from a min-entropy source has been studied in [13,21,24]. Roughly speaking, one can show that the min-entropy of the sample is proportional to the sample size and the min-entropy of the source. We use a version of this statement due to Wullschleger, which provides explicit bounds [25].[11]

Proposition 1. *Let $X \in \{0,1\}^n$ and Z be random variables and let T be a uniformly chosen subset of $\{1,\ldots,n\}$ of size $|T|$. Then*

$$\frac{H_{\min}(X_T|TZ)}{|T|} \geq f\left(\frac{H_{\min}(X|Z)}{n}\right) - \frac{5}{|T|},$$

where $f : [0,1] \to [0,1]$ is a monotonically strictly increasing function such that $f(1/2) > 1/144$.

Corollary 2. *Let $X \in \{0,1\}^n$ be uniformly distributed, let $Z \in \{0,1\}^k$ be an arbitrary random variable on $k \leq n/2$ bits, and let T be a uniformly chosen subset of $\{1,\ldots,n\}$ of size $|T|$. Then*

$$H_{\min}(X_T|TZ) > \frac{|T|}{144} - 5.$$

Proof. It follows from the chain rule (11) that conditioning on k bits cannot decrease the min-entropy by more than k bits, i.e.,

$$H_{\min}(X|Z) \geq H_{\min}(X) - k = n - k \geq n/2.$$

The claim then follows from Proposition 1. □

[11] Proposition 1 is a corollary of Theorem 1 of [25].

References

1. Andreeva, E., Mennink, B., Preneel, B.: On the indifferentiability of the Grøstl hash function. In: Garay, J.A., Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 88–105. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15317-4_7
2. Bertoni, G., Daemen, J., Peeters, M., Assche, G.: On the indifferentiability of the sponge construction. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78967-3_11
3. Backes, M., Pfitzmann, B., Waidner, M.: A general composition theorem for secure reactive systems. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 336–354. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24638-1_19
4. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: ACM Conference on Computer and Communications Security, pp. 62–73 (1993)
5. Canetti, R., Universally composable security: a new paradigm for cryptographic protocols. In: Proceedings of the 42nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2001, pp. 136–145. IEEE Computer Society Press, October 2001. Full version, http://eprint.iacr.org/2000/067
6. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, STOC 1998, pp. 209–218. ACM (1998)
7. Chang, D., Nandi, M.: Improved indifferentiability security analysis of chopMD hash function. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 429–443. Springer, Heidelberg (2008). doi:10.1007/978-3-540-71039-4_27
8. Coretti, S., Maurer, U., Tackmann, B.: Constructing confidential channels from authenticated channels—public-key encryption revisited. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 134–153. Springer, Heidelberg (2013). doi:10.1007/978-3-642-42033-7_8
9. Coron, J.-S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård revisited: how to construct a hash function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (2005). doi:10.1007/11535218_26
10. Demay, G., Gaži, P., Hirt, M., Maurer, U.: Resource-restricted indifferentiability. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 664–683. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38348-9_39
11. Dodis, Y., Reyzin, L., Rivest, R.L., Shen, E.: Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to MD6. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 104–121. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03317-9_7
12. Dodis, Y., Ristenpart, T., Steinberger, J., Tessaro, S.: To hash or not to hash again? (In)Differentiability results for H^2 and HMAC. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 348–366. Springer, Heidelberg (2012)
13. König, R., Renner, R.: Sampling of min-entropy relative to quantum knowledge. IEEE Trans. Inf. Theor. **57**, 4760–4787 (2011)
14. König, R., Renner, R., Schaffner, C.: The operational meaning of min- and max-entropy. IEEE Trans. Inf. Theor. **55**, 4337–4347 (2009)
15. Maurer, U.: Indistinguishability of random systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002). doi:10.1007/3-540-46035-7_8
16. Maurer, U.: Constructive cryptography - a new paradigm for security definitions and proofs. In: Moedersheim, S., Palamidessi, C. (eds.) TOSCA 2011. LNCS, vol. 6993, pp. 33–56. Springer, Heidelberg (2011)

17. Maurer, U., Renner, R.: Abstract cryptography. In: Chazelle, B. (ed.) The Second Symposium on Innovations in Computer Science, ICS 2011, pp. 1–21. Tsinghua University Press, January 2011
18. Maurer, U., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random Oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24638-1_2
19. Maurer, U., Rüedlinger, A., Tackmann, B.: Confidentiality and integrity: a constructive perspective. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 209–229. Springer, Heidelberg (2012). doi:10.1007/978-3-642-28914-9_12
20. Maurer, U., Tackmann, B.: On the soundness of authenticate-then-encrypt: formalizing the malleability of symmetric encryption. In: Proceedings of the 17th ACM Conference on Computer and Communication Security (ACM-CCS), pp. 505–515. ACM, October 2010
21. Nisan, N., Zuckerman, D.: Randomness is linear in space. J. Comput. Syst. Sci. **52**, 43–52 (1996)
22. Portmann, C., Matt, C., Maurer, U., Renner, R., Tackmann, B., Boxes, C.: Quantum information-processing systems closed under composition. eprint, arXiv:1512.02240 (2016)
23. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with composition: limitations of the indifferentiability framework. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 487–506. Springer, Heidelberg (2011). doi:10.1007/978-3-642-20465-4_27
24. Vadhan, S.P.: On constructing locally computable extractors and cryptosystems in the bounded storage model. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 61–77. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45146-4_4
25. Wullschleger, J.: Bitwise quantum min-entropy sampling and new lower bounds for random access codes. In: Bacon, D., Martin-Delgado, M., Roetteler, M. (eds.) TQC 2011. LNCS, vol. 6745, pp. 164–173. Springer, Heidelberg (2014). doi:10.1007/978-3-642-54429-3_11