Notions of Black-Box Reductions, Revisited

Paul Baecher[1], Christina Brzuska[2], and Marc Fischlin[1]

Abstract. Reductions are the common technique to prove security of cryptographic constructions based on a primitive. They take an allegedly successful adversary against the construction and turn it into a successful adversary against the underlying primitive. To a large extent, these reductions are black-box in the sense that they consider the primitive and/or the adversary against the construction only via the input-output behavior, but do not depend on internals like the code of the primitive or of the adversary. Reingold, Trevisan, and Vadhan (TCC, 2004) provided a widely adopted framework, called the RTV framework from hereon, to classify and relate different notions of black-box reductions.
Having precise notions for such reductions is very important when it comes to black-box separations, where one shows that black-box reductions cannot exist. An impossibility result, which clearly specifies the type of reduction it rules out, enables us to identify the potential leverages to bypass the separation. We acknowledge this by extending the RTV framework in several respects using a more fine-grained approach. First, we capture a type of reduction—frequently ruled out by so-called meta-reductions—which escapes the RTV framework so far. Second, we consider notions that are "almost black-box", i.e., where the reduction receives additional information about the adversary, such as its success probability. Third, we distinguish explicitly between efficient and inefficient primitives and adversaries, allowing us to determine how relativizing reductions in the sense of Impagliazzo and Rudich (STOC, 1989) fit into the picture.

1 Introduction

A fundamental question in cryptography refers to the possibility of constructing one primitive from another one. For some important primitives like one-way functions, pseudorandom generators, pseudorandom functions, and signature schemes it has been shown that one can be built from the other one [24,17,34]. For other primitives, however, there are results separating primitives like key agreement or collision-resistant hash functions from one-way functions [26,36].

Separations between cryptographic primitives usually refer to a special kind of reductions called *black-box* reductions. These reductions from a primitive $\mathcal{P}$ to another primitive $\mathcal{Q}$ treat the underlying primitive $\mathcal{Q}$ and/or the adversary as a black box. Reingold et al. [33] suggested a taxonomy for such reductions which can be divided roughly into three categories:

Fully Black-Box Reductions: A fully black-box reduction $\mathcal{S}$ is an efficient algorithm that transforms any (even inefficient) adversary $\mathcal{A}$, breaking any instance G^f of primitive $\mathcal{P}$, into an algorithm $\mathcal{S}^{\mathcal{A},f}$ breaking the instance f of $\mathcal{Q}$. Here, the reduction treats both the adversary as well as the primitive as a black box, and G is the (black-box) construction out of f.

Semi Black-Box Reductions: In a semi black-box reduction, for any instance G^f of $\mathcal{P}$, if an efficient adversary $\mathcal{A}^f$ breaks G^f, then there is an algorithm $\mathcal{S}^f$ breaking the instance f of $\mathcal{Q}$. Here, S^f can be tailor-made for $\mathcal{A}$ and f.

Weakly Black-Box Reductions: In a weakly black-box reduction, for any instance G^f of $\mathcal{P}$, if an efficient adversary $\mathcal{A}$ (now without access to f) breaks G^f, then there is an algorithm $\mathcal{S}^f$ breaking the instance f of $\mathcal{Q}$.

Reingold et al. [33] indicate that the notion of weakly black-box reductions is close to free reductions (with no restrictions), such that separation results for this type of reduction are presumably hard to find. They discuss further notions like "$\forall\exists$ versions" of the above definitions, where the construction G does not make black-box use of f but may depend arbitrarily on f, and relativizing reductions where security of the primitives should hold relative to any oracle. We discuss these notions later in more detail.

1.1 Black-Box Separation Techniques

Known black-box separations usually obey the following two-oracle approach: to separate $\mathcal{P}$ from $\mathcal{Q}$ one oracle essentially makes any instance of $\mathcal{P}$ insecure, whereas the other oracle implements an instance of $\mathcal{Q}$. It follows that one cannot build (in a black-box way) $\mathcal{P}$ out of $\mathcal{Q}$. For example, Impagliazzo and Rudich [26] separate key agreement from one-way permutations by using a PSPACE-complete oracle to break any key agreement, and a random permutation oracle to realize the one-way permutation. This type of separation rules out so-called relativizing reductions, and are in this case equivalent to semi black-box reductions via embedding of the PSPACE-complete oracle into the black-box primitive [33].

Later, Hsiao and Reyzin [25] consider simplified separations for fully black-box reductions. Roughly speaking, they move the breaking oracle into the adversary such that the reduction can only access this oracle through the adversary (instead of directly, as in [26]). Because this makes separations often much more elegant this technique has been applied successfully for many other primitives, e.g., [11,20,21,27,5,13,29,28,3].

Interestingly, recently there has been another type of separations based on so-called meta-reduction techniques, originally introduced by Boneh and Venkatesan [6], and subsequently used in many other places [9,30,22,14,31,15,10,35,12]. Such meta-reductions take an alleged reduction from $\mathcal{P}$ to $\mathcal{Q}$ and show how to use such a reduction to break the primitive $\mathcal{P}$ directly, simulating the adversary for the reduction usually via rewinding techniques. It turns out that meta-reductions are somewhat dual to the above notions for black-box reductions. They usually work against reductions which use the adversary only in a black-box way, whereas the reduction often receives the description of the primitive f. This notion then escapes the treatment in [33].

An interesting side effect when the reduction is given the description of f is that then the separation technique still applies to concrete problems like RSA or discrete logarithms, and to constructions which use zero-knowledge proofs relative to f. Such zero-knowledge proofs often rely on Karp reductions of f to an NP-complete language and therefore on the description of f. In contrast, for black-box use of the primitive f such constructions do not work in general, although some of them can still be rescued by augmenting the setup through a zero-knowledge oracle which allows to prove statements relative to f (see [7]). We also remark that in some cases, such as Barak's ingenious result about non-black-box zero-knowledge and related results [2,4], the security relies on the code of the adversary instead, though.

1.2 Our Results

The purpose of this paper is to complement the notions of fully, semi, and weakly black-box reductions. We also introduce a more fine-grained view on the involved algorithms, such as the distinction between efficient and non-efficient adversaries, or the question in how far the framework can deal with the reduction having partial knowledge about the adversary. We also formalize meta-reductions in the new framework and thus enable classification of this type of separation results. We give a comprehensive picture of the relationship of all reduction types. Next we discuss these results in more detail.

As explained above, we extend the classification of black-box reductions to other types, like meta-reductions relying on black-box access to the adversary but allowing to depend on the primitive's representation. This, interestingly, also affects the question of efficiency of the involved algorithms. That is, we believe that reductions for inefficient and efficient adversaries and primitives should in general not be resumed under a single paradigm, if efficiently computable primitives like one-way functions are concerned. For this class, classical separations techniques such as the embedding of the adversarially exploited PSPACE-complete oracle into the primitive do not work anymore. Hence, in this case one would need to additionally rely on a complexity assumption, such as for example in the work by Pass et al. [32]. To testify the importance of the distinction between efficient and inefficient adversaries in black-box reductions we show for example that black-box use of efficient adversaries is equivalent to non-black-box use, for constructions and reductions which are non-black-box for the primitive. Another example where the non-black-box use of the primitive turned out to be crucial is in the work by Mahmoody and Pass [29] where non-interactive commitments are built from non-black-box one-way functions, whereas constructions out of black-box one-way functions provably fail.

Another issue we address is the question in how far information about the adversary available to the reduction may be considered as covered by black-box notions. Technically speaking, the running time of an efficient fully black-box reduction must not depend on the adversary's running time, and thus for example on the number of queries the adversary makes to the primitive. Else, one would need to use a non-standard cost model for the reduction's oracle queries to the adversary. We overcome this dilemma by allowing the reduction's running time (or other parameters) to depend on adversarial parameters, such as the number of queries the adversary makes when attacking primitive $\mathcal{P}$. We call this a parameter-dependent reduction.

We can go even one step further and give the reduction the adversarial parameters as input. This is for example necessary to allow the reduction to depend on the adversary's success probability, but otherwise treating the adversary as a black box. A well-known example of such an "almost" fully black box reduction is the security proof of the Goldreich–Levin hardcore predicate [19], attributed to Rackoff in [16]. This reduction depends on the adversary's success probability for a majority decision, but does not rely on any specifics of the adversary nor the function to be inverted itself. We call such reductions parameter-aware.

We note that it is up to the designer of the reduction or separation to precisely specify the parameters. Such parametrized black-box reductions potentially allow authors to counteract the idea behind black-box reductions by placing the adversary's code in the parameters and thus making the reduction depend on the adversary again (via a universal Turing machine). But we assume that such trivial cases can be easily detected *if the dependency is signalized clearly*, just as in the case of a trivial reduction of a cryptographic protocol to its own security. So far, however, literature seems to be often less explicit on which parameters the reduction is based upon, and if the reduction should really count as black box. Stating reductions clearly as parametrized black-box reduction should make this more prominent.

In summary, we thus provide a more comprehensive and fine-grained view on black-box constructions and separations, allowing to identify and relate separations more clearly. In our view, two important results are that we can place relativizing reductions between non-black box constructions for inefficient and for efficient adversaries, and that for efficient adversaries the question of the reduction having black-box access to the adversary, or allowing full dependency on the adversary, is irrelevant. This holds as long as the construction and reduction itself make non-black-box use of the primitive. From a technical point of view, one of the interesting results is clearly that any reduction from the indistinguishability of hardcore bits to one-wayness, such as in the Goldreich–Levin case [19], must depend on the adversary's success probability (and thus needs to be parametrized).

Nevertheless, we view the contributions in this paper to be primarily on the conceptual side. Given the central role that reductions play in modern cryptography, our impression is that a fundamental—but rather coarse—work like [33] leaves some potential for refinement. Let us demonstrate this by the following two examples.

The Hsiao-Reyzin separation [25] is often termed fully black-box (according to [33]) and considered to be a rather "weak" separation. Our more fine-grained picture shows that the separation is actually of the NNN type and thus rather a low-level (i.e., strong) separation which cannot be bypassed through, say, any non-black-box technique in either direction of the CAP dimensions. Hence, non-black-box techniques cannot be used to sidestep this impossibility result; looking at efficient adversaries/primitives may help, though.

Similarly, according to [33], meta-reductions only rule out BBB reductions. So, the framework does not make any distinction between the strength of meta-reductions and some oracle separations. However, most meta-reductions today rely on unbounded adversaries. As our paper exhibits one might circumvent such meta-reductions by switching to the "parallel universe" of efficient adversaries, identifying exactly what kind of black-boxness is still admissible according to our implications (e.g., if the meta-reduction rules out NBN reductions, then one may still manage to find an NBNa reduction).

Thus, our framework reveals that some impossibility results actually rule out a great class of reductions and points exactly to the remaining few leverages to give positive results.

2 Notions of Reducibility

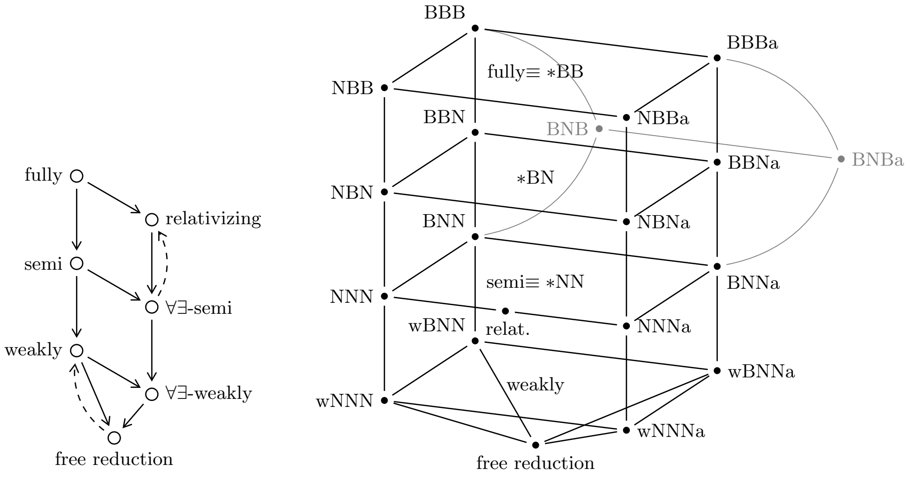
We extend the original framework for notions of reducibility by Reingold, Trevisan and Vadhan [33]. Since we augment the basic notions in various directions, we find it useful to use a different terminology for the reduction types. Instead of referring the original terms fully, semi, weakly, and their ∀∃ variants, we use a more descriptive three-character "CAP" notation with words from the language $\{\mathrm{B}, \mathrm{N}\}^3$, with the meaning that a 'B' in the first position (the C-position) refers to the fact that the Construction is black-box, in the second A-position that the Adversary is treated as a black-box by the reduction, and in the third P-position the Primitive is treated as black-box by the reduction. Accordingly, an entry 'N' stands for a non-black-box use. From each combination of constraints, we then derive the order of quantification to obtain the actual definitions.

Hence, a fully black-box reduction in the RTV framework corresponds to a BBB-reduction in our notation, and a ∀∃ fully black-box reduction is an NBB-reduction in our sense. The CAP notation will later turn out to be handy when showing implications from an XYZ-reduction to an $\widehat{X}\widehat{Y}\widehat{Z}$-reduction, whenever $\widehat{X}\widehat{Y}\widehat{Z}$ is pointwise at most as large as XYZ (with N being smaller than B). It also allows to see immediately that the RTV framework only covers a fraction of all 8 possibilities for the CAP choices (although the NNB type is actually not meaningful, as we discuss later), and that we fill in the missing types BBN, as often ruled out by meta-reductions, and the dual BNB type where the primitive but not the adversary is treated as a black-box.

Extending the RTV framework in another dimension, we differentiate further based on the (in)efficiency of the primitives and adversaries. We append the suffix 'a' to denote an efficiency requirement on the adversary, i.e., a BBBa-reduction only works for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$, while a BBB-reduction is a fully black-box reduction that transforms *any* adversary $\mathcal{A}$ into an adversary against another primitive. Likewise, we use 'p' to indicate that we restrict primitives to those which are efficiently computable; the suffix 'ap' naturally combines both restrictions.

2.1 Overview

At the top of the RTV hierarchy there are fully black-box reductions—or, BBB-reductions in our CAP terminology. These BBB-reductions from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$ is a pair $(G, \mathcal{S})$ consisting of a construction G and a reduction algorithm $\mathcal{S}$. Both treat the primitive in a black-box way and the reduction treats the adversary in a black-box way. So, for *all* adversaries $\mathcal{A}$ and *all* instantiations f of the primitive $\mathcal{Q}$, we have that, if the adversary $\mathcal{A}^f$ breaks G^f, then the reduction $\mathcal{S}^{\mathcal{A},f}$ with black-box access to the adversary $\mathcal{A}$ and f breaks the implementation f. As a consequence, the existence of primitive $\mathcal{Q}$ implies the existence of the primitive $\mathcal{P}$.



(a)

(b)

Fig. 1: (a) shows the relation of notions in the RTV framework. The dashed arrows indicate equivalence for a restricted class of reductions. In our framework (b), it is instructive to look at the vertical planes for fully, ∗BN, semi, and weakly. The left side corresponds to inefficient adversaries, the right side to efficient ones. The front is the ∀∃ layer, i.e., non-black-box constructions, and the back corresponds to black-box constructions. As NNB-reductions are not meaningful, we only need the BNB type (in gray). The w∗NN notions are equivalent to the weakly notions of RTV. A notion A implies notion B if there is a path of edges between both notions and notion A is located above notion B.

The RTV framework discusses several variants and relaxations of fully black-box reductions, called semi, weakly, and relativizing reductions. For semi black-box reductions (aka. BNN-reductions) $\mathcal{S}$ can depend on both, the description of the adversary $\mathcal{A}$ and of the instantiation f, and only the construction is black-box. For weakly black-box reductions (which are also of the BNN type) the adversary is additionally restricted to be efficient and does not get access oracle to the primitive (but may depend on it). There is a relativizing reduction between the primitives $\mathcal{P}$ and $\mathcal{Q}$, if for all oracles, the primitive $\mathcal{P}$ exists relative to an oracle whenever $\mathcal{Q}$ exists relative to this oracle. Figure 1a illustrates the relationships between these classes.

We augment the RTV framework by new classes which represent, among others, reductions that are ruled out by certain meta-reductions. That is, we first introduce the notion of BBN-reductions where $\mathcal{S}$ has to work for all (black-box) adversaries, but may depend on the code of f. The other case, where $\mathcal{S}$ is universal for all black-box f but may depend on $\mathcal{A}$, is called BNB-reduction. In both cases the initial 'B' indicates that the construction still makes black-box calls to the primitive. We remark that semi black-box and weakly black-box reductions are of the same BNN type in our notation as they only differ in regard to the adversary's access to f. As pointed out in [33] weakly black-box reductions are close to free reductions, and black-box separations are presumably only possible at the semi level or above. In a sense, our CAP model only captures these levels above, and other types like free or relativizing (or weakly) reductions are special. For the sake of completeness, we symbolically denote (but do not define) weakly reductions w$*$NN and remark that they essentially correspond to the weakly type of RTV. Note that weakly black-box reductions are called mildly black-box in some versions of RTV.

The RTV framework also considers the type of construction (black-box vs. non-black-box) and uses the prefix $\forall\exists$ to indicate that construction G does not need to be universal for all f but can, instead, depend on the description of f. In our CAP terminology this "flips" the initial 'B' to an 'N'. By this, we get 8 combinations, of which 7 are reasonable. The notion of NNB-reduction is not meaningful, because we are restricted by the following dependencies: the construction may depend on the primitive, the reduction may depend on the adversary, and the reduction should be universal for the primitive. Thus, there is only one way to order the quantifiers ($\forall\mathcal{A}\exists\mathcal{S}\forall f\exists G$) which does not seem to be a reasonable notion of security, because the construction can now depend on the adversary (and if it does not, we are in the other cases).

We note that the notion of an NBB-reduction is debatable, because it relies on a universal reduction which works for arbitrary constructions. That is, the order of quantifiers is $\exists\mathcal{S}\forall f\exists G\forall\mathcal{A}$. But since there may indeed be such reductions, say, a trivial reduction from a primitive to itself, we do not exclude this type of reduction here.

2.2 Definitions of Reductions

We next provide definitions of BBB (aka. fully black-box) reductions, BNB and BBN reductions; the remaining definitions are delegated to the full version of this paper [1].

A primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ is represented as a set $\mathcal{F}_\mathcal{Q}$ of random variables, corresponding to the set of implementations, and a relation $\mathcal{R}_\mathcal{Q}$ that describes the security of the primitive as tuples of random variables, i.e., a random variable $\mathcal{A}$ is said to break an instantiation $f \in \mathcal{F}_\mathcal{Q}$, if and only if $(f, \mathcal{A}) \in \mathcal{R}_\mathcal{Q}$.

CAP	[33] name	Remark(s)
BBB	fully	known meta reductions: [8,22]
BBN		
BNB		known reduction: [19]
BNN	semi (weakly)	
NBB	∀∃-fully	formally not defined in [33], only “trivial” reductions
NBN		known meta reductions: [6,22,14,31]
NNB		not meaningful
NNN	∀∃-semi (∀∃-weakly)	

Fig. 2: CAP indicates whether the construction (C), the adversary in the reduction (A), or the primitive in the reduction (P) is treated in a black-box (B) or non-black-box (N) way.

Following [33], we say that a primitive exists if there is a polynomial-time computable instantiation $f \in \mathcal{F}_Q$ such that no polynomial-time random variable breaks the primitive. Indeed, [33] demand that primitive sets $\mathcal{F}_Q$ are non-empty, but do not motivate this further. We drop this requirement here as reductions explicitly depend on primitives, such that one can enforce such non-empty sets by investigating only such primitives if necessary. Still, we remark that all our implications and separations would work in this case as well.

For efficient primitives or adversaries we stipulate that the random variable is efficiently computable in the underlying machine model which, unless mentioned differently, is assumed to be Turing machines; the results remain valid for other computational models like circuit families. Considering security as a general relation allows to cover various (if not all) notions of security: games such as CMA-UNF for unforgeability of signature schemes, simulation-based notions such as implementing a UC commitment functionality, and even less common notions such as distributional one-way functions. In the full version of this paper [1] we define as examples the DDH assumption (cast as a primitive) and the indistinguishability of the ElGamal encryption scheme . We also present the reduction from the ElGamal encryption to the DDH assumption and identify its type according to our terminology. Note that a “black-boxness” consideration in this particular setting is indeed meaningful, because the DDH assumption can hold in a variety of group distributions and the concrete procedures that sample from these group distributions can be abstracted away. In the full version we discuss another example of weak one-way functions (and the construction of strong one-way functions [37]) to highlight that the type of reduction hinges on the exact formulation of the underlying primitive: the construction and the reduction is then either of the NBN type or of the BBB kind.

We stress that the distinction between the *mathematical object* describing the adversary as a random variable, and its *implementation* through, say, a Turing machine is important here; else one can find counter examples to implications among black-box reduction types proven in [33]. The problem is roughly that the relation may simply be secure because it syntactically excludes all oracle Turing machines $\mathcal{A}^f$. We note that Reingold et al. [33] indeed define the relations for adversarial *machines*. Our discussion in [1] shows that only interpreting such adversaries as abstract objects sustains the implications in [33]. However, for sake of convenience, we too often refer to $\mathcal{A}^f$ by the machine implementing it, even when considering the mathematical random process for relations $\mathcal{R}_\mathcal{Q}$. In this case it is understood that we actually mean the abstract random variable instead. The same holds for the constructions of the form G^f and the first component of the security relations. An alternative approach, also presented in the full version is to rely on machines, but to formally introduce semantical relations. These relations roughly require that, for any algorithm $\mathcal{A}$ in $\mathcal{R}_\mathcal{Q}$, any oracle machine $\mathcal{A}^f$ with the same output behavior is also in $\mathcal{R}_\mathcal{Q}$.

We now turn to the actual definitions. Many (but not all) reductions in cryptography fall into the class of so-called fully black-box reductions, a very restrictive notion, where the reduction algorithm is only provided with black-box access to the primitive and the adversary. Throughout the paper, if there is a XYZ-reduction from primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$, we notate this as $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$-reduction. Note that the correctness is requirement is the same for all definitions. Therefore, the shorthand notation towards the end of each definition covers the security requirement only.

Definition 1 **($(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBB or Fully Black-Box Reduction).** *There exists a* fully black-box (or BBB-)reduction *from a primitive $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ to a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ if there exist probabilistic polynomial-time oracle algorithms G and $\mathcal{S}$ such that:*

Correctness. *For every $f \in \mathcal{F}_\mathcal{Q}$, it holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

Security. *For every implementation $f \in \mathcal{F}_\mathcal{Q}$ and every machine $\mathcal{A}$, if $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

$$\exists \mathsf{PPT} G \ \exists \mathsf{PPT} \mathcal{S} \ \forall f \in \mathcal{F}_\mathcal{Q} \ \forall \mathcal{A} \ \left((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}\right).$$

Definition 2 **($(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNB-reduction).** *There exists a* BNB-reduction *from a primitive $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ to a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ if there exists a probabilistic polynomial-time oracle machine G such that:*

Correctness. *For every $f \in \mathcal{F}_\mathcal{Q}$, it holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

Security. *For every machine $\mathcal{A}$, there is a probabilistic polynomial-time oracle algorithm $\mathcal{S}$ such that: for every implementation $f \in \mathcal{F}_\mathcal{Q}$, if $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

$$\exists \mathsf{PPT} G \ \forall \mathcal{A} \ \exists \mathsf{PPT} \mathcal{S} \ \forall f \in \mathcal{F}_\mathcal{Q} \ \left((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}\right).$$

Definition 3 **($(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBN-reduction).** *There exists a* BBN-reduction *from a primitive $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ to a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ if there exists a probabilistic polynomial-time oracle machine G such that:*

Correctness. *For every $f \in \mathcal{F}_\mathcal{Q}$, it holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

Security. *For every implementation $f \in \mathcal{F}_\mathcal{Q}$, there is a probabilistic polynomial-time oracle algorithm $\mathcal{S}$ such that for every machine $\mathcal{A}$, if $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

$$\exists \mathsf{PPT} G \ \forall f \in \mathcal{F}_\mathcal{Q} \ \exists \mathsf{PPT} \mathcal{S} \ \forall \mathcal{A} \ \ ((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}).$$

Name	Summary of definition				
BBB	$\exists\mathsf{PPT}G$	$\exists\mathsf{PPT}\mathcal{S}$	$\forall f \in \mathcal{F}_\mathcal{Q}$	$\forall\mathcal{A}$	$((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q})$
BNB	$\exists\mathsf{PPT}G$	$\forall\mathcal{A}$	$\exists\mathsf{PPT}\mathcal{S}$	$\forall f \in \mathcal{F}_\mathcal{Q}$	$((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q})$
BBN	$\exists\mathsf{PPT}G$	$\forall f \in \mathcal{F}_\mathcal{Q}$	$\exists\mathsf{PPT}\mathcal{S}$	$\forall\mathcal{A}$	$((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q})$
BNN	$\exists\mathsf{PPT}G$	$\forall f \in \mathcal{F}_\mathcal{Q}$	$\forall\mathcal{A}$	$\exists\mathsf{PPT}\mathcal{S}$	$((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q})$
NBB	$\exists\mathsf{PPT}\mathcal{S}$	$\forall f \in \mathcal{F}_\mathcal{Q}$	$\exists\mathsf{PPT}G$	$\forall\mathcal{A}$	$((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q})$
NBN	$\forall f \in \mathcal{F}_\mathcal{Q}$	$\exists\mathsf{PPT}G$	$\exists\mathsf{PPT}\mathcal{S}$	$\forall\mathcal{A}$	$((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q})$
NNN	$\forall f \in \mathcal{F}_\mathcal{Q}$	$\exists\mathsf{PPT}G$	$\forall\mathcal{A}$	$\exists\mathsf{PPT}\mathcal{S}$	$((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q})$
weakly-BB	$\exists\mathsf{PPT}G$	$\forall\mathcal{A}$	$\forall f \in \mathcal{F}_\mathcal{Q}$	$\exists\mathsf{PPT}\mathcal{S}$	$((G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q})$
∀∃-weakly-BB	$\forall f \in \mathcal{F}_\mathcal{Q}$	$\exists\mathsf{PPT}G$	$\forall\mathcal{A}$	$\exists\mathsf{PPT}\mathcal{S}$	$((G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q})$

Fig. 3: Overview of notions of reducibility.

Note that we always grant $\mathcal{S}$ black-box access to f and $\mathcal{A}$, as they may not be efficiently computable so that the probabilistic polynomial-time reduction algorithm $\mathcal{S}$ cannot efficiently simulate them, even if it knows the code of f, respectively, of $\mathcal{A}$. For a compact summary of all definitions, see Figure 3; the full definitions omitted above appear in the full version of this paper [1].

2.3 Efficient versus Inefficient Algorithms

Reductions usually run the original adversary as a subroutine. However, in many cases, the reduction does not use the code of the original adversary, but instead only transforms the adversary's inputs and outputs. Thus, one might consider the reduction algorithm as having black-box access to the adversary only. An efficient reduction can then also be given black-box access to an inefficient adversary, and, maybe surprisingly, most reductions even work for inefficient adversaries. Imagine, for example, the case that one extracts a forgery against a signature scheme from a successful intrusion attack against an authenticated channel. Then, the extraction usually still works for inefficient adversaries. On the other hand, (unconditional) impossibility results often require the reduction algorithm to be able to deal with inefficient adversaries.

When designing a fine-grained framework for notions of reducibility, one thus needs to decide whether one considers efficient or inefficient adversaries. Reingold et al. [33] defined their most restrictive notion of reductions, the fully-BB-reductions (aka. BBB), for inefficient adversaries. In contrast, their notion of semi-BB-reduction treats only efficient adversaries thus making it easier to find such a reduction. Surprisingly, even for such a weak notion, they were able to give impossibility results. The reason is that they used inefficient primitives, which allow to embed arbitrary oracles so that they could make use of two-oracle separation techniques. Hence, the efficiency question does not only apply to adversaries, but also to the primitives (and, consequently, to the combination of both). We postpone the treatment of the case of primitives for now and refer the reader to Section 2.6.

We now define the efficient adversary analogues of the notions of reduction introduced in the previous section. Note that we still give the reduction $\mathcal{S}$ oracle access to the adversary $\mathcal{A}$ in *all* notions, even though the latter can be dropped for all cases where $\mathcal{S}$ depends on $\mathcal{A}$ in a non-black-box way. In these cases, a probabilistic polynomial-time reduction $\mathcal{S}$ can simulate the now likewise efficient adversarial algorithm $\mathcal{A}$. For consistency, though, we keep the $\mathcal{A}$ oracles in the definitions. To distinguish the two cases of efficient and unbounded adversaries, denote by BBBa-reduction a reduction only dealing with efficient adversaries.

Definition 4 **($(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBBa-reduction for Efficient Adversaries).** *There exists a BBBa-reduction from a primitive $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ to a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ if there exist probabilistic polynomial-time oracle machines G and $\mathcal{S}$ such that:*

Correctness. For every $f \in \mathcal{F}_\mathcal{Q}$, it holds that $G^f \in \mathcal{F}_\mathcal{P}$.

Security. For every implementation $f \in \mathcal{F}_\mathcal{Q}$ and every probabilistic polynomial-time machine $\mathcal{A}$, if $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, i.e.,

$$\exists \mathsf{PPT} G \ \exists \mathsf{PPT} \mathcal{S} \ \forall f \in \mathcal{F}_\mathcal{Q} \ \forall \mathsf{PPT} \mathcal{A} \ \left((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}\right).$$

Again, the definitions for the remaining types of reductions are presented in the full version of this paper [1].

2.4 Relations Amongst the Definitions

We first note that a number of implications among the reductions is immediately clear by simply shifting quantifiers, that is, if we have an for-all quantifier, there is certainly an existential version of the reduction in question. The next proposition states this formally, we omit the proof because it is only syntactical.

Theorem 1. *Let XYZ and $\widehat{X}\widehat{Y}\widehat{Z}$ be two types of CAP reductions such that $\widehat{X}\widehat{Y}\widehat{Z} \leq XYZ$ point-wise (where $N \leq B$) and let $\mathcal{P}$ and $\mathcal{Q}$ be two primitives. If there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$-reduction, then there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$\widehat{X}\widehat{Y}\widehat{Z}$ reduction. Also, if there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZa$-reduction, then there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$\widehat{X}\widehat{Y}\widehat{Z}a$ reduction.*

In the full version of this paper [1], we prove via means of counterexamples that for all notions for inefficient adversaries, almost all the above implications are, indeed, strict. These separations are split into a number of interesting observations. For example, we prove that the Goldreich–Levin hardcore bit reduction [19] has to depend on the success probability of the adversary (Theorem D.3 of [1]). Moreover, we show that the construction of one-way functions out of weak one-way functions ([37,18]) needs to depend on the weakness parameter of the weak one-way function (Theorem D.2 of [1]). Interestingly, some of the implications of Theorem 1 are not strict when one is concerned with reductions for efficient adversaries. Maybe surprisingly, NNNa-reductions and NBNa-reductions are, indeed, equivalent. Note that this means that knowledge of the code of the adversary does not lend additional power to the reduction:

Theorem 2 (Equivalence of NNNa and NBNa). *For all primitives $\mathcal{P}$ and $\mathcal{Q}$, there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBNa-reduction if and only if there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNNa-reduction.*

Proof. Using straightforward logical deductions, it follows that NBNa-reductions imply NNNa-reductions. For the converse direction, assume that we have two primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNNa-reduction. We now have to show that there also is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBNa-reduction, that is, we have to give a reduction algorithm $\mathcal{S}$ that depends on f in a non-black-box-way, and yet $\mathcal{S}$ depends on $\mathcal{A}$ only in a black-box way. We proceed by case distinction over f.

Case I: Suppose $f \in \mathcal{F}_{\mathcal{Q}}$ such that for all constructions G, the primitive G^f is a secure implementation of $\mathcal{P}$, i.e., for all polynomial-time adversaries $\mathcal{A}$ it holds that $(G^f, \mathcal{A}^f) \notin \mathcal{R}_{\mathcal{P}}$. Then proving the existence of a reduction satisfying the implication $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$ is trivial, as the premise of the implication is never satisfied.

Case II: For any $f \in \mathcal{F}_{\mathcal{Q}}$ outside the class described in Case I, we know that there exists a PPT construction G such that for all $\mathcal{A}$ there is a reduction algorithm $\mathcal{S}$ that satisfies $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$, and such an efficient $\mathcal{A}$ with $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}}$ exists. For any such f, we now fix a unique adversary $\mathcal{A}_f$, say, by taking the random variable $\mathcal{A}_f$ with the shortest description according to a particular encoding, such that it satisfies $(G^f, \mathcal{A}_f^f) \in \mathcal{R}_{\mathcal{P}}$. For such an $\mathcal{A}_f$ let $\mathcal{S}$ be a probabilistic polynomial-time reduction making black-box use of $\mathcal{A}_f$ such that $(f, \mathcal{S}^{\mathcal{A}_f,f}) \in \mathcal{R}_{\mathcal{Q}}$. Consider the oracle algorithm $\mathcal{S}_f^f$ that has the same behavior as $\mathcal{S}^{\mathcal{A}_f,f}$, but it incorporates $\mathcal{A}_f$ and only has an f-oracle. The algorithm $\mathcal{S}_f^f$ only depends on f, satisfies $(\mathcal{S}_f^f, f) \in \mathcal{R}_{\mathcal{Q}}$, and is implementable in probabilistic polynomial time, as $\mathcal{S}$ and $\mathcal{A}_f$ are both polynomial time algorithms. Thus, regardless of construction G, we showed that for all f there is an efficient reduction $\mathcal{S}$ such that $(\mathcal{S}^f, f) \in \mathcal{R}_{\mathcal{Q}}$, namely by choosing $\mathcal{S}^f = \mathcal{S}_f^f$. Thus, we also know that for all f, there is a reduction $\mathcal{S}$ such that for all $\mathcal{A}$, if $(\mathcal{A}, G^f) \in \mathcal{R}_P$ then $(\mathcal{S}^f, f) \in \mathcal{R}_{\mathcal{Q}}$. If now, we add an adversary oracle $\mathcal{A}$ that is ignored[1] by $\mathcal{S}$, we also obtain that $(\mathcal{S}^f, f) \in \mathcal{R}_{\mathcal{Q}}$. And thus, there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBNa-reduction. □

We now show that, while a reduction for inefficient adversaries always implies a reduction for efficient adversaries of the same type, the converse is not true in general.

[1] Here, we require the relation to be machine-independent.

Theorem 3. *For each $XYZ \in \{BBB, BNB, BBN, NBB, BNN, NBN, NNN\}$, there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$a-reduction, but no $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$-reduction.*

Proof. For the primitive $\mathcal{P}$ we consider a trivial primitive, namely the constant all-zero function, denoted f_0. Let $\mathcal{L}$ be an $\mathsf{EXPTIME}$-complete problem. The pair $(f_0, \mathcal{A})$ is in the relation $\mathcal{R}_\mathcal{P}$ if and only if the adversary $\mathcal{A}$ is a deterministic function that decides $\mathcal{L}$. Let $\mathcal{F}_\mathcal{Q}$ also consist of the set that only contains the all-zero function f_0. The relation $\mathcal{R}_\mathcal{Q}$ is empty. Observe that, for efficient adversaries, the primitive $\mathcal{P}$ is secure because $\mathsf{EXPTIME}$ strictly contains the complexity class P [23]. Thus, there is a trivial reduction since the premise of the implication

$$(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$$

is never satisfied for any efficient adversary $\mathcal{A}$. Hence, for all $XYZ \neq \text{NNB}$, there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$a-reduction. In contrast, inefficient adversaries can break the primitive $\mathcal{P}$, while, as $\mathcal{R}_Q$ is empty, no reduction $\mathcal{S}$ can break $\mathcal{R}_\mathcal{Q}$, even oracle $\mathcal{A}$. Thus, for all $XYZ \in \{\text{BBB}, \text{BNB}, \text{BBN}, \text{NBB}, \text{BNN}, \text{NBN}, \text{NNN}\}$, there is no $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$-reduction. □

2.5 Relativizing Reductions

In complexity theory as in cryptography, most reductions relativize in the presence of oracles, i.e., if a (secure instantiation of the) primitive $\mathcal{P}$ can be built from a (secure instantiation of the) primitive $\mathcal{Q}$, then the construction still works, if additionally, all parties get access to a random oracle (or any other oracle). We say that there is a *relativizing* reduction from $\mathcal{P}$ to $\mathcal{Q}$, if for all oracles Π, the primitive $\mathcal{P}$ exists relative to Π, whenever $\mathcal{Q}$ exists relative to Π. Often, separation results rule out such reductions.

Definition 5 (Relativizing Reduction). *There exists a* relativizing *reduction from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$, if for all oracles Π, the primitive $\mathcal{P}$ exists relative to Π whenever $\mathcal{Q}$ exists relative to Π. A primitive $\mathcal{P}$ is said to exist relative to Π if there is an $f \in \mathcal{F}_\mathcal{P}$ which has an efficient implementation when having access to the oracle Π such that there is no probabilistic polynomial-time algorithm $\mathcal{A}$ with $(f, \mathcal{A}^{\Pi,f}) \in \mathcal{R}_\mathcal{P}$.*

We remark that, since we define security relations over random variables and not their implementations, it is understood that the implementation of f may actually depend on Π, too. According to Reingold et al. [33], relativizing reductions are a relatively restrictive notion of reducibility that they place between BBB-reductions and NNNa-reductions. Jumping ahead, we note this is due their treatment of (in-)efficient adversaries: they require BBB-reductions to also work for inefficient adversaries $\mathcal{A}$, and so do we. In contrast, for NNNa-reductions, Reingold et al. allow the reduction algorithm to fail for inefficient adversaries $\mathcal{A}$. As we can show, *all* notions of reducibility for inefficient adversaries, including NNN-reductions, imply relativizing reductions, i.e., we can place relativizing reductions between NNN- and NNNa-reductions showing that, in fact, the notion is very liberal compared to notions of reductions that treat inefficient adversaries. In contrast, for efficient adversaries, relativizing reductions imply NNNa- and (the equivalent) NBNa-reductions and are incomparable to all stronger notions that treat efficient adversaries.

We now prove that relativizing reductions are implied by NNN-reductions for inefficient adversaries, i.e., according to Definition C.4 of [1]. The proof is inspired by Reingold et al. [33] who show that BBB-reductions imply relativizing reductions.

Theorem 4. *If there is a $(P \hookrightarrow Q)$-NNN-reduction, then there is a relativizing reduction from $\mathcal{P}$ to $\mathcal{Q}$.*

Proof. Assume there is an NNN-reduction between two primitives $\mathcal{P}$ and $\mathcal{Q}$ and assume towards contradiction that there is an oracle Π such that $\mathcal{Q}$ exists relative to this oracle, but $\mathcal{P}$ does not. Let $f \in \mathcal{F}_{\mathcal{Q}}$ be an instantiation of $\mathcal{Q}$ that is efficiently computable by an algorithm that has oracle access to Π and such that f is secure against all efficient oracle machines $\mathcal{S}$, i.e., for all probabilistic polynomial-time machines $\mathcal{S}$, one has $(f, \mathcal{S}^{\Pi}) \notin \mathcal{R}_{\mathcal{Q}}$. By assumption of a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNN-reduction, there exists a PPT oracle algorithm G for f, such that for all (possibly unbounded) adversaries $\mathcal{A}$ there is a PPT reduction algorithm $\mathcal{S}$ such that $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}}$ implies $(f, \mathcal{S}^{f,\mathcal{A}}) \in \mathcal{R}_{\mathcal{Q}}$. Now, G^f is efficiently computable relative to the oracle Π, because G is PPT and f is efficiently computable relative to Π. Since $\mathcal{P}$ does not exist relative to Π, there is an efficient adversary $\mathcal{A}$ such that $(G^f, \mathcal{A}^{\Pi}) \in \mathcal{R}_{\mathcal{P}}$, i.e., by considering that the relations are defined over random variables, setting $\mathcal{A}' := \mathcal{A}^{\Pi}$ one also has $(G^f, \mathcal{A}'^f) \in \mathcal{R}_{\mathcal{P}}$. Thus, the NNN-reduction gives an efficient reduction $\mathcal{S}$ such that $(f, \mathcal{S}^{\mathcal{A}',f}) \in \mathcal{R}_{\mathcal{Q}}$. As $\mathcal{S}$ is PPT and as f and $\mathcal{A}'$ are efficiently computable relative to oracle Π, one has that $\mathcal{S}^{\mathcal{A}',f}$ is efficiently computable relative to Π. Thus, f is not "$\mathcal{Q}$-secure" against all efficient oracle machines with oracle access to Π, yielding a contradiction. □

This proves that for inefficient adversaries, relativizing reductions are implied by NNN-reductions, the most liberal notion of reductions for inefficient adversaries. Conversely, for efficient adversaries, relativizing reductions imply NNNa and NBNa reductions, but they are not implied by any of the stronger notions. We adapt the proof due to Reingold et al. [33] for the following theorem.

Theorem 5. *If there is a relativizing reduction from $\mathcal{P}$ to $\mathcal{Q}$, then there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNNa-reduction, and a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBNa-reduction.*

Proof. It suffices to show that relativizing reductions imply NNNa-reductions for efficient adversaries, as Theorem 2 proves that NBNa-reductions and NNNa-reductions are equivalent. Assume that there is a relativizing reduction between the primitives $\mathcal{P}$ and $\mathcal{Q}$, and assume towards contradiction that there is an $f \in \mathcal{F}_{\mathcal{Q}}$ such that for all constructions G, there is an efficient adversary $\mathcal{A}$ such that for all efficient reductions algorithms $\mathcal{S}$, it holds that $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}}$ but, simultaneously, $(f, \mathcal{S}^{\mathcal{A},f}) \notin \mathcal{R}_{\mathcal{Q}}$. Then, by definition, relative to oracle f, the primitive $\mathcal{Q}$ exists, as no efficient algorithm with oracle access to f can break f.

Note that we can view $\mathcal{S}^f$ as an algorithm $\mathcal{S}'^{\mathcal{A},f}$ which does not query $\mathcal{A}$ but has the same output distribution, if viewed as random variables. By assumption, there exists a relativizing reduction between $\mathcal{P}$ and $\mathcal{Q}$, and thus, relative to the oracle f, not only $\mathcal{Q}$ exists but also the primitive $\mathcal{P}$. In particular, there is a probabilistic polynomial-time oracle machine G such that G^f implements $\mathcal{P}$ and such that for all efficient oracle machines $\mathcal{A}$, one has $(G^f, \mathcal{A}^f) \notin \mathcal{R}_{\mathcal{P}}$, i.e., $\mathcal{P}$ is secure against all efficient adversaries that get f as an oracle, a contradiction. □

Theorem 6. *For $XYZ \in \{BBB, NBB, BBN, BNB, BNN, NBN, NNN\}$, there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-XYZa-reduction for efficient adversaries, but no relativizing reduction.*

Proof. We show that BBBa-reductions do not imply relativizing reductions; as BBBa-reductions imply the "lower level" reductions, the other cases follow. We use the same approach as for Theorem 3.

Let $\mathcal{Q}$ be the primitive that contains the constant 0-function f_0. We define the relation $\mathcal{R}_{\mathcal{P}}$ such that $\mathcal{P}$ is trivially secure against all *efficient* adversaries, namely, let $\mathcal{L}$ be an $\mathsf{EXPTIME}$-complete language, then $(f_0, \mathcal{A})$ is in $\mathcal{R}_{\mathcal{P}}$ if $\mathcal{A}$ is a deterministic function and decides $\mathcal{L}$. As the complexity class P is strictly contained in $\mathsf{EXPTIME}$, no efficient adversary can break $\mathcal{P}$. Let $\mathcal{Q}$ also be the primitive that contains the constant 0-function f_0, but with a different relation, namely $\mathcal{R}_{\mathcal{Q}}$ is empty. In particular, no adversary can break $\mathcal{Q}$. Hence, there is a trivial $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBBa-reduction, because the premise of the implication

$$(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$$

is never satisfied for efficient adversaries and the implication is thus trivially true. In contrast, there is no relativizing reduction between the two primitives. That is, assume, we add an oracle that decides the $\mathsf{EXPTIME}$-complete language $\mathcal{L}$, then relative to this oracle, there are suddenly efficient adversaries that break $\mathcal{P}$. However, as $\mathcal{R}_{\mathcal{Q}}$ is still empty, there cannot be a reduction $\mathcal{S}$ in this oracle world, giving us a contradiction. □

Reingold et al. [33] note that BNNa-reductions for efficient adversaries and relativizing reductions are often equivalent. In particular, they prove that if a primitive $\mathcal{Q}$ allows any oracle Π to be embedded into it, then a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNNa-reduction implies a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-relativizing reduction. However, *efficient* primitives $\mathcal{Q}$ such as one-way functions (as opposed to random oracles, for example), are not known to satisfy this property. We discuss this issue in more detail in the following section about efficient primitives.

2.6 Efficient Primitives versus Inefficient Primitives

A reduction for *efficient* primitives is a reduction that only works if $f \in \mathcal{F}_{\mathcal{Q}}$ is efficiently implementable, i.e., in probabilistic polynomial-time. If we make this distinction then, according to Figure 1, we unfold another dimension (analogously to the case of efficient adversaries). As we discuss below our results for non-efficient primitives hold in this "parallel universe" of efficient primitives as well, and between the two universes there are straightforward implications and separations (as in the case of efficient and inefficient adversaries).

Technically, one derives the efficient primitive version XYZp of an XYZ-reduction by replacing all universal quantifiers over primitives f in $\mathcal{F}_{\mathcal{Q}}$ by universal quantifiers that are restricted to efficiently implementable f in $\mathcal{F}_{\mathcal{Q}}$. More concretely, we replace $\forall f \in \mathcal{F}_{\mathcal{Q}}$ by the term $\forall \mathsf{PPT} f \in \mathcal{F}_{\mathcal{Q}}$. For example, the notion of a BBBp-reduction then reads as follows:

Definition 6 (($\mathcal{P} \hookrightarrow \mathcal{Q}$)-BBBp or Fully Black-Box Reduction for Efficient Primitives). *There exists a* fully black-box (or BBBp-)reduction *for efficient primitives from $\mathcal{P} = (\mathcal{F}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$ to $\mathcal{Q} = (\mathcal{F}_{\mathcal{Q}}, \mathcal{R}_{\mathcal{Q}})$ if there exist probabilistic polynomial-time oracle algorithms G and $\mathcal{S}$ such that:*

Correctness. *For every polynomial-time computable function $f \in \mathcal{F}_{\mathcal{Q}}$, it holds that $G^f \in \mathcal{F}_{\mathcal{P}}$.*

Security. *For every polynomial-time computable function $f \in \mathcal{F}_{\mathcal{Q}}$ and every machine $\mathcal{A}$, if $(G^f, \mathcal{A}) \in \mathcal{R}_{\mathcal{P}}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$, i.e.,*

$$\exists \mathsf{PPT} G \ \exists \mathsf{PPT} \mathcal{S} \ \forall \mathsf{PPT} f \in \mathcal{F}_{\mathcal{Q}} \ \forall \mathcal{A} \ \left((G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}\right).$$

In the same manner, for any XYZ-reduction, we can define the corresponding XYZp-reduction. Similarly, one can transform all reduction types XYZa for efficient adversaries into reduction types XYZap for efficient adversaries *and efficient primitives*. Most relations that this paper establishes for XYZ-reductions and XYZa-reductions also hold for XYZp- and XYZap-reductions, except for the relation to relativizing reductions, where only some of the results carry over, see Theorem 2.15 of [1]. Building on proof ideas of Theorem 3, we also establish in Theorem 2.14 of [1] that the implication from reductions for arbitrary primitives to reductions for efficient primitives is strict. We refer the reader to the full version of this paper [1] for formal theorem statements, proofs and further discussion of the relations of reductions for efficient primitives.

3 Parametrized Black-Box Reductions

Many reductions in cryptography commonly classified as "black box" technically do not fall in this class, as a black box reduction algorithm must not have any information about the adversary beyond the input/output behavior, except for the sole guarantee that it breaks security with non-negligible probability. Strictly speaking, this excludes information such as running time, number of queries, or the actual success probability of a given adversary. This prompts the question of what the "natural" notion of a black-box reduction should be. Not surprisingly, the answer is a matter of taste, just like the question whether fully black-box or semi black-box is the "right" notion of a black-box reduction. As in the case of different notions of black-box reductions, we can nonetheless give a technically profound, and yet easy-to-use notion of *parametrized* black-box reductions (of any type). In the full version [1] we motivate and formalize two different degrees of parameterization by distinguishing between parameter-*aware* and parameter-*dependent* reductions. The difference is essentially whether or not the reduction algorithm receives the parameter values as input.

We note that parametrized black-box reductions and separations rely critically on the specific parameters. In particular, some of our separations consider reductions that are required to depend on, say, the success probability of the adversary, as in the case of the Goldreich–Levin hardcore bit. This separation does not carry over to the parametrized case. In contrast, separations for efficient/inefficient adversaries as well as the theorems on relativized reductions still apply.

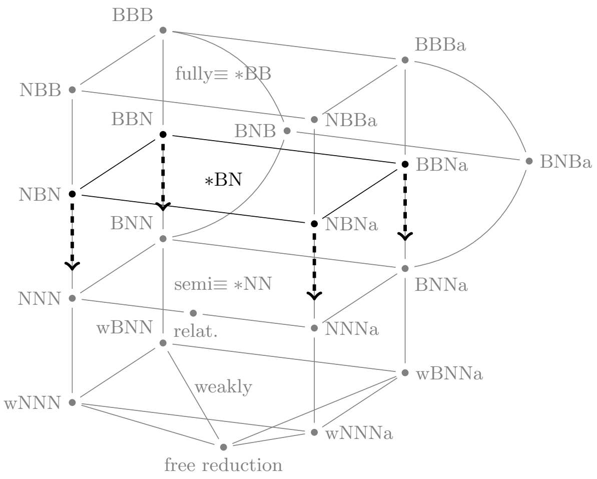


Fig. 4: The effect of parametrization (in the case of ∗BN-reductions). Parametrized counterparts of each type partly descend towards the corresponding ∗NN-reduction with full dependency on the adversary.

More pictorially, one can imagine parametrized black-box reductions in light of our Figure 1 as descending from the ∗B∗ plane for black-box adversaries towards the ∗N∗ plane, where the reduction can completely depend on the adversary, see Figure 4. The parameters and the distinction between awareness and dependency determines how far one descends. Analogously, parametrization for BBB-reductions means to descend from the top node BBB to BNB (also in the case of efficient adversaries). As such, it is clear that implications along edge paths remain valid, e.g., a parametrized NBN-reduction still implies a NNN-reduction.

The case of NBB-reductions, however, shows that parametrization cannot fully bridge the gap to NNB-reductions. As explained before, the latter type with quantification $\forall\mathcal{A}\exists\mathcal{S}\forall f\exists G$ does not seem to be meaningful, because the construction G would now depend on the adversary $\mathcal{A}$. Parametrization of NBB-reductions (with quantification $\exists\mathcal{S}\forall f\exists G\forall\mathcal{A}$) still makes sense, though, because the dependency of $\mathcal{S}$ on the adversary is only through the running time or the input. Put differently, the parametrization allows for the "admissible non-black-boxness" for the NBB type of reduction. If one parametrizes the black-box access to the primitive, either for the construction or the reduction, then this parametrization corresponds to a (partial) shift from back plane to the front plane resp. from the top $*$BB plane to the $*$BN plane. In the full version of this paper [1], we establish formal relationships between parameter-awareness and parameter-depedency.

4 Conclusion

We provide a comprehensive framework to classify black-box reductions more precisely. We believe that this is important to fully understand and appreciate the implications and limitations of black-box separation results. In particular, we point out how subtleties such as different possibilities to define a primitive, the distinction between efficient and non-efficient adversaries and primitives, or parameterization, affect the results. Such details have previously been often neglected, and our work draws more attention to these issues.

Acknowledgements

We thank the anonymous reviewers and Pooya Farshim for valuable comments on previous versions of this work. Paul Baecher is supported by grant Fi 940/4-1 of the German Research Foundation (DFG). Christina Brzuska was supported in part by the Israel Ministry of Science and Technology (grant 3-9094) and by the Israel Science Foundation (grant 1155/11 and grant 1076/11); parts of the work done while being at TU Darmstadt and CASED (`www.cased.de`). Marc Fischlin is supported by the Heisenberg Program of the DFG under grant Fi 940/3-1.

References

1. Baecher, P., Brzuska, C., Fischlin, M.: Notions of black-box reductions, revisited. Cryptology ePrint Archive, Report 2013/101 (2013), http://eprint.iacr.org/
2. Barak, B.: How to go beyond the black-box simulation barrier. In: 42nd FOCS. pp. 106–115. IEEE Computer Society Press (Oct 2001)
3. Barhum, K., Holenstein, T.: A cookbook for black-box separations and a recipe for UOWHFs. In: TCC 2013. LNCS, Springer (2013)
4. Bitansky, N., Paneth, O.: From the impossibility of obfuscation to a new non-black-box simulation technique. In: Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) 2012. pp. 223–232. IEEE Computer Society Press (2012)

5. Boldyreva, A., Cash, D., Fischlin, M., Warinschi, B.: Foundations of non-malleable hash and one-way functions. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 524–541. Springer (Dec 2009)
6. Boneh, D., Venkatesan, R.: Breaking RSA may not be equivalent to factoring. In: Nyberg, K. (ed.) EUROCRYPT'98. LNCS, vol. 1403, pp. 59–71. Springer (May / Jun 1998)
7. Brakerski, Z., Katz, J., Segev, G., Yerukhimovich, A.: Limits on the power of zero-knowledge proofs in cryptographic constructions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 559–578. Springer (Mar 2011)
8. Bresson, E., Monnerat, J., Vergnaud, D.: Separation results on the "one-more" computational problems. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 71–87. Springer (Apr 2008)
9. Coron, J.S.: Optimal security proofs for PSS and other signature schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer (Apr / May 2002)
10. Dodis, Y., Haitner, I., Tentes, A.: On the instantiability of hash-and-sign RSA signatures. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 112–132. Springer (Mar 2012)
11. Dodis, Y., Oliveira, R., Pietrzak, K.: On the generic insecurity of the full domain hash. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 449–466. Springer (Aug 2005)
12. Fischlin, M., Fleischhacker, N.: Limitations of the meta-reduction technique: The case of schnorr signatures. In: EUROCRYPT 2013. LNCS, Springer (2013)
13. Fischlin, M., Lehmann, A., Ristenpart, T., Shrimpton, T., Stam, M., Tessaro, S.: Random oracles with(out) programmability. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 303–320. Springer (Dec 2010)
14. Fischlin, M., Schröder, D.: On the impossibility of three-move blind signature schemes. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 197–215. Springer (May 2010)
15. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 99–108. ACM Press (Jun 2011)
16. Goldreich, O.: Foundations of Cryptography: Basic Applications, vol. 2. Cambridge University Press, Cambridge, UK (2004)
17. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. Journal of the ACM 33, 792–807 (1986)
18. Goldreich, O., Impagliazzo, R., Levin, L.A., Venkatesan, R., Zuckerman, D.: Security preserving amplification of hardness. In: FOCS. pp. 318–326. IEEE Computer Society (1990)
19. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: 21st ACM STOC. pp. 25–32. ACM Press (May 1989)
20. Haitner, I., Hoch, J.J., Reingold, O., Segev, G.: Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In: 48th FOCS. pp. 669–679. IEEE Computer Society Press (Oct 2007)
21. Haitner, I., Holenstein, T.: On the (im)possibility of key dependent encryption. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 202–219. Springer (Mar 2009)
22. Haitner, I., Rosen, A., Shaltiel, R.: On the (im)possibility of Arthur-Merlin witness hiding protocols. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 220–237. Springer (Mar 2009)

23. Hartmanis, J., Stearns, R.E.: On the computational complexity of algorithms. Transactions of the American Mathematical Society 117, 285–306 (1965)
24. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999)
25. Hsiao, C.Y., Reyzin, L.: Finding collisions on a public road, or do secure hash functions need secret coins? In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 92–105. Springer (Aug 2004)
26. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: 21st ACM STOC. pp. 44–61. ACM Press (May 1989)
27. Kiltz, E., Pietrzak, K.: On the security of padding-based encryption schemes - or - why we cannot prove OAEP secure in the standard model. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 389–406. Springer (Apr 2009)
28. Lindell, Y., Omri, E., Zarosim, H.: Completeness for symmetric two-party functionalities - revisited. In: Wang, X., Sako, K. (eds.) Advances in Cryptology — Asiacrypt 2012. Lecture Notes in Computer Science, vol. 7658, pp. 116–133. Springer-Verlag (2012)
29. Mahmoody, M., Pass, R.: The curious case of non-interactive commitments - on the power of black-box vs. non-black-box use of primitives. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 701–718. Springer (Aug 2012)
30. Paillier, P., Villar, J.L.: Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 252–266. Springer (Dec 2006)
31. Pass, R.: Limits of provable security from standard assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 109–118. ACM Press (Jun 2011)
32. Pass, R., Tseng, W.L.D., Venkitasubramaniam, M.: Towards non-black-box lower bounds in cryptography. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 579–596. Springer (Mar 2011)
33. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of reducibility between cryptographic primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer (Feb 2004)
34. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: 22nd ACM STOC. pp. 387–394. ACM Press (May 1990)
35. Seurin, Y.: On the exact security of schnorr-type signatures in the random oracle model. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 554–571. Springer (Apr 2012)
36. Simon, D.R.: Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In: Nyberg, K. (ed.) EUROCRYPT'98. LNCS, vol. 1403, pp. 334–345. Springer (May / Jun 1998)
37. Yao, A.C.: Theory and applications of trapdoor functions. In: 23rd FOCS. pp. 80–91. IEEE Computer Society Press (Nov 1982)