Notions of Black-Box Reductions, Revisited

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

[1] Darmstadt University of Technology, Germany
[2] Tel-Aviv University, Israel

November 29, 2013

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.

Contents

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 [HILL99, GGM86, Rom90]. For other primitives, however, there are results separating primitives like key agreement or collision-resistant hash functions from one-way functions [IR89, Sim98].

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. [RTV04] 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. [RTV04] 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 [IR89] 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 [RTV04].

Later, Hsiao and Reyzin [HR04] 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 [IR89]). Because this makes separations often much more elegant this technique has been applied successfully for many other primitives, e.g., [DOP05, HHRS07, KP09, HH09, BCFW09, FLR$^+$10, MP12, LOZ12, BH13].

Interestingly, recently there has been another type of separations based on so-called meta-reduction techniques, originally introduced by Boneh and Venkatenesan [BV98], and subsequently

used in many other places [Cor02, PV06, HRS09, FS10, Pas11, GW11, DHT12, Seu12, FF13]. 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 [RTV04].

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 [BKSY11]). We also remark that in some cases, such as Barak's ingenious result about non-black-box zero-knowledge and related results [Bar01, BP12], 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. [PTV11]. 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 [MP12] 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 [GL89], attributed to Rackoff in [Gol04]. 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 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 [GL89], must depend on the adversary's success probability (and thus needs to be parametrized).

2 Notions of Reducibility

We extend the original framework for notions of reducibility by Reingold, Trevisan and Vadhan [RTV04]. 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 $\forall\exists$ 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 $\forall\exists$ 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

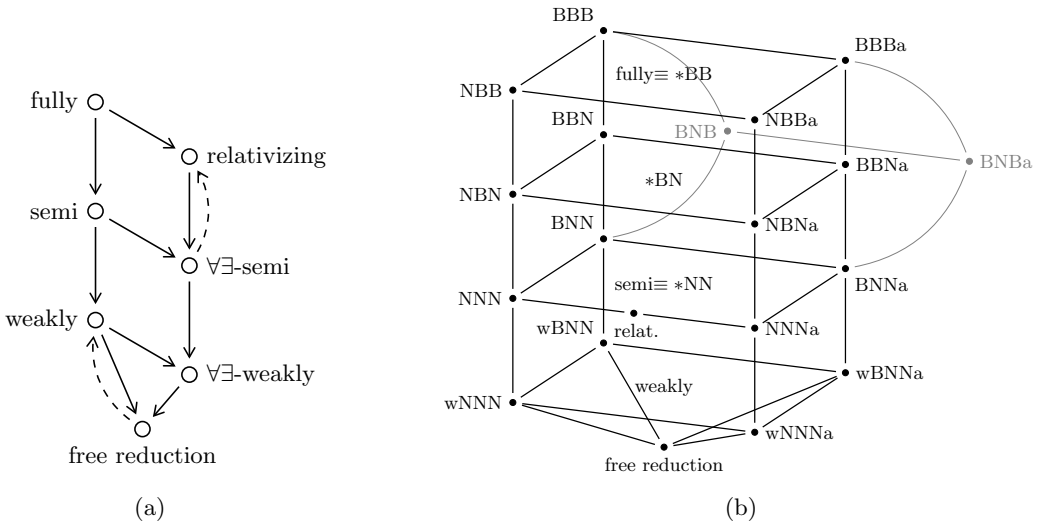

Figure 1: (a) shows the relation of notions in the RTV framework. The dashed arrows indicate equivalence for a restricted class of reductions. In our framework (b), it is instructive to look at the vertical planes for fully, ∗BN, semi, and weakly. The left side corresponds to inefficient adversaries, 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.

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 $\mathcal{P}$.

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.

CAP	[RTV04] name	Remark(s)
BBB	fully	known meta reductions: [BMV08, HRS09]
BBN		
BNB		known reduction: [GL89]
BNN	semi (weakly)	
NBB	∀∃-fully	formally not defined in [RTV04], only "trivial" reductions
NBN		known meta reductions: [BV98, HRS09, FS10, Pas11]
NNB		not meaningful
NNN	∀∃-semi (∀∃-weakly)	

Figure 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.

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 [RTV04] 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 ∀∃ 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 Appendix C.

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 $(f, \mathcal{A}) \in \mathcal{R}_\mathcal{Q}$. Following [RTV04], we say that a primitive exists if there is a polynomial-

time computable instantiation $f \in \mathcal{F}_\mathcal{Q}$ such that no polynomial-time random variable breaks the primitive. Indeed, [RTV04] demand that primitive sets $\mathcal{F}_\mathcal{Q}$ 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 Appendix A.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 Appendix A.2 we discuss another example of weak one-way functions (and the construction of strong one-way functions [Yao82]) 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 [RTV04]. 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. [RTV04] indeed define the relations for adversarial *machines*. Our discussion in Appendix B shows that only interpreting such adversaries as abstract objects sustains the implications in [RTV04]. 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 Appendix B 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 2.1 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-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*

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

Figure 3: Overview of notions of reducibility.

$(f,\mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, *i.e.,*

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

Definition 2.2 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-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.,*

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

Definition 2.3 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-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.,*

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

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 Appendix C.

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. [RTV04] 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 2.4 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-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.,

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

Again, the definitions for the remaining types of reductions are presented in Appendix C.

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 2.5 *Let XYZ and $\widehat{X}\widehat{Y}\widehat{Z}$ be two types of CAP reductions such that $\widehat{X}\widehat{Y}\widehat{Z} \leq XYZ$ pointwise (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 Appendix D, 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 [GL89] has to depend on the success probability of the adversary (Theorem D.2). Moreover, we show that the construction of one-way functions out of weak one-way functions ([Yao82, GIL$^+$90]) needs to depend on the weakness parameter of the weak one-way function (Theorem D.3). Interestingly, some of the implications of Theorem 2.5 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 2.6 (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.*

The proof of Theorem 2.6 is deferred to Appendix D. 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.

Theorem 2.7 *For each* $XYZ \in \{BBB, BNB, BBN, NBB, BNN, NBN, NNN\}$*, there are primitives* $\mathcal{P}$ *and* $\mathcal{Q}$ *such that there is a* $(\mathcal{P} \hookrightarrow \mathcal{Q})$*-XYZa-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 [HS65]. 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 $XYZ \in \{\mathrm{BBB}, \mathrm{BNB}, \mathrm{BBN}, \mathrm{NBB}, \mathrm{BNN}, \mathrm{NBN}, \mathrm{NNN}\}$, 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 $XYZ \in \{\mathrm{BBB}, \mathrm{BNB}, \mathrm{BBN}, \mathrm{NBB}, \mathrm{BNN}, \mathrm{NBN}, \mathrm{NNN}\}$, 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 2.8 (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. [RTV04], 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. The proof is inspired by Reingold et al. [RTV04] who show that BBB-reductions imply relativizing reductions.

Theorem 2.9 *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. [RTV04] for the following theorem.

Theorem 2.10 *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 2.6 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 2.11 *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 2.7.

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. [RTV04] 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 2.12 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-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.,*

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

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*. For example, the notion of a BBBap-reduction is as follows:

Definition 2.13 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBBap Reduction) *There exists a* fully black-box (or BBBap-)reduction *for efficient adversaries and 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 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.,*

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

We will now review the separations proved in this paper. Basically, all relations that hold for XYZ-reductions and XYZa-reductions, also hold for XYZp and XYZap reductions, except for the relation to relativizing reductions as we will see below in Theorem 2.15. Firstly, the proof of Theorem 2.6, showing equivalence of black-box access and non-black-box access to the efficient adversary, works for all classes of primitives and in particular for efficiently implementable ones. We conclude that there is an NNNap-reduction from $\mathcal{P}$ to $\mathcal{Q}$ if and only if there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNNap-reduction. For Theorem 2.7, separating reductions for the cases of efficient resp. inefficient adversaries, we observe that the primitive there, the constant 0-function, is efficiently implementable. The proof hence also shows that for all $XYZ \in \{\mathrm{BBB}, \mathrm{BNB}, \mathrm{BBN}, \mathrm{NBB}, \mathrm{BNN}, \mathrm{NBN}, \mathrm{NNN}\}$, there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$ap-reduction, but no $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$p-reduction. Note that this observation neither separates XYZ-reductions from XYZp-reductions, nor does it separate XYZa-reductions from XYZap-reduction. These two classes of separations will be taken care of by Theorem 2.14.

Similarly to the constant 0-function case, all results that rely on random oracles carry through, as random oracles are efficiently computable. That is, Theorem D.2, that uses a special class of weak one-way functions implemented as one-way oracles, still holds and shows that there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is an $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNBp-reduction, but no BBBp-reduction. The same theorem establishes that for the same two primitives, there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNNp-reduction, but no BBNp-reduction, and there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNNp-reduction, but no NBNp-reduction. Moreover, as Theorem D.3 also relies on random oracles only, we conclude that there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that for all $YZ \in \{\mathrm{BB}, \mathrm{BBa}, \mathrm{NN}, \mathrm{NNa}\}$, there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-N$YZ$p-reduction, but there is neither a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-B$YZ$p-reduction, nor a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBNp-reduction, nor a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBNap-reduction. Finally, Theorem D.4 only uses the efficiently implementable constant 0-function, and thus, the

proof of Theorem D.4 also establishes that for $X \in \{\mathrm{N}, \mathrm{B}\}$ there exist primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$X$BNp-reduction, a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$X$BNap-reduction, a BNNp-reduction and BNNap-reduction but such that there is no $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$X$BBp/$X$BBap/BNBp/BNBap-reduction.

We now prove an analogue to Theorem 2.7, to separate reductions for arbitrary reductions from reductions for efficient primitives.

Theorem 2.14 *For each $XYZ \in \{BBB, BNB, BBN, NBB, BNN, NBN, NNN\}$, there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$p-reduction, a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$ap-reduction, but neither a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$a-reduction, nor a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$-reduction.*

Proof. In the proof of Theorem 2.7 we make the primitive $\mathcal{Q}$ unbreakable. Thus, a reduction can only exist if one of the universal quantifiers $\forall\mathcal{A}$ or $\forall f$ quantifies over the empty set. We can use the same technique here, but this time we swap the role of the adversary and the role of the function f. Let $\mathcal{L}$ be an $\mathsf{EXPTIME}$-complete language, and let f be the characteristic function for $\mathcal{L}$. Now, let $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ be defined through the singleton function set $\mathcal{F}_\mathcal{Q} = \{f\}$ and the empty relation $\mathcal{R}_\mathcal{Q} = \emptyset$. Let $\mathcal{P}$ be the primitive where $\mathcal{F}_\mathcal{P}$ only contains the constant 0-function, denoted by f_0, and where all PPT adversaries $\mathcal{A}$ break this function 0, i.e., $\mathcal{R}_\mathcal{P} := \{(f_0, \mathcal{A}) | \mathcal{A} \text{ is } \mathsf{PPT}\}$. Then, let G be the construction that ignores its oracle and constantly returns 0. Thus, for any $f \in \mathcal{F}_\mathcal{Q}$ the construction G^f implements the 0-function.

Now, any universal PPT reduction algorithm $\mathcal{S}$ implements a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$p-reduction, a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$ap-reduction, as the quantifier $\forall\mathsf{PPT}f \in \mathcal{F}_\mathcal{Q}$ quantifies over the empty set. On the other hand, no pair of a construction G and a reduction algorithm $\mathcal{S}$ (depending on f and $\mathcal{A}$ or not) will implement a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$a-reduction or a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$reduction. This is because the premise of the implication

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

can be easily satisfied, as G^f implements the all-zero function and any PPT adversary $\mathcal{A}$ breaks it. On the other hand, the conclusion is impossible to achieve since $\mathcal{R}_\mathcal{Q}$ is empty.

As mentioned before, the original RTV paper required that $\mathcal{F}_\mathcal{P}$ contain at least one efficiently computable primitive. Thus, we have to slightly adapt our proof to work also in their setting. To do so, we add the constant 0-function f_0 to $\mathcal{F}_\mathcal{Q}$, i.e., $\mathcal{F}_\mathcal{Q} := \{f, f_0\}$ and define $\mathcal{R}_\mathcal{Q} := \mathcal{R}_\mathcal{P}$, i.e., for f, there is still no adversary that breaks f. Then, the same analysis applies. □

Note that the proof of Theorem 2.7 also shows the stronger statement that there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$p-reduction, a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$-reduction, but neither a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$a-reduction, nor a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$ap-reduction. This is, because all considered primitives in the proof of Theorem 2.7 are efficiently computable, namely the constant 0-function.

As in the case of efficient adversaries, XYZp-reductions are not strong enough to imply relativizing reductions.

Theorem 2.15 *There exists primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is an $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$p-reduction, but no relativizing reduction.*

Proof. Consider the primitives $\mathcal{P}$ and $\mathcal{Q}$ designed in the previous proof. We saw that there is an $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$XYZ$p-reduction. We will show that there is no relativizing reduction from $\mathcal{P}$ to $\mathcal{Q}$ by proving that, relative to an oracle Π that implements the characteristic function of an $\mathsf{EXPTIME}$-complete problem, the primitive $\mathcal{Q}$ exists, while the primitive $\mathcal{P}$ does not. First note that f is efficiently computable relative to an $\mathsf{EXPTIME}$-complete oracle. Moreover, by definition of $\mathcal{R}_\mathcal{Q}$,

there is no adversary $\mathcal{A}$ such that $(f, \mathcal{A})$ is in $\mathcal{R}_{\mathcal{Q}}$ and thus, f is efficiently computable relative to the EXPTIME-complete oracle and cannot be broken even by an adversary that is given access to the oracle. We showed that f exists relative to Π. On the other hand, $\mathcal{Q}$ does not exist relative to any oracle as the constant 0-adversary $\mathcal{A}$ always breaks all implementations in $\mathcal{P}_{\mathcal{Q}}$, as $\mathcal{R}_{\mathcal{P}} = \{(0, \mathcal{A}) | \mathcal{A} \text{ is PPT}\}$. □

Theorem 2.10 shows that relativizing reductions imply NNNa-reductions and thus, they also imply NNNap-reductions. However, it is not clear whether they equally imply NNNp-reductions. Theorem 2.10 considers a (possibly inefficient) implementation f of $\mathcal{F}_{\mathcal{Q}}$ as an oracle and argues that, relative to this oracle, the primitive $\mathcal{F}_{\mathcal{P}}$ exists, i.e., there is an efficient oracle algorithm G such that G^f implements $\mathcal{P}$ and cannot be broken by an adversary that has access to f. When switching the roles of the function f and the adversary $\mathcal{A}$, then this argument does not carry over, as the construction G does not get access to the adversary $\mathcal{A}$.

Note that the separation in Theorem 2.15 tells us that the use of efficient primitives is a possible way to bypass the important class of oracle separations with inefficient oracles. Nevertheless, it might also be interesting to explore the converse direction, i.e., whether relativizing reductions imply any type of XYZp-reduction or not.

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). Before going into the details we first consider some motivating examples of dependencies on parameters of the adversary.

3.1 Parameter-Aware and Parameter-Dependent Reductions

Let us reduce unforgeability of a MAC scheme to its own unforgeability, i.e., the reduction algorithm $\mathcal{R}$ merely relays queries and answers between the unforgeability game and the adversary $\mathcal{A}$. Although the reduction algorithm is trivial, its running time depends on the adversary's behavior. Namely, the running time of the reduction is polynomial in the security parameter n and qry, the number of queries placed by the adversary. Hence the running time of $\mathcal{S}$ actually depends on $\mathcal{A}$, while the code of the strictly polynomial-time algorithm $\mathcal{S}$ should be universal for all $\mathcal{A}$, thus allowing only an a-priori limited number of interactions with the adversary.

Another example is the well-known Goldreich–Levin hardcore bit reduction [GL89], in the version attributed to Rackoff [Gol04]. Recall that the reduction algorithm receives some input $f(x)$ and has access to an adversary that predicts a hardcore bit with some non-trivial advantage $\epsilon(n)$. The reduction then uses amplification techniques by asking the adversary on many different input strings and thereby yields a pre-image of $f(x)$ with non-negligible probability. As the amplification step heavily depends on $\epsilon(n)$, the reduction is not universal anymore; it changes with different values of $\epsilon(n)$. Moreover, the running time of the reduction depends on $\frac{1}{\epsilon(n)}$ or, more precisely, on

the polynomial $p(n)$ with $\frac{1}{p(n)} > \frac{1}{\epsilon(n)}$. Other than that, the reduction treats both the adversary and the primitive as black-boxes.

The MAC example above shows that we sometimes want to allow the reduction, especially its running time, to depend on adversarial parameters such as its number of queries. In the second example the reduction needs (one of) the parameters as explicit input. We call the latter (black-box) reductions *parameter-aware*, and the former *parameter-dependent*. In fact, we make parameter-aware reductions strictly stronger by also making the reduction's running time depend on the input parameters.

The difference between the two notions is roughly that, in the parameter-aware case the reduction receives some auxiliary information about the adversary which may not be even known by the adversary itself (like the success probability), akin to non-uniform advice. In the parameter-dependent case the reduction only has sufficient time to run the adversary without violating prematurely fixed bounds on the running time. As another example one may consider knowledge extractors in proofs of knowledge [BG93] which run in expected polynomial time related to the prover's success probability to convince the verifier. This knowledge extractor can be oblivious about the actual success bound, and yet the running time depends on it. Remarkably, in order to prune the expected polynomial time by standard techniques to make the algorithm run in strict polynomial time, one needs to know the success probability and obtains a parameter-aware algorithm.

To simplify we only define the two cases for BBB-reductions and refrain from distinguishing between different parameters (for inputs resp. for running time dependency). We formalize this by having a function par mapping adversaries $\mathcal{A}$ to a function $\mathsf{par}_{\mathcal{A}}$ which, in turn, maps the security parameter to the desired parameters like the number of queries (to simplify, we include again the security parameter in these parameters). As such, when saying below that the reduction runs in polynomial time in $\mathsf{par}_{\mathcal{A}}$ it should be understood as saying that the running time of the reduction is polynomial in the output length of $\mathsf{par}_{\mathcal{A}}$ for the security parameter. This usually assumes some unitary encoding of the parameters. In the parameter-aware case we simply give the reduction the transformed input $\mathsf{par}_{\mathcal{A}}$ instead and write $\mathcal{S}(\mathsf{par}_{\mathcal{A}})$.

Definition 3.1 (Parameter-aware and parameter-dependent BBB-Reduction) *There exists a* parameter-aware BBB-reduction *from a primitive $\mathcal{P} = (\mathcal{F}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$ to a primitive $\mathcal{Q} = (\mathcal{F}_{\mathcal{Q}}, \mathcal{R}_{\mathcal{Q}})$ with respect to par, 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 machine $\mathcal{A}$, if $(G^f, \mathcal{A}) \in \mathcal{R}_{\mathcal{P}}$, then we have $(f, \mathcal{S}^{\mathcal{A},f}(\mathsf{par}_{\mathcal{A}}) \in \mathcal{R}_{\mathcal{Q}})$, i.e.,*

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

where algorithm $\mathcal{S}$ runs in polynomial-time in its input and $\mathsf{par}_{\mathcal{A}}$. There is a parameter-dependent BBB-reduction *if the above holds if input $\mathsf{par}_{\mathcal{A}}$ is not given to $\mathcal{S}$ (but the running time may still depend on $\mathsf{par}_{\mathcal{A}}$).*

Although we state parametrized reductions in a rather general way, various standard choices for par are conceivable. In the light of the examples above, reasonable choices could be parameter functions that map descriptions of adversaries and the security parameter to the number of queries they make ($\mathsf{par}_{\mathsf{q}}$), to (the inverse of) their success probability (par_{ϵ}), or to their running time ($\mathsf{par}_{\mathsf{t}}$). This usually requires a refinement of the formalization of primitives to some form of games, e.g., to

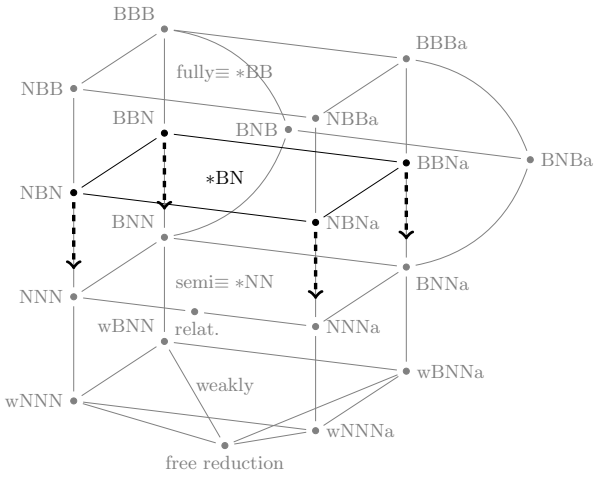

Figure 4: The effect of parametrization (in the case of ∗BN-reductions). Parametrized counterparts of each type partly descend towards the corresponding ∗NN-reduction with full dependency on the adversary.

be able to specify the number of queries of the adversary. Consequently, this allows us to capture many known black-box reductions in the literature as BBB-reductions with explicit parameters. At the same time, however, we get a very strict notion of a black-box reduction by letting $\mathsf{par}_{\mathcal{A}} = \bot$. In fact, this recovers Definition 2.1 and corresponds to our view on black-box reductions so far. We note that our definition leaves open whether the adversary actually knows $\mathsf{par}_{\mathcal{A}}$ itself or not.

Finally, let us stress that we could also "parametrize" the other black-box objects, i.e., give the construction some hint about the black-box primitive such as its computation time, or hand the reduction further information about the primitive's parameters. We refrain from doing this formally as it is straightforward from the definition above and since, unlike in the case of adversarial parameters, we are not aware of "natural" examples for such cases.

3.2 Relationships

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 (see Theorem D.2). 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.

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 that one descends 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.

3.3 Parameter Awareness and Parameter Dependency

Concerning the relationship of the two types of parametrized reductions we note that parameter-awareness is not more powerful than parameter-dependent reductions, in the case that one can compute $\mathsf{par}_{\mathcal{A}}$ in time depending on the parameters. For sake of concreteness, we discuss this for the number of adversarial queries, showing that the two notions are equivalent in this case, except for the cases where the reduction cannot depend on the construction. This equivalence, of course, only makes sense for security relations in which there is a game between the adversary and the primitive in which the adversary actually poses queries. We call a relation $\mathcal{R}_{\mathcal{P}}$ *challenger-based* if there is an efficient algorithm $C^{\mathcal{A},G^f}$ and a relation $\mathcal{R}^C_{\mathcal{P}}$ such that $(G^f,\mathcal{A}) \in \mathcal{R}_{\mathcal{P}}$ iff $(G^f, C^{\mathcal{A},G^f}) \in \mathcal{R}^C_{\mathcal{P}}$. In this case we denote by $\mathsf{par}_{\mathsf{q}}$ the number of oracle calls of C to $\mathcal{A}$ (as a function of the security parameter). Note that this number now includes the first invocation of the adversary, but can essentially be thought of the number of queries $\mathcal{A}$ poses to the security game. We also assume that C simply makes additional queries if the adversary stops early, such that the number of oracle calls remains identical for all runs.

Proposition 3.2 *Let $\mathcal{P} = (\mathcal{F}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$ be a primitive with a challenger-based relation $\mathcal{R}_{\mathcal{P}}$. For any parameter-aware $(\mathcal{P} \hookrightarrow \mathcal{Q})$-XYZ-reduction (of type XYZ $\notin \{NNB, NBB\}$) with respect to $\mathsf{par}_{\mathsf{q}}$ there is also parameter-dependent $(\mathcal{P} \hookrightarrow \mathcal{Q})$-XYZ-reduction of the same type XYZ. (This holds also for XYZa reductions.)*

Proof. Consider a parameter-aware reduction $\mathcal{S}$ of some admissible type, and let the C be the algorithm implementing the guaranteed challenger game for $\mathcal{R}_{\mathcal{P}}$. Then we build a parameter-dependent reduction $\mathcal{S}'$ as follows. Reduction $\mathcal{S}'$ first runs C with the adversary oracle once against G^f, simulating oracle G^f for C with the help of (possibly black-box access to) f. It counts the number of queries $\mathsf{par}_{\mathsf{q}}$ the challenger (resp. the adversary) makes. Note that $\mathcal{S}'$ only needs $\mathcal{O}(\mathsf{par}_{\mathsf{q}})$ steps for simulating the black-box adversary oracle, even if given $\mathcal{A}$ as black-box. Also, $\mathcal{S}'$ requires for the simulation to be able to depend on the running time of G^f, which it indeed does for the admissible reduction types. Finally, once $\mathcal{S}'$ has received $\mathsf{par}_{\mathsf{q}}$ it can simply invoke $\mathcal{S}$ for additional input $\mathsf{par}_{\mathsf{q}}$. □

It is now conceivable that, if one cannot compute $\mathsf{par}_{\mathcal{A}}$ (given that one can run in time depending on $\mathsf{par}_{\mathcal{A}}$), then parameter-aware reductions should be more powerful. This, however, presumes some notion of unpredictability which, in turn, stipulates some form of verifiability of correct outputs. As mentioned before, even the adversary itself may not be able to verify such outputs, e.g., think of its own success probability. This adds an additional layer of dependency of parameters, which is beyond our scope here.

4 Meta-Reductions

In this section, we define meta-reductions within our augmented RTV framework and prove that if there exists a meta-reduction from a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-reduction to the primitive $\mathcal{Q}$, then there is no

reduction from $\mathcal{P}$ to $\mathcal{Q}$, provided that $\mathcal{Q}$ exists. More generally, if there exists a meta-reduction from a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-reduction to a primitive $\mathcal{N}$, then there is no reduction from $\mathcal{P}$ to $\mathcal{Q}$, provided that $\mathcal{N}$ exists, i.e., there is an efficient implementation of $\mathcal{N}$ such that no efficient adversary breaks it. We now rephrase meta-reductions for all notions introduced in the previous section. Below we usually put the statement in terms of reductions and meta-reduction of the same CAP type. It is clear that a meta-reduction of the XYZ type, ruling out reductions of the XYZ type, also exclude all higher-level (i.e., "more black-box") reductions of type $\widehat{X}\widehat{Y}\widehat{Z} \geq XYZ$. Such higher-level reductions imply a reduction of the type $XYZ$and would thus contradict the impossibility result.

4.1 Meta-Reductions for BYZ-Reductions

Definition 4.1 ($(\mathcal{P} \hookrightarrow \mathcal{Q}$-$BBB) \hookrightarrow \mathcal{N}$ (aka. Fully Black-Box-)Meta-Reduction) *For primitives $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{N}$, a probabilistic polynomial-time algorithm $\mathcal{M}$ is a $(\mathcal{P} \hookrightarrow \mathcal{Q}$-$BBB) \hookrightarrow \mathcal{N}$-*meta-reduction *from a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBB-reduction to $\mathcal{N}$, if the following holds for all $g \in \mathcal{F}_{\mathcal{N}}$:*

Reduction implies Insecurity. *If $\mathcal{P}$ BBB-reduces to $\mathcal{Q}$ via a construction G and a reduction algorithm $\mathcal{S}$, then there is a* PPT *$\mathcal{M}$ such that one has $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$.*

$$\forall g \in \mathcal{F}_{\mathcal{N}}\ \forall \mathsf{PPT} G\ \forall \mathsf{PPT} \mathcal{S}\ \exists f \in \mathcal{F}_{\mathcal{Q}}\ \exists \mathcal{A}\ \exists \mathsf{PPT} \mathcal{M}$$

$$\Big[\big((G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}\big) \tag{1}$$

$$\Rightarrow (g, \mathcal{M}^g) \in \mathcal{R}_N\Big] \tag{2}$$

To construct a $(\mathcal{P} \hookrightarrow \mathcal{Q}$-$BBB) \hookrightarrow \mathcal{Q}$-meta-reduction, one usually instantiates f via g and picks an (possibly inefficient) adversary $\mathcal{A}$ that breaks G^f. The efficient reduction algorithm $\mathcal{S}$ will turn $\mathcal{A}$ into a successful adversary against $f = g$. Thus, the meta-reduction $\mathcal{M}$ aims at simulating $\mathcal{A}$ *efficiently* for $\mathcal{S}$. For this purpose, the meta-reductions rewinds the reduction, e.g., to extract a signature from the reduction that simulates a signing oracle—the extracted signature can then be presented as a genuine fresh signature to a rewound version of the reduction $\mathcal{S}$.[1] Consider the order of quantifiers in the above definition: the meta-reduction may use non-black-box information about g and $\mathcal{S}$ such as the running time of $\mathcal{S}$ or its success probability. This definition is as liberal as possible on the meta-reduction while preserving its ultimate goal: if a reduction $(G, \mathcal{S})$ exists and a meta-reduction, then clearly, the primitive $\mathcal{Q}$ cannot exist.

Theorem 4.2 *If $\mathcal{N}$ exists and if there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}$-$BBB) \hookrightarrow \mathcal{N}$-(aka. fully black-box-)meta-reduction, then there is no $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBB-reduction.*

Corollary 4.3 *If there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}$-$BBB) \hookrightarrow \mathcal{Q}$(aka. fully black-box -)meta-reduction, then a secure instantiation of $\mathcal{P}$ cannot be based on the existence of $\mathcal{Q}$ via a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBB-reduction.*

Proof (of Theorem 4.2). Assume that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-reduction and a $(\mathcal{P} \hookrightarrow \mathcal{Q}$-$BBB) \hookrightarrow \mathcal{N}$-meta-reduction. To derive a contradiction, we show that $\mathcal{N}$ cannot exist. Let $g \in \mathcal{F}_{\mathcal{N}}$ be arbitrary but fixed. As there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBB-reduction, let $(G, \mathcal{S})$ be a pair of a probabilistic polynomial-time construction G and a probabilistic polynomial-time reduction $\mathcal{S}$ such that for all f and $\mathcal{A}$, if $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$. Moreover, for $(G, \mathcal{S})$, let $\mathcal{M}$ be the meta-reduction together with the corresponding f and $\mathcal{A}$ granted by the meta-reduction property. As the condition $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$ is satisfied for all f and $\mathcal{A}$, we have that the meta-reduction

[1] We hide the details under the rug; indeed, the actual analysis is usually more complicated.

$\mathcal{M}$ breaks g, formally $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$. Note that $\mathcal{M}^g$ is efficiently computable if and only if g is efficiently computable. As the analysis holds for all $g \in \mathcal{F}_{\mathcal{N}}$, we derive that all efficiently computable instantiations of $\mathcal{N}$ can be broken by an efficient adversary. □

Note that all introduced notions for meta-reductions easily translate into meta-reductions for efficient adversaries by only quantifying over efficient $\mathcal{A}$.

We complete the picture regarding the remaining types of meta-reductions in Appendix E.

4.2 Examples of Meta-Reductions

Meta-reductions have been used for the first time (albeit not explicitly under this name) in the work of Boneh and Venkatesan [BV98] to study the relation between breaking RSA and factoring. Their result says that there is no (straight line respectively algebraic) reduction from breaking RSA to factoring since such a reduction would immediately yield an efficient algorithm for factoring. Since they consider concrete problem instantiations, neither of the primitives is black-box. In order to look at this result in terms of meta-reductions, it is instructive to view the RSA oracle as the adversary and the reduction as a generic straight line program evaluation machine for the actual reduction's output that handles embedded RSA oracle calls within the program by forwarding them to the adversary. This makes the adversarial access black-box and results in a NBN type meta-reduction.

Bresson et al. [BMV08] discuss separations amongst so-called one-more problems where an adversary may query an oracle for solutions on n instances but needs to provide eventually $n + 1$ instance/solution pairs in order to be successful. The results indicate that solving such problems on n instances does not reduce to the case of solving the same problem on $n - 1$ instances (using fresh randomness). Again, the adversary and the primitive for the n-instance problem is treated in a black-box manner by the reduction. One may argue that the construction is black-box as well since the problem can be constructed for an arbitrary number of instances solely by given access to oracles for generating and verifying one instance. Hence, this is an example for a BBB-meta-reduction. We note that all the reductions in this work come with certain restrictions though and *meta*-reductions appear both as black-box and non-black-box—in the case of algebraic reductions—flavors.

The work of Haitner et al. [HRS09] uses meta-reductions to show that witness-hiding of certain proof systems cannot be based on either a specific hardness assumption or, separately, on any implementation of a primitive. These two variants precisely reflect the difference how the primitive is treated within the reduction and the construction. The latter case indicates a BBB-meta-reduction. For specific assumptions, the reduction may depend on the primitive and the authors call this a *weakly*-black-box reduction which shall not be confused with the weakly terminology of [RTV04]. In our framework this type of reduction classifies as a NBN-meta-reduction.

[FS10] Fischlin and Schröder [FS10] prove the impossibility of basing blind signatures on a non-interactive standard assumption using a meta-reduction. Here, the construction may be non-black-box, the adversary is treated as a black-box, but the reduction is not restricted to black-box access to the primitive. This classifies as a NBN-meta-reduction.

Finally, the recent work by Pass [Pas11] presents a powerful framework to show that a certain type of argument system cannot be based on certain standard assumptions. By restating several interesting constructions as an argument system, it follows that these constructions cannot be based on standard assumptions either. More specifically, these constructions include the Schnorr identification scheme, the adaptive selective decommitment problem, one-more inversion assumptions, and unique blind signatures (generalizing the aforementioned result). Again, the underlying technique of this framework is a meta-reduction. These results hold whenever the adversary in the

reduction is treated as a black-box but allows arbitrary constructions, which, in particular, may be non-black-box. Since the reduction may depend on the standard assumptions as well, this type of meta-reduction is considered a NBN-meta-reduction in our terminology.

5 Conclusion

We provide a comprehensive framework which can be used 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.

Acknowledgments

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

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd FOCS*, pages 106–115. IEEE Computer Society Press, October 2001. (Cited on page 2.)

[BCFW09] Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi. Foundations of non-malleable hash and one-way functions. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 524–541. Springer, December 2009. (Cited on page 1.)

[BG93] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 390–420. Springer, August 1993. (Cited on page 15.)

[BH13] Kfir Barhum and Thomas Holenstein. A cookbook for black-box separations and a recipe for UOWHFs. In *TCC 2013*, LNCS. Springer, 2013. (Cited on page 1.)

[BKSY11] Zvika Brakerski, Jonathan Katz, Gil Segev, and Arkady Yerukhimovich. Limits on the power of zero-knowledge proofs in cryptographic constructions. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 559–578. Springer, March 2011. (Cited on page 2.)

[BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the "one-more" computational problems. In Tal Malkin, editor, *CT-RSA 2008*, volume 4964 of *LNCS*, pages 71–87. Springer, April 2008. (Cited on pages 5 and 19.)

[BP12] Nir Bitansky and Omer Paneth. 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*, pages 223–232. IEEE Computer Society Press, 2012. (Cited on page 2.)

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 59–71. Springer, May / June 1998. (Cited on pages 1, 5, and 19.)

[Cor02] Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 272–287. Springer, April / May 2002. (Cited on page 2.)

[DHT12] Yevgeniy Dodis, Iftach Haitner, and Aris Tentes. On the instantiability of hash-and-sign RSA signatures. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 112–132. Springer, March 2012. (Cited on page 2.)

[DOP05] Yevgeniy Dodis, Roberto Oliveira, and Krzysztof Pietrzak. On the generic insecurity of the full domain hash. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 449–466. Springer, August 2005. (Cited on page 1.)

[FF13] Marc Fischlin and Nils Fleischhacker. Limitations of the meta-reduction technique: The case of schnorr signatures. In *EUROCRYPT 2013*, LNCS. Springer, 2013. (Cited on page 2.)

[FLR+10] Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 303–320. Springer, December 2010. (Cited on page 1.)

[FS10] Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 197–215. Springer, May 2010. (Cited on pages 2, 5, 19, and 34.)

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986. (Cited on page 1.)

[GIL+90] Oded Goldreich, Russell Impagliazzo, Leonid A. Levin, Ramarathnam Venkatesan, and David Zuckerman. Security preserving amplification of hardness. In *FOCS*, pages 318–326. IEEE Computer Society, 1990. (Cited on page 9.)

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989. (Cited on pages 3, 5, 9, 14, 29, and 30.)

[Gol04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004. (Cited on pages 3 and 14.)

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011. (Cited on pages 2 and 24.)

[HH09] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 202–219. Springer, March 2009. (Cited on page 1.)

[HHRS07] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *48th FOCS*, pages 669–679. IEEE Computer Society Press, October 2007. (Cited on page 1.)

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. (Cited on page 1.)

[HR04] Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 92–105. Springer, August 2004. (Cited on page 1.)

[HRS09] Iftach Haitner, Alon Rosen, and Ronen Shaltiel. On the (im)possibility of Arthur-Merlin witness hiding protocols. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 220–237. Springer, March 2009. (Cited on pages 2, 5, and 19.)

[HS65] Juris Hartmanis and Richard Edwin Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965. (Cited on page 9.)

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989. (Cited on page 1.)

[KP09] Eike Kiltz and Krzysztof Pietrzak. On the security of padding-based encryption schemes - or - why we cannot prove OAEP secure in the standard model. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 389–406. Springer, April 2009. (Cited on page 1.)

[LOZ12] Yehuda Lindell, Eran Omri, and Hila Zarosim. Completeness for symmetric two-party functionalities - revisited. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology — Asiacrypt 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 116–133. Springer-Verlag, 2012. (Cited on page 1.)

[LTW05] Henry Lin, Luca Trevisan, and Hoeteck Wee. On hardness amplification of one-way functions. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 34–49. Springer, February 2005. (Cited on pages 31 and 32.)

[Mau02] Ueli M. Maurer. Indistinguishability of random systems. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. Springer, April / May 2002. (Cited on page 25.)

[MP12] Mohammad Mahmoody and Rafael Pass. The curious case of non-interactive commitments - on the power of black-box vs. non-black-box use of primitives. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 701–718. Springer, August 2012. (Cited on pages 1 and 2.)

[Pas11] Rafael Pass. Limits of provable security from standard assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 109–118. ACM Press, June 2011. (Cited on pages 2, 5, and 19.)

[PTV11] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Towards non-black-box lower bounds in cryptography. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 579–596. Springer, March 2011. (Cited on pages 2 and 36.)

[PV06] Pascal Paillier and Jorge L. Villar. Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 252–266. Springer, December 2006. (Cited on page 2.)

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990. (Cited on page 1.)

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 1–20. Springer, February 2004. (Cited on pages 1, 2, 3, 5, 6, 8, 10, 11, 19, 25, 26, 29, and 36.)

[Seu12] Yannick Seurin. On the exact security of schnorr-type signatures in the random oracle model. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 554–571. Springer, April 2012. (Cited on page 2.)

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 334–345. Springer, May / June 1998. (Cited on page 1.)

[TY98] Yiannis Tsiounis and Moti Yung. On the security of ElGamal based encryption. In Hideki Imai and Yuliang Zheng, editors, *PKC'98*, volume 1431 of *LNCS*, pages 117–134. Springer, February 1998. (Cited on page 23.)

[Yao82] Andrew C. Yao. Theory and applications of trapdoor functions. In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, November 1982. (Cited on pages 6, 9, 25, and 31.)

A Examples

A.1 An Example: Reduction from ElGamal encryption to the DDH Assumption

In this section we present the reduction from the indistinguishability of ElGamal encryption to the DDH problem. It is well known [TY98] that the DDH assumption, basically stating that (g, g^a, g^b, g^{ab}) is indistinguishable from (g, g^a, g^b, g^c), is equivalent to the indistinguishability of the ElGamal encryption, with ciphertexts $(g^r, pk^r \cdot m)$.

The Primitives. One way to capture the DDH assumption as a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ according to our terminology is to let the set $\mathcal{F}_\mathcal{Q}$ consist of random variables f that output a random group instance whose size is determined by the security parameter input. The relation $\mathcal{R}_\mathcal{Q}$, on input a pair $(f, \mathcal{A})$ of a instance and an adversary, generates such a group with generator g through f, picks a random bit d and random elements a, b, c in the range of the group's order. It then runs

the adversary $\mathcal{A}$ on a DDH tuple (g, g^a, g^b, g^{ab}) if $d = 0$, or on a random tuple (g, g^a, g^b, g^c) in case of $d = 1$. The adversary $\mathcal{A}$ is in the relation if it can predict d with non-negligible advantage over $\frac{1}{2}$.

Note that the above is *one* way to capture the DDH assumption and that there may be others. The choice may also influence the type of reduction we obtain, underlining once more the importance of specifying the primitives clearly. Our choice here matches the idea of the one-way function case, where the functional part provides the core functionality of the primitive, i.e., allowing the evaluation of the primitive, and the relation part defines its security property. More generally, we can model any falsifiable hardness assumptions (in the sense of [GW11]) analogously, letting the relation take on the role of the challenger in the (possibly interactive) security game with the adversary.

The second primitive, namely $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$, represents any IND-CPA secure encryption scheme. Here, the set $\mathcal{F}_\mathcal{P}$ contains all triples of algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfying the correctness property of a public key encryption scheme. Accordingly, we define $\mathcal{R}_\mathcal{P}$ by saying that $(G, \mathcal{S}) \in \mathcal{R}_\mathcal{P}$ if and only if $\mathcal{S}$ wins the IND-CPA distinguishing game for G with non-negligible probability.

The (c)onstruction. We can construct $\mathcal{P}$ from $\mathcal{Q}$ in the obvious way. That is, we may specify an ElGamal construction G^f that uses black-box access to $f \in \mathcal{F}_\mathcal{Q}$ (as specified above) to obtain a description of a random group. Using this description, the construction performs operations on the group in order to implement the algorithms of the ElGamal scheme. The construction is black-box with respect to the DDH primitive, matching the intuition that for any group (distribution), we obtain an encryption scheme whose security is directly related to the hardness of the DDH assumption in the underlying group (distribution).

The reduction – (a)dversary. Let us briefly recall the interaction between the reduction $\mathcal{S}$ and the adversary $\mathcal{A}$ in order to see that $\mathcal{S}$ uses $\mathcal{A}$ only in a black-box way. The reduction obtains a group description and a challenge triple (g^a, g^b, g^c) as input from the DDH game. Then, it runs the adversary oracle on the public key g^a and embeds the DDH challenge into the challenge ciphertext during the simulation of the IND-CPA game, i.e., it calculates $C \leftarrow (g^b, g^c \cdot m_d)$ for a randomly chosen bit d and returns C to the adversary. Finally, the reduction outputs 1 if and only if the adversary's output d' matches d. Hence, the reduction only uses the adversary as an oracle.

The reduction – (p)rimitive. Here, the same discussion as for the construction applies—the primitive is treated as a black box; the reduction merely performs group operations on the group that is generated by the primitive oracle. This matches the intuition that the reduction works for an arbitrary group (distribution).

In conclusion we hence have a BBB reduction in this case.

A.2 The Case of Weak One-Way Functions

Let us consider the primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ describing weak one-way functions, i.e., functions f for which there exists a function $\epsilon(n)$ bounded away non-negligibly from 1, such that for any PPT adversary $\mathcal{A}$ we have

$$\mathrm{Prob}\left[\mathcal{A}(1^n, f(x)) \to x' \in f^{-1}(x)\right] \leq \epsilon(n) + \mathsf{negl}(n).$$

The relation $\mathcal{R}_\mathcal{Q}$ associates to each $f \in \mathcal{F}_\mathcal{Q}$ a function ϵ as above, and contains a pair $(f, \mathcal{A})$ if $\mathcal{A}$'s inversion probability exceeds $\epsilon(n)$ non-negligibly.

The concatenation construction which evaluates f for $\Theta(n/\epsilon(n))$ independent inputs yields a (strong) one-way function [Yao82]. Note that in the construction, the number of function evaluations depends on the specific parameter ϵ of the weak one-way function. Since the reduction only makes black-box use of the adversary but also relies on ϵ, the transformation is thus an NBN-reduction in our terminology.

However, if we change the viewpoint slightly, and define the primitive $\mathcal{Q}_\epsilon = (\mathcal{F}_{\mathcal{Q}_\epsilon}, \mathcal{R}_{\mathcal{Q}_\epsilon})$ to contain all functions for some global bound ϵ, i.e., such that $\mathcal{R}_{\mathcal{Q}_\epsilon}$ consists of all pairs $(f, \mathcal{A})$ with $f \in \mathcal{F}_{\mathcal{Q}_\epsilon}$ and where $\mathcal{A}$'s inversion probability is non-negligibly larger than $\epsilon(n)$, then we obtain a BBB-reduction for the concatenation construction (and its reduction) based on the same global ϵ. This shows that the type of reduction critically depends on the definition of the primitive.

B On the Definition of Secure Primitives

Recall that Reingold et al. [RTV04] define what it means that an adversary $\mathcal{A}$ breaks an instance f of a primitive via relations $\mathcal{R}$ over pairs $(f, \mathcal{A})$. In this sense their notion of semi black-box reductions demands that, if there exists an adversary $\mathcal{A}$ with $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$, then there exits an adversary $\mathcal{S}^f$ such that $(f, \mathcal{S}^f) \in \mathcal{R}_\mathcal{Q}$. Accordingly, for weakly black-box reductions the prerequisite becomes $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$. Reingold et al. then claim, among others, that any semi black-box reduction implies a weakly one. They attribute the proof as straightforward from the definition ([RTV04, Lemma 1]).

This implication, however, is only true in general if one uses a *functional* description of adversaries, not through Turing machines. Otherwise, the intuitively convincing argument that for weakly black-box reductions the requirement that $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$ is satisfied if $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$ (i.e., if $\mathcal{A}$ can also query the instance f, as in semi black-box reductions) can be refuted via contrived relations $\mathcal{R}_P$ which output 0 if the adversary is syntactically an oracle machine, and 1 otherwise. That is, any non-oracle adversary breaks primitive $\mathcal{P}$ but any oracle machine fails. If in addition $\mathcal{R}_\mathcal{Q}$ is constantly 0, then there exists a trivial semi black-box reduction from $\mathcal{P}$ to $\mathcal{Q}$, but provably no weakly black-box reduction. This is quite remarkable in light of the discussion in [RTV04] that weakly black-box reduction are close in spirit to free (i.e., non-black-box reductions) and that separation result usually would only hold above the level of weakly reductions.

If one uses Turing machines to represent adversaries the problem is the (lack of) semantics of the security relations: any oracle machine should be at least as powerful as an oracle machine which decides to ignore its oracle. This restriction must then be put explicitly into the security notion defined via relations and must be checked for each separation, if one relies on the hierarchy of separations. We note, however, that we are not aware of any common security notion of a primitive which violates this semantics requirement. Alternatively, and this is our approach in the main part of the paper here, one can simply use a functional definition of adversaries in terms of random variables such that the structural details of the implementations cannot be used in the relation.[2]

To use the machine model, while simultaneously ruling out syntactical oddities, we say that a relation is semantical if purely syntactical changes do not affect the success of an adversary.

Definition B.1 (Output Distribution) *A probabilistic interactive (oracle) Turing machine $\mathcal{A}$ together with its oracle defines an output distribution, namely, each fixed finite sequence of inputs fed to $\mathcal{A}$ induces a distribution on the output sequences by considering all random choices of $\mathcal{A}$ and*

[2] Yet another way to define primitives would be to adopt the notion of random systems as put forward by Maurer [Mau02]. We feel, however, that the concept of a random variable is more readily accessible, even when they are possibly augmented with oracles and thereby implicitly induce the actual random variable.

its oracle. The output distribution of $\mathcal{A}$ is defined to be the set of these distributions, indexed by the finite sequences of input values.

Definition B.2 (Semantical Primitive) *A primitive $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ is called* semantical*, if for all $f \in \mathcal{F}_\mathcal{P}$ and all probabilistic (oracle) Turing machines $\mathcal{S}$ and $\mathcal{S}'$ (including their oracles), it holds: If $\mathcal{S}$ induces the same output distribution as $\mathcal{S}'$, then $(f, \mathcal{S}) \in \mathcal{R}_\mathcal{P}$ iff $(f, \mathcal{S}') \in \mathcal{R}_\mathcal{P}$.*

With this semantic property all the (functional) results in the paper remain true even for machines.

C Remaining Notions of Reducibility

In this Section, we present a complete picture containing all the notions introduced in [RTV04] and in the present paper. Moreover, we give the omitted non-abbreviated versions of all definitions used in the paper. As a "look-up" table for all definitions, the reader might find Figure 3 helpful.

C.1 Notions of Reducibility for Inefficient Adversaries

The most liberal notion of (BYZ) reductions are BNN (aka. semi black-box) reductions. Here, the reduction algorithm may depend on both, the code of the primitive and the code of the adversary.

Definition C.1 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNN or Semi-blackbox reduction) *There exists a BNN 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 $f \in \mathcal{F}_\mathcal{Q}$ and for all machines $\mathcal{A}$ there is a probabilistic polynomial-time oracle machine $\mathcal{S}$ such that if $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$ then $(f, \mathcal{S}^{f,\mathcal{A}}) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

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

Definition C.2 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBB-reduction) *There exists a* NBB 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 $\mathcal{S}$ for every $f \in \mathcal{F}_\mathcal{Q}$ there exists a probabilistic polynomial-time oracle machine G such that:*

Correctness. *It holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

Security. *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.,*

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

Definition C.3 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBN-reduction) *There exists an* NBN-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 for every $f \in \mathcal{F}_\mathcal{Q}$, there exists a probabilistic polynomial-time oracle machine G such that:*

Correctness. *It holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

Security. *There is a probabilistic polynomial-time oracle algorithm $\mathcal{S}$ such that: for 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.,*

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

Definition C.4 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNN-reduction or $\forall\exists$-Semi-BB reduction) *There exists an* NNN-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 for every $f \in \mathcal{F}_\mathcal{Q}$, there exists a probabilistic polynomial-time oracle machine G such that:*

Correctness. *It holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

Security. *For all machines $\mathcal{A}$ such that $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$, then there exists a probabilistic polynomial-time oracle machine $\mathcal{S}$ such that $(f, \mathcal{S}^f) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

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

C.2 Notions of Reducibility for Efficient Adversaries

Definition C.5 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNBa-reduction for Efficient Adversaries) *There exists a* BNBa-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 probabilistic polynomial-time 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.,*

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

Definition C.6 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBNa-reduction for Efficient Adversaries) *There exists a* BBNa-reduction *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 probabilistic polynomial-time 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.,*

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

Definition C.7 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNNa-reduction for Efficient Adversaries) *There exists a* BNNa-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 $f \in \mathcal{F}_\mathcal{Q}$ and for all probabilistic polynomial-time oracle machines $\mathcal{A}$ there is a probabilistic polynomial-time oracle machine $\mathcal{S}$ such that if $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$ then $(f, \mathcal{S}^{f,\mathcal{A}}) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

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

Definition C.8 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBBa-reduction for Efficient Adversaries) *There exists an* NBBa-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 for every $f \in \mathcal{F}_\mathcal{Q}$, there exist probabilistic polynomial-time oracle machines G and $\mathcal{S}$ such that:*

Correctness. *It holds that* $G^f \in \mathcal{F}_\mathcal{P}$.

Security. *For every probabilistic polynomial-time 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.,*

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

We now turn to the notions that allow the construction G to depend on the instantiation of the primitive $\mathcal{Q}$ in a non-black-box way.

Definition C.9 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBNa-reduction for Efficient Adversaries) *There exists an* NBNa-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 for every* $f \in \mathcal{F}_\mathcal{Q}$*, there exists a probabilistic polynomial-time oracle machine* G *such that:*

Correctness. *It holds that* $G^f \in \mathcal{F}_\mathcal{P}$.

Security. *There is a probabilistic polynomial-time oracle algorithm* $\mathcal{S}$ *such that: for every probabilistic polynomial-time 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.,*

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

Definition C.10 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNNa-reduction for Efficient Adversaries) *There exists an* NNNa-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 for every* $f \in \mathcal{F}_\mathcal{Q}$*, there exists a probabilistic polynomial-time oracle machine* G *such that:*

Correctness. *It holds that* $G^f \in \mathcal{F}_\mathcal{P}$.

Security. *For all probabilistic polynomial-time machines* $\mathcal{A}$ *such that* $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$*, then there exists a probabilistic polynomial-time oracle machine* $\mathcal{S}$ *such that* $(f, \mathcal{S}^f) \in \mathcal{R}_\mathcal{Q}$*, i.e.,*

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

D Remaining Relations

Theorem D.1 (Equivalence of NNNa and NBNa—Theorem 2.6) *For all primitives* $\mathcal{P}$ *and* $\mathcal{Q}$*, there is a* $(\mathcal{P} \hookrightarrow \mathcal{Q})$*-NBNa-reduction for efficient adversaries* $\mathcal{A}$ *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[3] 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. □

Reingold et al. [RTV04] conjecture that there is no inherent restriction in treating the adversary as a black-box. Theorem 2.6 partially confirms this conjecture, at least for efficient adversaries. For notions of inefficient adversaries, however, non-black-box use of the adversary is a promising approach to overcome existing impossibility results, as the following theorem shows. Namely, we prove that for inefficient adversaries, BNB-reductions do not imply BBB-reductions, and BNN-reductions do not imply BBN reductions. For both separations, we will consider a reduction that has to depend on the adversary in a non-black-box way, namely the famous Goldreich–Levin hardcore bit reduction [GL89].

Theorem D.2 *There are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is an $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNB reduction, but no BBB reduction. For the same two primitives, there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNN reduction, but no BBN reduction, as well as a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNN reduction, but no NBN reduction.*

Proof. We will prove that there are two primitives such that there is an BNB-reduction and thus, by Theorem 2.5 also a BNN and a NNN reduction. We will then prove that for the same two primitives, there is no NBN-reduction and therefore (again according to Theorem 2.5) neither a BBN, nor a BBB reduction. The common element here is, of course, that for a reduction between the primitives, the reduction needs to depend on the adversary in a non-black-box way. We will see that the Goldreich–Levin hardcore bit construction has this property.

We define primitives $\mathcal{Q}$ and $\mathcal{P}$ both as random oracles for length-doubling functions with different interfaces/security games/security relations. For security parameter λ, the input length to the functions is λ. To break $\mathcal{Q}$, the adversary is given $f(x)$ for a random x and may query the random oracle. The adversary is successful if it determines x' with $f(x') = f(x)$ with non-negligible probability. To break $\mathcal{P}$, an adversary is given $f(x)$ for a random x and a random string r of length $|x|$. That is, G^f is the primitive which samples x, r and outputs $f(x)$ and r. The adversary wins if it can determine the inner product $\langle x, r \rangle$ over $\mathbb{Z}_2$ with a non-negligible advantage over $\frac{1}{2}$. In short, $\mathcal{Q}$ is the game for one-wayness of a random oracle, and $\mathcal{P}$ is the game for the prediction of the Goldreich–Levin hardcore bit.

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

The Goldreich–Levin reduction [GL89] now proves that if there is a successful adversary $\mathcal{A}$ against $\mathcal{P}$ then we can invert the one-way function. As the reduction highly depends on the success probability of the adversary $\mathcal{A}$ but is black-box with respect to the implementation of the one-way function, and as, likewise, the hardcore bit construction is black-box, the Goldreich–Levin reduction is a BNB-reduction.

If we can prove that the dependence of the reduction on the adversary's success probability is necessary, then we know that there is no NBN-reduction from $\mathcal{P}$ to $\mathcal{Q}$. However, as our primitives are random oracles, this follows by a simple information-theoretic argument, namely, if the reduction asks the adversary, say, p times while the adversary's success probability is less than, say, $\frac{\lambda}{4p}$, then, on average, the reduction can extract at most $p \cdot \frac{\lambda}{4p} = \frac{\lambda}{4}$ bits of information from the adversary. Using Hoeffding's bound, we obtain that the adversary gives a correct response at most $\frac{\lambda}{2}$ times, which is not enough to computa a pre-image of length λ. We now state this argument formally.

Let $p(\lambda) \geq \lambda^2$ be an upper bound on the running time of the reduction $\mathcal{S}$, then $\mathcal{S}$ queries $\mathcal{A}$ at most $p(\lambda)$ times. We will now construct an (inefficient) adversary $\mathcal{A}$ that has non-negligible winning advantage against $\mathcal{P}$, and yet, $\mathcal{S}$ only breaks the one-wayness of $\mathcal{Q}$ with negligible probability when given access to $\mathcal{A}$. Let $\epsilon(\lambda) := \frac{\lambda}{4p(\lambda)}$. On input $(f(x), r)$, the adversary $\mathcal{A}$ returns a random bit with probability $1-\epsilon$. With probability ϵ, the adversary computes the smallest x' such that $f(x) = f(x')$ and returns $\langle x', r\rangle$. Note that with overwhelming probability, for a random x, there is no second pre-image for $f(x)$ under f, as it is a length-doubling function. Thus, the adversary $\mathcal{A}$'s success probability is (negligibly close to) $\frac{1}{2} + \epsilon$, and the adversary therefore breaks primitive $\mathcal{P}$.

We now prove that with overwhelming probability, the adversary does not decide to return the correct answer more than $\frac{\lambda}{2}$ times. Towards this goal, consider the Chernoff-Hoeffding bound for independent Benoulli random variables X_i that all have mean μ, and let $0 \leq \delta \leq 1$ be a parameter.

$$\text{Prob}\left[\sum_{i=1}^{n} X_i > (1+\delta)n\mu\right] \leq e^{-\frac{n\mu\delta^2}{3}}.$$

We set $X_i = 0$, if the adversary decides to return a random bit on the ith query and $X_i = 1$, if the adversary decides to return the correct answer. Then, μ equals $\epsilon(\lambda) = \frac{\lambda}{4p(\lambda)}$. We set $n := p(\lambda)$ as the upper bound on the number of queries made by the reduction. Set $\delta := 1$, then we yield that the probability that the adversary decides to return the correct answer more than $\frac{1}{2\lambda}$ times is negligible, i.e:

$$\text{Prob}\left[\sum_{i=1}^{p(\lambda)} X_i > \frac{\lambda}{2}\right] = \text{Prob}\left[\sum_{i=1}^{p(\lambda)} X_i > (1+1)p(\lambda)\frac{\lambda}{4p(\lambda)}\right] \leq e^{-p(\lambda)\frac{\lambda}{4p(\lambda)}\frac{1^2}{3}} = e^{-\frac{\lambda}{12}}.$$

We conclude that the probability that the adversary decides to return a correct answer (and not a random reply) more than once when beeing invoked $p(n)$ times, is negligible. As a thought experiment, we can thus replace the (stateless) adversary $\mathcal{A}$ by a stateful adversary $\mathcal{A}'$ which draws a set of $\frac{\lambda}{2}$ random indices between 1 and $p(n)$ in the beginning. Then, whenever the index is in this set, the adversary computes the smallest x' such that $f(x) = f(x')$ and returns $\langle x', r\rangle$. Else, the adversary returns a random response. Moreover, we can have $\mathcal{A}'$ to return some additional information indicating the reply's correctness, namely 0, if it outputs a random bit, and 1, if it returns the actual bit $\langle x', r\rangle$. This adversary is actually "more helpful" to the reduction than the original one, because the reduction knows which bits are correct and which bits are random. In a next step, since the other answers of $\mathcal{A}'$ are all random bits, we can replace $\mathcal{A}'$ by the adversary $\mathcal{A}''$ who can only be queried only $\frac{\lambda}{2}$ times and who always returns $\langle x', r\rangle$ on input $f(x)$, where x' is the smallest element such that $f(x) = f(x')$.

It remains to prove that no efficient reduction can invert f on a random input when being allowed a $\frac{\lambda}{2}$ queries to $\mathcal{A}''$. Towards this goal, we consider the random oracle via lazy sampling.

Before making a query to $\mathcal{A}''$, the reduction's probability of finding a pre-image of y in a single query is roughly $2^{-|y|} + 2^{-|x|} = 2^{-3\lambda} + 2^{-\lambda}$, i.e., the probability that y is sampled as the answer plus the probability, that the reduction queries the real pre-image x. After learning the inner product of x with some value r, the pre-image space for x is divided into two halves, so that the reduction's success probability increases to $2^{-3\lambda} + 2 \cdot 2^{-\lambda}$ per query, which is still negligible. Repeating this process $\frac{\lambda}{2}$ times yields a success probability of roughly $2^{-3\lambda} + 2^{\frac{\lambda}{2}} \cdot 2^{-\lambda}$ which is still negligible. □

For the following theorem, we interpret the results of Lin, Trevisan, and Wee [LTW05] and Yao [Yao82] as an instance of our framework. Namely, Yao [Yao82] shows how to construct strong one-way functions out of weak one-way functions via an NBN-reduction, while Lin, Trevisan, and Wee [LTW05] show that any such construction has to depend on the weakness parameter of the weak one-way function. In other words, one cannot have any BYZ-reduction between these two primitives.

Theorem D.3 *There exists primitives $\mathcal{P}$ and $\mathcal{Q}$ such that for all $YZ \in \{BN, BNa, NN, NNa\}$, there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$NYZ$-reduction but there is no $(\mathcal{P} \hookrightarrow \mathcal{Q})$-$BYZ$-reduction.*

Proof. For completeness, we now review the construction by Yao [Yao82] and also give a simple impossibility in the spirit of Lin, Trevisan and Wee [LTW05] to explain why one cannot build strong one-way functions out of weak one-way functions via a BYZ-reduction.

Recall that a one-way function is a function that is hard to invert on a random input, i.e., for all efficient adversaries $\mathcal{A}$ we have

$$\text{Prob}\left[\mathcal{A}(1^n, f(x)) \to x' \in f^{-1}(x)\right] \leq \mathsf{negl}(n).$$

A *weak* one-way function is a function that is one-way on a certain fraction on its input domain. In other words, a weak one-way function is secure if there is a function $\epsilon(n)$ bounded away non-negligibly from 1, such that for any efficient adversary $\mathcal{A}$ the inverting probability is essentially at most $\epsilon(n)$. Formally,

$$\text{Prob}\left[\mathcal{A}(1^n, f(x)) \to x' \in f^{-1}(x)\right] \leq \epsilon(n) + \mathsf{negl}(n).$$

Yao [Yao82] proved that any weak one-way function can be transformed into a one-way function via concatenation, i.e., if f is a weak one-way function with parameter $\epsilon(n)$, then for $k := n \cdot \left\lceil \frac{1}{\epsilon(n)} \right\rceil$, one has that

$$G^f(x_1||...||x_k) := f(x_1)||...||f(x_k), \text{ where } |x_i| = n,$$

is a one-way function. The reason is, that with overwhelming probability, for a random $x = x_1||...||x_k$ at least one of the x_i lies in the hard ϵ-fraction of the weak one-way function f. Clearly, this is a non-black-box construction, as we use the parameter ϵ to construct G. Note that, depending on f and G, the adversary $\mathcal{A}$ now expects inputs of a certain format and thus, Yao's reduction is only black-box with respect to the adversary, but not with respect to the function f. Thus, it is an NBN-reduction. By Theorem 2.5, it is also an NNN-reduction, and due to Theorem 2.7, these reductions also work when restricted to efficient adversaries.

We now give some intuition why the dependence on ϵ is necessary. Consider Yao's concatenation operator construction, where the number of queries that the construction makes to f depends on the parameter $\epsilon(n)$ which determines the degree of one-wayness of the weak one-way function f.

Assume that the concatenation construction is the above construction for some value $k = n \cdot n^c$ with $c > 1$ (The proof holds even more for any k for which $n \cdot n^c$ is an upper bound.). Then, set $\epsilon(n) := n^{-2c}$ and let f be a length-preserving function that behaves like a random oracle on a $\epsilon(n)$ fraction of its inputs, and let f return the all-zero string, else. Formally, let R be a length-preserving random oracle, we define

$$f(x_i) := \begin{cases} R(x_i) & \text{if } \langle x_i \rangle < \epsilon(|x_i|)2^{|x_i|}\,; \\ 0^{|x_i|} & \text{otherwise.} \end{cases}$$

Here, $\langle x_i \rangle$ denotes the value of x_i when x_i is interpreted as a natural number. Let $\mathcal{A}$ be the adversary that on input $(1^n, y)$ returns a random value $x \leftarrow \{0,1\}^{|y|}$. We prove that $\mathcal{A}$ has a noticeable winning probability against G^f. Towards this goal, we show that with noticeable probability over x, the construction returns the all-zero string, i.e., $G^f(x) = 0^{|x|}$. If the challenge value is the all-zero string and if the adversary picks a preimage of an all-zero string, then the adversary is successful. As we will show, both events happen with noticeable probability and as both events are independent (the adversary ignores its input challenge), the overall success probability of the adversary is noticeable, too.

Recall that $k = n \cdot n^c$. For a random $x = x_1||...||x_k$, we have that the probability that $G^f(x) = 0^{|x|}$, i.e., that for all i, it holds that $x_i \geq \epsilon(|x_i|)2^{|x_i|}$ is lower bounded by

$$\begin{aligned} \text{Prob}_{x_i}\left[G^f(x) = 0^{|x_i|}\right] &\geq (1-\epsilon(n))^k \\ &\geq (1-n^{-2c})^{n \cdot n^c} \\ &\geq 1 - n^{-2c} \cdot n \cdot n^c \qquad (3) \\ &\geq 1 - n^{-(c-1)} \qquad (4) \\ &\geq \tfrac{7}{8}, \end{aligned}$$

where (3) follows from the Bernoulli inequality, stating that $(1+z)^r \geq 1 + rz$ for $r \geq 0$ and $z \geq -1$ and (4) holds for sufficiently big security parameters n. We see that the probability that $G^f(x) = 0^{|x|}$ for a random x is greater than $\frac{7}{8}$ and thus, the probability that $\mathcal{A}$ returns a pre-image of an all zero-string is greater than $\frac{7}{8}$ and the probability that the adversary's challenge is the all-zero string is also greater than $\frac{7}{8}$. As both events are independent, the probability that both events happen is greater than $(\frac{7}{8})^2$ and thus, $\mathcal{A}$ has a noticeable success probability against G^f.

For general constructions, we refer the reader to Lin, Trevisan, and Wee [LTW05]. Note that they do not only show that the dependence on ϵ is necessary, they also prove quantitative lower bounds on the number of queries that the construction needs to make.

Theorem D.4 *For $X \in \{N, B\}$, there exist primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-XBN-reduction, a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-XBNa-reduction, a BNN-reduction and BNNa-reduction but such that there is no $(\mathcal{P} \hookrightarrow \mathcal{Q})$-XBB/XBBa/BNB/BNBa-reduction.*

Proof. As in previous proofs, we will show two separations via a single separation, namely, we define two primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBN-reduction (and thus an $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBN-reduction via Theorem 2.5, as well as an NBNa-reduction and a BBNa reduction), but no $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBBa-reduction (and thus no BBBa/BBB/NBBa-reduction). We will then show the case BNB/BNBa versus BBN/BBNa separately.

For the primitive $\mathcal{P}$, we consider a trivial primitive, namely the constant zero function, denoted f_0. The pair $(f_0, \mathcal{A})$ is in $\mathcal{R}_\mathcal{P}$ for all adversaries $\mathcal{A}$. For the primitive $\mathcal{Q}$, we consider the two random variables f_0 and f_1 that take as input a string. The random variable f_b returns 1, if the first bit of

the input string is b, and 0, else. We define $(b, \mathcal{A}) \in \mathcal{R}_{\mathcal{Q}}$ if and only if $\mathcal{A}$ on input $\perp$ queries b to the primitive with probability 1 for $b \in \{0, 1\}$. In the BBNa-reduction, the reduction $\mathcal{S}$ may depend on the primitive $\mathcal{Q}$ and ignores the given adversary $\mathcal{A}$. If the primitive $\mathcal{Q}$ is instantiated by f_0, then the reduction $\mathcal{S}$ constantly returns 0. If the primitive $\mathcal{Q}$ is instantiated by f_1, then the reduction $\mathcal{S}$ constantly returns 1. Thus, there is a BBN-reduction. In contrast, there is no NBBa-reduction, as we can consider the adversary $\mathcal{A}$ that does nothing (and is still successful by definition) and then, $\mathcal{S} = \mathcal{S}^{\mathcal{A}}$ cannot return both, 0 and 1 with probability 1. Thus, $\mathcal{S}$ fails for at least one of the two primitives $f_0, f_1 \in \mathcal{F}_{\mathcal{Q}}$, which concludes the proof.

By inspection, the given reduction is also a BNN/BNNa-reduction, and the impossibility considerations also apply to BNN/BNNa-reductions.

E Meta-Reductions: The Complete Picture

Definition E.1 (BNB-Meta-Reduction) *For primitives $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{N}$, a probabilistic polynomial-time algorithm $\mathcal{M}$ is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}BNB) \hookrightarrow \mathcal{N}$-meta-reduction from a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNB-reduction to $\mathcal{N}$, if the following holds for all $g \in \mathcal{F}_{\mathcal{N}}$:*

Reduction implies Insecurity. *If $\mathcal{P}$ BNB-black-box reduces to $\mathcal{Q}$ via a construction G, then there is an adversary $\mathcal{A}$ such that for all reductions $\mathcal{S}$ that instantiate the simple black-box reduction for $\mathcal{A}$, there is a $\mathsf{PPT}\mathcal{M}$ such that $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$.*

$$\begin{aligned}
&\forall g \in \mathcal{F}_{\mathcal{N}}\ \forall \mathsf{PPT} G\ \exists \mathcal{A}\ \forall \mathsf{PPT}\mathcal{S}\ \exists f \in \mathcal{F}_{\mathcal{Q}}\ \exists \mathsf{PPT}\mathcal{M} \\
&\Big[\big((G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}\big) \\
&\Rightarrow (g, \mathcal{M}^g) \in \mathcal{R}_N\Big]
\end{aligned}$$

Constructing BBB-meta-reductions is substantially easier than building a BNB-meta-reduction $\mathcal{M}$. The reason is that in BNB-reductions, the algorithm $\mathcal{S}$ might depend on the adversary $\mathcal{A}$ in a non-black-box way. We now indicate a possible technique that makes BNB-reductions applicable in some cases, for example, when there is an inefficient adversary that is successful for all primitives f and only depends on the primitive f in a black-box way. For efficiently computable one-way functions (i.e., not one-way oracles such as random oracles), an adversary might simply break the function via brute force.

Let G be a construction that instantiates a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNB-reduction. Assume, that for all such constructions G, there is an inefficient adversary $\mathcal{A}$ which is successful against G^f for all *all* implementations f of $\mathcal{Q}$. We let the meta-reduction $\mathcal{M}$ proceed as follows: For the non-efficient adversary $\mathcal{A}$, let $\mathcal{S}$ be a tailor-made reduction algorithm. Note that that $\mathcal{S}$ may depend on the code of the adversary $\mathcal{A}$, so that $\mathcal{S}$ can internally simulate parts of $\mathcal{A}$. However, $\mathcal{A}$ is (possibly) inefficient and thus $\mathcal{S}$ is given black-box access to $\mathcal{A}$. If the meta-reduction $\mathcal{M}$ can simulate the output distribution defined by $\mathcal{A}$ efficiently (by rewinding the reduction $\mathcal{S}$, for example) then $\mathcal{M}$ can break every $f \in \mathcal{F}_{\mathcal{Q}}$, and in particular g.

We now turn to BNB-meta-reductions, for which a large number of results (mainly based on rewinding techniques) is known.

Theorem E.2 *If $\mathcal{N}$ exists and if there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}BNB) \hookrightarrow \mathcal{N}$-meta-reduction, then there is no BNB-reduction from $\mathcal{P}$ to $\mathcal{Q}$.*

The proof is analogously to the proof of Theorem 4.2. As before, we obtain a similar corollary as corollary 4.3.

Corollary E.3 *If there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}BNB) \hookrightarrow \mathcal{Q}$-meta-reduction, then then a secure instantiation of $\mathcal{P}$ cannot be based on the existence of $\mathcal{Q}$ via a BNB-reduction from $\mathcal{P}$ to $\mathcal{Q}$.*

Note that the corollary immediately implies, too, that one cannot have a reduction of the "higher" BBB type from $\mathcal{P}$ to $\mathcal{Q}$, as such a reduction would imply a BNB-reduction.

The next definition captures the reduction type ruled out by, for instance, by Fischlin and Schröder [FS10]: the adversary is treated as a black-box, while the primitive can also be used in a non-black-box way.

Definition E.4 (BBN-Meta-Reduction) *A meta-reduction $\mathcal{M}$ is a* BBN *from primitive $\mathcal{P}$ to primitive $\mathcal{Q}$, if the following holds for all implementations $g \in \mathcal{F}_{\mathcal{N}}$:*

Reduction implies impossibility. *If $\mathcal{P}$ BBN-reduces to $\mathcal{Q}$, then for all constructions G that implement the $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBN-reduction, there is an implementation $f \in \mathcal{F}_{\mathcal{Q}}$ such that for all efficient reductions $\mathcal{S}$ implementing the $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BBN-reduction for f, there exists a probabilistic polynomial-time oracle machine $\mathcal{M}$ such that $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$.*

$$\begin{aligned}&\forall g \in \mathcal{F}_{\mathcal{N}}\ \forall G\ \exists f \in \mathcal{F}_{\mathcal{Q}}\ \forall \mathsf{PPT} \mathcal{S}\ \exists \mathcal{A}\ \exists \mathsf{PPT} \mathcal{M}\\ &\Big[\big((G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}\big)\\ &\Rightarrow (g, \mathcal{M}^g) \in \mathcal{R}_N\Big]\end{aligned}$$

As an example for the approach for BBN-meta-reductions, consider the approach by Fischlin and Schröder [FS10] for blind signatures. There, $\mathcal{N}$ equals $\mathcal{Q}$ and is some non-interactive cryptographic assumption, while $\mathcal{P}$ is a (blind) signature scheme. To prove a BBN-meta-reduction, the natural choice for f is to be equal to g. From each reduction algorithm $\mathcal{S}$, via rewinding techniques—used to extract signatures—one now constructs an adversary $\mathcal{A}$ that is successful against the primitive g. As a consequence, $\mathcal{S}^{\mathcal{A},g}$ breaks the primitive g and so does $\mathcal{M}^g$ when simulating the behavior $\mathcal{S}^{\mathcal{A},g}$ efficiently. (Note that, as for usual cryptographic security notions, the semantics requirement comes into play here.)

Again, a BBN-meta-reduction implies that a BBN-reduction cannot exist.

Theorem E.5 *If $\mathcal{N}$ exists and if there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}BBN) \hookrightarrow \mathcal{N}$-meta-reduction, then there is no BBN-reduction from $\mathcal{P}$ to $\mathcal{Q}$.*

Corollary E.6 *If there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}BBN) \hookrightarrow \mathcal{Q}$-meta-reduction, then then a secure instantiation of $\mathcal{P}$ cannot be based on the existence of $\mathcal{Q}$ via a BBN-reduction from $\mathcal{P}$ to $\mathcal{Q}$.*

For completeness, we also define BNN-meta-reductions. As for BNB-reductions, we are not aware of any existing results that use these as a technique.

Definition E.7 (BNN-Meta-Reduction) *A meta-reduction $\mathcal{M}$ is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}BNN) \hookrightarrow \mathcal{N}$-meta-reduction from a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNN-reduction to the primitive $\mathcal{N}$, if the following holds for all $g \in \mathcal{F}_{\mathcal{N}}$:*

Reduction implies insecurity. *If $\mathcal{P}$ BNN-reduces to $\mathcal{Q}$, then for all constructions G that implement the $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNN-reduction, there is an implementation $f \in \mathcal{F}_{\mathcal{Q}}$ and an adversary $\mathcal{A}$ such that for all efficient reduction algorithms $\mathcal{S}$ implementing the $(\mathcal{P} \hookrightarrow \mathcal{Q})$-BNN-reduction for f and $\mathcal{A}$, there exists a probabilistic polynomial-time oracle machine $\mathcal{M}$ such*

that $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$.

$$\begin{aligned}&\forall g \in \mathcal{F}_{\mathcal{N}}\ \forall \mathsf{PPT} G\ \exists f \in \mathcal{F}_{\mathcal{Q}}\ \exists \mathcal{A}\ \forall \mathsf{PPT} \mathcal{S}\ \exists \mathsf{PPT} \mathcal{M}\\ &\Big[\big((G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}\big)\\ &\Rightarrow (g, \mathcal{M}^g) \in \mathcal{R}_N\Big]\end{aligned}$$

Theorem E.8 *If $\mathcal{N}$ exists and if there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}BNN) \hookrightarrow \mathcal{N}$-meta-reduction, then there is no BNN-reduction from $\mathcal{P}$ to $\mathcal{Q}$.*

Corollary E.9 *If there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}BNN) \hookrightarrow \mathcal{Q}$-meta-reduction, then then a secure instantiation of $\mathcal{P}$ cannot be based on the existence of $\mathcal{Q}$ via a BNN-reduction from $\mathcal{P}$ to $\mathcal{Q}$.*

E.1 Meta-Reductions for N$**$-Reductions

We now introduce the corresponding reductions that can be used to rule out N$**$-type reductions, i.e., for all the following definitions ($YZ \in \{\mathrm{BB}, \mathrm{BN}, \mathrm{NN}\}$), it holds that if there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}) \hookrightarrow \mathcal{N}$-N$YZ$-meta-reduction from a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-N$YZ$-reduction to the primitive $\mathcal{N}$, then if the primitive $\mathcal{N}$ exists, then there cannot be a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-N$YZ$-reduction. In particular, if $\mathcal{Q} = \mathcal{N}$, then $(\mathcal{P} \hookrightarrow \mathcal{Q})$-N$YZ$-reductions are ruled out unconditionally.

Definition E.10 ($(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}NBB) \hookrightarrow \mathcal{N}$-Meta-Reduction) *A meta-reduction $\mathcal{M}$ is an NBB-meta-reduction from a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBB-reduction to the primitive $\mathcal{N}$, if the following holds for all $g \in \mathcal{F}_{\mathcal{N}}$:*

Reduction implies insecurity. *If $\mathcal{P}$ NBB-reduces to $\mathcal{Q}$, then for reductions $\mathcal{S}$ that implement the $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBB-reduction, there is an implementation $f \in \mathcal{F}_{\mathcal{Q}}$ such that for all efficient constructions G that implement the $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBB-reduction, there is an adversary $\mathcal{A}$ such that there exists a probabilistic polynomial-time oracle machine $\mathcal{M}$ that breaks g, i.e., $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$.*

$$\begin{aligned}&\forall g \in \mathcal{F}_{\mathcal{N}}\ \forall \mathsf{PPT} \mathcal{S}\ \exists f \in \mathcal{F}_{\mathcal{Q}}\ \forall \mathsf{PPT} G\ \exists \mathcal{A}\ \exists \mathsf{PPT} \mathcal{M}\\ &\Big[\big((G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}\big)\\ &\Rightarrow (g, \mathcal{M}^g) \in \mathcal{R}_N\Big]\end{aligned}$$

Definition E.11 ($(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}NBN) \hookrightarrow \mathcal{N}$-black-box Meta-Reduction) *A meta-reduction $\mathcal{M}$ is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}NBN) \hookrightarrow \mathcal{N}$-meta-reduction from a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBN-reduction to the primitive $\mathcal{N}$, if the following holds for all $g \in \mathcal{F}_{\mathcal{N}}$:*

Reduction implies insecurity. *If $\mathcal{P}$ NBN-reduces to $\mathcal{Q}$, then there is an implementation $f \in \mathcal{F}_{\mathcal{Q}}$ such that for all efficient constructions G and reductions $\mathcal{S}$ that implement the $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NBN-reduction, there is an adversary $\mathcal{A}$ and a PPT machine $\mathcal{M}$ that breaks g, i.e., $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$.*

$$\begin{aligned}&\forall g \in \mathcal{F}_{\mathcal{N}}\ \exists f \in \mathcal{F}_{\mathcal{Q}}\ \forall \mathsf{PPT} G\ \forall \mathsf{PPT} \mathcal{S}\ \exists \mathcal{A}\ \exists \mathsf{PPT} \mathcal{M}\\ &\Big[\big((G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}\big)\\ &\Rightarrow (g, \mathcal{M}^g) \in \mathcal{R}_N\Big]\end{aligned}$$

Definition E.12 ($(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}NNN) \hookrightarrow \mathcal{N}$-semi black-box meta-reduction) *A meta-reduction $\mathcal{M}$ is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}NNN) \hookrightarrow \mathcal{N}$-meta-reduction from a $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNN-reduction to the primitive $\mathcal{N}$, if the following holds for all $g \in \mathcal{F}_{\mathcal{N}}$:*

Reduction implies insecurity. *If $\mathcal{P}$ NNN-reduces to $\mathcal{Q}$, then there is an implementation $f \in \mathcal{F}_{\mathcal{Q}}$ such that for all efficient constructions G that implement the $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNN-reduction, there is an adversary $\mathcal{A}$ such that for all efficient reductions $\mathcal{S}$ implementing the $(\mathcal{P} \hookrightarrow \mathcal{Q})$-NNN-reduction there is a* PPT *machine $\mathcal{M}$ that breaks g, i.e., $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$.*

$$\begin{aligned}&\forall g \in \mathcal{F}_{\mathcal{N}}\ \exists f \in \mathcal{F}_{\mathcal{Q}}\ \forall \mathsf{PPT} G\ \exists \mathcal{A}\ \forall \mathsf{PPT} \mathcal{S}\ \exists \mathsf{PPT} \mathcal{M}\\ &\Big[\big((G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}\big)\\ &\Rightarrow (g, \mathcal{M}^g) \in \mathcal{R}_N\Big]\end{aligned}$$

For meta-reduction, a dual version of Theorem 2.5 holds, namely, the lower in the hierarchy a meta-reduction lies, the more general it is.

Theorem E.13 *Fix two types of CAP meta-reductions XYZ and $\widehat{X}\widehat{Y}\widehat{Z}$ such that $\widehat{X}\widehat{Y}\widehat{Z} \leq XYZ$ point-wise (where $N \leq B$) and let $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{N}$ be three primitives. If there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}\widehat{X}\widehat{Y}\widehat{Z}) \hookrightarrow \mathcal{N}$-meta-reduction, then there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}XYZ) \hookrightarrow \mathcal{N}$-meta-reduction. If there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}\widehat{X}\widehat{Y}\widehat{Z}a) \hookrightarrow \mathcal{N}$-meta-reduction, then there is a $(\mathcal{P} \hookrightarrow \mathcal{Q}\text{-}XYZa) \hookrightarrow \mathcal{N}$-meta-reduction.*

F Beyond the Impossible—More Liberal Notions of Reducibility

So far, we mainly considered unconditional separation results. However, there are natural limitations of those, namely, when ruling out very liberal notions of reductions by an unconditional separation result, then one often obtains a proof that **P** does not equal **NP** along the way. The latter can for example be circumvented by proving a meta-reduction and deploying complexity assumptions, such as the existence of a primitive or $\mathbf{NP} \not\subseteq \mathbf{BPP}$. Recently, Pass et al. [PTV11] gave a new proof to show that one-way functions do not imply one-way-permutations. Their proof uses a meta-reduction and is based on the assumption that $\mathrm{Dist}^{\text{one-sided}}\mathrm{co}\mathbf{NP} \not\subseteq \mathrm{Heur}_{1/\mathrm{poly}}\mathbf{AM}$. This even rules out very liberal reductions, that Reingold et al. [RTV04] call $\forall\exists$-weakly-BB reductions in their framework (wNNNa in our taxonomy). Interestingly, Reingold et al. prove that if there is some (even non-constructive, non-reductionist) proof that a primitive $\mathcal{Q}$ implies the existence of a primitive $\mathcal{P}$, then there is also a $\forall\exists$-weakly-BB reduction from $\mathcal{Q}$ to $\mathcal{P}$.

Definitions

Definition F.1 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-Weakly-BBa or wBNNa reduction for efficient adversaries) *There exists a wBNNa 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 probabilistic polynomial-time machine $\mathcal{A}$, for every implementation $f \in \mathcal{F}_{\mathcal{Q}}$, there is a probabilistic polynomial-time reduction algorithm $\mathcal{S}$ such that: if $(G^f, \mathcal{A}) \in \mathcal{R}_{\mathcal{P}}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$, i.e.,*

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

Definition F.2 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-wBNN reduction for inefficient adversaries) *There exists a* wBNN *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 $\mathcal{A}$, for every implementation $f \in \mathcal{F}_\mathcal{Q}$, there is a probabilistic polynomial-time reduction algorithm $\mathcal{S}$ such that: if $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

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

Definition F.3 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-$\forall\exists$-Weakly-BBa or wNNNa reduction for efficient adversaries) *There exists a* wNNNa *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 for every $f \in \mathcal{F}_\mathcal{Q}$, there exists a probabilistic polynomial-time oracle machine G such that:*

Correctness. *It holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

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

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

Definition F.4 ($(\mathcal{P} \hookrightarrow \mathcal{Q})$-wNNN reduction for inefficient adversaries) *There exists a* wNNN *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 for every $f \in \mathcal{F}_\mathcal{Q}$, there exists a probabilistic polynomial-time oracle machine G such that:*

Correctness. *It holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

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

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

Definition F.5 (Free Reduction) *There exists a* free *reduction from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$, if $\mathcal{P}$ exists whenever $\mathcal{Q}$ exists. A primitive $\mathcal{P}$ is said to exist if there is an $f \in \mathcal{F}_\mathcal{P}$ which has an efficient implementation such that there is no polynomial-time algorithm $\mathcal{A}$ with $(f, \mathcal{A}^f) \in \mathcal{F}_\mathcal{P}$.*

Clearly, wNNNa-reductions are implied by wBNNa-reductions, and both imply a free reduction. Moreover the inefficient adversary versions imply the efficient adversary variant.

F.1 Remarks

On the Equivalence of Free and wBNNa Reductions. Reingold et al. prove that if one-way functions imply key agreement via a free reduction, then there is a wBNNa-reduction for efficient adversaries between key agreement and one-way functions. As it is unlikely to rule out free reductions unconditionally (as this would prove unconditionally the existence of one-way functions and the non-existence of key agreement), the existing separation by Impagliazzo and Rudich can probably not be strengthened to also rule out wBNN-reductions for inefficient adversaries.

Inefficient adversaries and Impossibility Results. Note that, however, this does not hold for weakly-BB-reductions for inefficient adversaries, as the proof of equivalence by Reingold et al. does not carry over to the non-efficient notions.

We do not explore wBNN-reductions and wNNN-reductions in detail. Both notions are very liberal, as the reduction is only required to work for adversaries that do not have access to the primitive. We are not aware of any reductions in the literature that need such an advanced level of freedom. Thus, it is nice if separation results, conditional or unconditional ones, can be strengthened to also rule out these reductions, but from today's point of view, they do not seem to give more practical advice than separations for $*$NN-(aka semi and $\forall\exists$-semi-)reductions resp. their variants for efficient adversaries.

BNN and wBNN Reductions (seemingly) Equivalent. Another caveat with general relations is to correctly classify a reduction. The difference between BNN and wBNN-reductions is whether the adversary $\mathcal{A}$ gets access to an instance f or not. Consider security notions which are given as games, that, themselves, give the adversary $\mathcal{A}$ black-box access to f. Then, BNN and wBNN reductions are equivalent for this notion and formally, one might have an impossibility result even for wBNN reductions. However, when defining the relation carefully, for wBNN reductions, one should actually only consider those adversaries $\mathcal{A}$ that do not query f within the game. It is difficult to prevent such subtle misinterpretations, as general relations are very liberal notions; and further restrictions might affect the power of the framework. Restricting, for example, the attention to games would exclude distributionally one-way functions or the notion of **BPP** languages that, currently, can all be incorporated under a single paradigm.