Lifting Standard Model Reductions to Common Setup Assumptions

Ngoc Khanh Nguyen[1], Eftychios Theodorakis[2,3], and Bogdan Warinschi[2,4]

[1] IBM Research Europe – Zurich and ETH Zurich
[2] DFINITY
[3] Dicrypt
[4] University of Bristol
nkn@zurich.ibm.com crypto@eftychis.org bogdan.warinschi@gmail.com

Abstract. In this paper we show that standard model black-box reductions naturally lift to various setup assumptions, such as the random oracle (ROM) or ideal cipher model. Concretely, we prove that a black-box reduction from a security notion P to security notion Q in the standard model can be turned into a non-programmable black-box reduction from $P_{\mathcal{O}}$ to $Q_{\mathcal{O}}$ in a model with a setup assumption $\mathcal{O}$, where $P_{\mathcal{O}}$ and $Q_{\mathcal{O}}$ are the natural extensions of P and Q to a model with a setup assumption $\mathcal{O}$.

Our results rely on a generalization of the recent framework by Hofheinz and Nguyen (PKC 2019) to support primitives which make use of a trusted setup. Our framework encompasses standard idealized settings like the random oracle and the ideal cipher model. At the core of our main result lie novel properties of negligible functions that can be of independent interest.

1 Introduction

Security reductions. In this paper we investigate the interplay between security reductions and setup assumptions. *Security reductions* [15] are perhaps the single most powerful idea that underlies modern cryptography. Roughly speaking, a security reduction is an algorithm which turns an adversary that breaks some protocol Q into one which breaks some underlying primitive P [5]. If such a reduction exists, it follows that if P is secure, then so is Q, so the security of a complex system is reduced to that of its underlying components. Here, and throughout the paper, P and Q are understood as *security notions*; that is classes of instantiations together with an experiment which defines their security. Furthermore, reductions correspond to specific constructions which turn an instance of P into an instance of Q.

[5] We do not attempt to make a sharp distinction between primitives and protocols. We use the terms primitive and protocol loosely and only to emphasize that one employs the other in its design. P may also stand for cryptographic assumption, e.g. factorization is hard, just as Q may stand for a more involved primitive, e.g. authenticated encryption.

The landscape of reductions has been carefully mapped via a taxonomy introduced by Reingold, Trevisan and Vadhan (RTV) [22] and later refined by Baecher, Brzuska, and Fischlin [1]. In its latest incarnation, the taxonomy identifies three components relevant to reductions, namely, the construction, the adversary and the instance used in the construction, and classifies reductions depending on how they access these components. Of central interest in this taxonomy are the so-called "black-box" reductions where the reduction only gets oracle access (i.e. only input/output) to the adversary; further variants distinguish between how the reduction accesses the construction and the instance of the primitive used in the construction. Indeed, with only very few exceptions, the cryptographic practice employs black-box reductions. Not only are these reductions simpler to design but black-box access to the adversary and the instance enables a hierarchical modular design, thereby helping tame the inherent complexity of cryptographic designs.

Setup assumptions. For efficiency reasons, or to circumvent impossibility results, concrete instantiations of cryptographic constructs often rely on setup assumptions. That is, constructions make use of already set-up trusted components. Well-known examples of such assumptions include the random beacon model [21], the random oracle [3], the ideal cipher model [23], the common random string model [4], and its common reference string variant. Other examples include the quantum random oracle [5] or access to specific hardware [9,20,14].

Reductions may use setup assumptions in fundamental ways. They may track the adversary's queries towards the random oracle and program the output of random oracles at dynamically identified inputs. They can access a trapdoor associated to a common random string which allows them to decrypt adversarial ciphertexts or equivocate commitments.

In this paper we are interested in the interplay between reductions and security assumptions. Most of the previous work on classifying reductions does not explicitly surface the use of setup assumptions and, a priori, are set in the standard (vanilla) model. A notable exception is the work of Fischlin et al. [13] who extends the black-box separation techniques to get impossibility results for various constructions even in settings with a (or with variants of the) random oracle model.

The problem. In this paper we investigate the interaction between reductions and setup assumptions from a different perspective which we detail below. For concreteness, in our motivating discussion we use the random oracle as an example setup assumption under consideration. Nonetheless, our work treats generically other settings as well.

Assume that we have already designed a black-box reduction from some protocol Q to a primitive P. The reduction is in the standard model. Then consider protocol $Q_{\mathcal{O}}$ which uses in its construction instantiations of the primitive $P_{\mathcal{O}}$ which potentially rely on the random oracle $\mathcal{O}$. Can we conclude something about the security of $Q_{\mathcal{O}}$? Put differently, does a black-box reduction from Q to P in the standard model *lift* to a reduction from $Q_{\mathcal{O}}$ to $P_{\mathcal{O}}$? And what, and how does one define the extension of $P_{\mathcal{O}}$ to the random oracle, in the first place?

We contend that the answer is far from obvious. The first obstacle is a syntactic one. Observe that an adversary against $Q_{\mathcal{O}}$ makes queries that exercise the functionality of the protocol yet also queries the random oracle. The reduction from Q to P "knows" how to deal with the former type of queries but makes no provisions for the latter type. We consider the natural extension of the standard model reduction to a non-programming random oracle reduction, where the reduction simply forwards the queries and answers between the adversary and the random oracle.

The second obstacle is more substantial. How would one argue that the random oracle reduction works? Since the existence of the standard model reduction is the only available handle one would need to relate the event that a random oracle adversary wins to the event that a standard adversary wins. However, there is a fundamental difference between the standard model setting and the random oracle one.

In the standard model the only information available to the adversary about the internal state of the protocol is whatever can be inferred from their communication mediated by the security game. In the random oracle model, however, the adversary and the protocol indirectly share state through their joint access to the random oracle. With this in mind, it is unclear how to map events from the joint-state setting to the standard model or, indeed, whether this is even possible. It is conceivable that the adversary manages to break the protocol *because* of the shared state and this is something which the standard model reduction may not even account for. Looking ahead, we show how to bypass these obstacles and provide a positive answer to the question we posed above. We detail our results next.

Our contributions

Formal foundations. Our first contribution is a framework which allows to talk about "lifting" notions and reductions between notions from the standard model to a model with setup assumptions. Our starting point is the recent framework of Hofheinz and Nguyen [16], who in turn build on the work of Reingold, Trevisan, and Vadhan [22]. In their framework, the notion of a primitive has two key ingredients: i) a set theoretic notion of an *instance* (essentially the set of all instantiations for the primitive), and ii) an explicit notion of a *security* game – defined as an interactive (oracle) Turing machine (and an associated advantage function). We extend this framework in two ways. First, we formalize *setup assumptions* as a mathematical object, essentially as family of distributions over sets of functions. Later in the section, we outline a number of technical challenges we need to overcome to make this approach rigorous, and make our definitions precise and general. One can then extend arbitrary security games to include setup assumptions by providing to the adversary (and the primitive) black-box access to a function; which is sampled in an eager manner (from a computational perspective) according to the distribution(s) before the execution begins. Our abstract approach subsumes many of the widely used setup assumptions including the random oracle model, the CRS model, and the ideal cipher model.

Second, we provide a careful treatment of the notion of primitive instances. Both [22] and [16] define instances of a primitive as arbitrary sets of Turing machines. This approach is too abstract for our purposes since it does not give rise to a meaningful way of lifting the notion of an instantiation from the standard model to the random oracle model. A more concrete definition is necessary that allows one to explicitly define correctness and security membership sets for the primitives. We opt for identifying primitive instances by considering an explicit *correctness* game associated to the primitive.

With these extensions in place, we can then rigorously define the extension of a particular cryptographic notion to a specific setup assumption and the lifting of a reduction from the standard model to a model with a setup assumption.

Main result. Our main result establishes that fully black-box reductions in the standard model indeed *do* lift to setup assumptions. That is, if a standard model reduction from some protocol Q to some primitive P exists, then the reduction (or rather its canonical extension) also "works" in the setup assumption. The proof of this result is along the following lines. Once an individual instance of the setup assumption is fixed, then the adversary can be viewed as an adversary in the standard model with the instance of the setup hardwired in its code; the same observation holds for the primitive. One can therefore establish a relation between the success of the reduction and the success of the adversary, for each individual instance of the setup assumption. The crux of the proof is to show how to "aggregate" the distinct individual bounds on advantage functions to get a bound on the adversary's advantage when the setup is sampled according to its defining distribution.

A related and somewhat simpler case of this problem is to show that instantiating a protocol with a *correct* instance of the primitive with a setup assumption yields a *correct* instance of the protocol (with a setup assumption). We cast both of these problems as a generic property of (countable) sets of certain type of families of negligible functions.

Technical challenges. As it soon becomes clear in the paper, we require a lot of mathematical machinery, and it is instructive to understand the source of some of the complications we deal with. In particular, there are two related challenges rooted in an interesting interplay between fully black-box reductions and random oracles: (i) how to define a generic notion of a setup assumption and (ii) how to define the adversary's advantage. Recall that a fully black-box reduction "works" even if the adversary against the protocol is unbounded. In particular, the reduction needs to work even for an adversary that with some small probability does not stop and instead keeps on querying the random oracle on increasingly larger inputs. How should one then define the advantage of the adversary? The difficulty here is identifying the underlying sample space of the experiment since an unbounded adversary will essentially require an unbounded random tape.

This discussion also sheds some light on our modeling of a setup assumption. Intuitively, we would like to define a setup assumption simply as a function from some domain X to some co-domain Y to which the different parties involved in

the execution get access. The function would need to be sampled, eagerly, at the beginning of the execution. This intuitively appealing approach does not work for the type of infinite execution in the above discussion. For the random oracle model we would have $X = Y = \{0,1\}^*$ and it's not clear how to sample from this space "uniformly at random" as one would expect.

Our solution is to view the setup as a family of sampling algorithms indexed by a natural number ℓ. For each ℓ the setup is sampled from finite sets of functions with (now bounded) domain X_ℓ and range Y. Our formalization enforces that $X_\ell \subseteq X_{\ell+1}$ and that sampling is "consistent" across the parameters, that is the distribution on $Y^{X_{\ell+1}}$ extends naturally the distribution on Y^{X_ℓ}. For each parameter ℓ we define a corresponding execution model where the execution of the game aborts if either the adversary or the construction queries the setup on a point outside X_ℓ. With these bounds in place, we can rigorously show the sample space, required by the setup assumption, is well-defined, and the advantage of the adversary for each individual parameter is well defined and converges. That is the corresponding sums parameterized over ℓ converge and thus it makes sense to define the notions of adversarial advantage and correctness.

Applications. In order to illustrate the practicality of our main result, we present the following simple example. Consider the Lamport construction of a one-time (OT) signature scheme out of a one-way function (OWF). Let us call the generic construction $\mathsf{Lamp}[\cdot]$. The traditional reduction shows that $\mathsf{Lamp}[f]$ is a secure OT signature scheme if f is a OWF: for any OWF instance f, an adversary against $\mathsf{Lamp}[f]$ can be used in a black-box way to break f. Note that the reduction allows to establish the security of OT signature instances of the form $\mathsf{Lamp}[f]$ *only for instantiations* of f in the standard model.

Consider now an OWF instance which uses a random oracle (RO), e.g. consider the construction $g^{\mathcal{O}}$, where g simply forwards its inputs to the random oracle $\mathcal{O}$ and returns the result. We claim that, given the state of the art, it is not possible to immediately conclude $\mathsf{Lamp}[g^{\mathcal{O}}]$ is a secure OT signature. Indeed, one cannot draw any rigorous conclusions from existing results: even brushing under the carpet that $g^{\mathcal{O}}$ is "obviously" a OWF, the key observation is that the scheme $\mathsf{Lamp}[g^{\mathcal{O}}]$ is a scheme in the RO model. So, the existing reduction does not apply. It is here where our main result is useful: it lifts the reduction from the standard model to the random oracle model and allows us to conclude that the security of $\mathsf{Lamp}[f]$ reduces to that of f, even if f is a construction in the RO.

Obviously, one can re-establish the security of $\mathsf{Lamp}[g^{\mathcal{O}}]$ directly, in the random oracle, but that would require a new proof where one would have to redo the interesting part of the reduction.

To give another example, consider the black-box construction of a NM-CPA scheme out of a semantically secure scheme by Choi, Soled, Malkin, and Wee [10]. For brevity we shall call the construction CSMW. Their result shows that CSMW[Enc] is an NM-CPA scheme for any semantically secure scheme in the standard model.

Consider now the instantiation of CSMW[BR] where BR is the concrete semantically secure scheme from the original RO paper by Bellare and Rogaway[2],

i.e. for a trapdoor permutation f then $\mathrm{Enc}(m) = f(r)||H(r) \oplus m$. Our results allow one to conclude that CSMW[BR] is an NM-CPA scheme in the RO model. Without this contribution, one would have to the best of our knowledge provide a direct reduction to the security of f [6]. Generally, our results expose the concrete security gap, yet the theorems allow for abstract, and relatively simple, proofs as shown in section 4.5.

Discussion. Our main result shows how to lift fully black-box reductions set in the standard model to a model with a setup. In particular, we rely in a reasonably strong way on the fact that such reductions can deal with unbounded adversaries – at some point we need to hardwire a potentially large table (representing the setup) into adversaries and implementations. Consequently, the resulting constructions may not be efficient anymore. In turn this implies that our result does not immediately extend to a reduction which is only guaranteed to work for efficient adversaries e.g. [1, Section 2.6]. That would be extending our result to BBBa reductions using the terminology of [1]. Restricting our results to such a setting would, however, allow us to avoid many complications that the unbounded nature of the random oracle causes, as outlined above.

An intriguing question is whether our results extend to the case where starting reduction/construction is already in a model with already an idealized setup (as opposed to the standard model). In particular, answering this question raises the question of how idealized models interact/compose. We leave both of these questions to further work.

Related work. The closest work to ours is the work of Hofheinz and Nguyen [16]. They introduce a generic framework for abstractly specifying games (and security reductions) and use it to study the relation between single instance and multi-instance security of primitives. Our work extends their framework with setup assumptions and explicit correctness games, and we study a different extension of the reduction. A somewhat related line of work studies "relativizing" reductions [18,22,1]. This concept borrowed from complexity theory is about establishing relations of the type: if primitive P can be instantiated (securely) then primitive Q can also be instantiated (securely). Such a relation "relativizes" if the statement holds even if the adversaries against P and Q have access to an arbitrary oracle Π. Although apparently related, the focus and results of that line of work are quite different.

RTV [22] assumed probabilistic polynomial time oracle machines when introducing relativizing reductions. In particular, they asserted that for a primitive with oracle access to exist there must be a PPTOM machine that can compute it and that no PPTOM machine breaks it (see Definition 2 in the full version of [22] or definition 5 of [1]). Lifting does not have that restriction, and is as such a more general and flexible notion. A lifted reduction holds even if the adversary performs a countable number of queries from an infinite query space. Furthermore, we do not require an efficient implementation in the idealized model (see definition 5). The same holds for the security definition – even though as is

[6] Note that this would incorporate the reduction in CSMW as well as some specific, potentially smarter way, of answering RO queries of the adversary against CSMW[BR].

standard we define here security via a PPT game – one may transfer our result to non-polynomial time security settings. As such our tooling and approach can accommodate oracle querying with non-trivial, non-finite, underlying probability distributions.

Note also the difference is one of intent. In relativizing reductions, the security of the primitive under question is not impacted by the choice of oracle the adversary has access to. The reduction must hold *for any* oracle. This of course can rule out such reductions due to oracle separation results. In our work we focus on a particular idealized model and inquire if we can "lift" the security to this idealized model.

A rich line of research, also originating in the seminal work of Rudich and Impagliazzo [18] and continuing with the works of Boneh and Venkatesan [6], Simon [24] and Hsiao and Reyzin [17] has developed a number of black-box separation techniques. These can be used to show negative results of the type: no black-box construction of protocol Q out of primitive P exists, or conversely that no black-box reduction from Q to P exists. Such results are important to rule out minimal assumption for the existence of Q, or identify the need for non-black box constructions but do not serve as support for drawing positive results.

2 Preliminaries

2.1 Notation

For two arbitrary sets X and Y we write Y^X for the set of all functions from X to Y. Let $\mathbb{N} = \{0, 1, 2, 3, \dots, \}$ be the set of natural numbers. We use $\lambda \in \mathbb{N}$ to denote the security parameter, which is a natural number; we assume that it is implicitly provided to all algorithms in the unary representation 1^λ, unless stated otherwise.

We use the shorthand PPT for the Probabilistic Polynomial Time algorithms – in the (unary) security parameter λ. We describe $(y_1, \dots) \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, x_1, \dots; r)$ as an event when $\mathcal{A}$ gets $(1^\lambda, x_1, \dots)$ as input, uses fresh random coins r and outputs $(y_1, \dots)$. If $\mathcal{A}$ is deterministic then we simply write $(y_1, \dots) \leftarrow \mathcal{A}(1^\lambda, x_1, \dots)$. Let us write $\mathcal{A}^B$ to denote that $\mathcal{A}$ has black-box access to algorithm B, meaning it sees only its input-output behaviour. The notation $\mathcal{A}^{(\cdot)}$ means that $\mathcal{A}$ expects a black-box access to some other algorithm. Similarly as in [13], we highlight that when an algorithm $\mathcal{B}$ is given oracle access to $\mathcal{A}^{\mathcal{O}}$ for a particular oracle $\mathcal{O}$ then $\mathcal{B}$ does not get to answer $\mathcal{A}$'s queries to $\mathcal{O}$. Throughout the paper, $\perp$ denotes an error symbol.

For a finite set S, we denote its cardinality by $|S|$ and write $s \leftarrow\!\!\$\ S$ meaning that we choose an element s from S uniformly at random. For readability, we define $[k] = \{1, \dots, k\}$ for $k \in \mathbb{N}$ and $[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}$. Set S is countable if there exists an injective map $\phi : S \to \mathbb{N}$.

A function $\epsilon : \mathbb{N} \to \mathbb{R}$ is negligible if for any $c \in \mathbb{N}$, there exists $N \in \mathbb{N}$ such that for all $\lambda \geq N$: $|\epsilon(\lambda)| < 1/\lambda^c$. We write $\mathsf{negl}(\lambda)$ for an unspecified negligible function in λ. In general, we denote with negl to be a set of all negligible functions.

Similarly, we define UBnegl to be the set of functions $f : \mathbb{N} \to \mathbb{R}$ which are upper-bounded by a negligible function. Concretely, $f \in \mathsf{UBnegl}$ if and only if there exists $\epsilon(\cdot) \in \mathsf{negl}$ such that $f(\lambda) \leq \epsilon(\lambda)$ for all $\lambda \in \mathbb{N}$. We highlight that functions in UBnegl are not necessarily negligible, e.g. the constant function $f(\lambda) = -1$. By definition of a negligible function we obtain the following lemma.

Lemma 1. *Let $f : \mathbb{N} \to \mathbb{R}$ be a function. Then, the following conditions are equivalent.*

1. *$f \in \mathsf{UBnegl}$,*
2. *function $g(\lambda) := \max\{f(\lambda), 0\}$ is negligible,*
3. *for all $c \in \mathbb{N}$, there exists $N \in \mathbb{N}$ such that for all $\lambda \geq N$: $f(\lambda) < 1/\lambda^c$.*

2.2 Limits and Suprema

Let A be a (possibly uncountable) set. Then, for a function $f : A \to \mathbb{R}$ we define the supremum $\sup_{a \in A} f(a)$ to be the smallest real number t (if exists) such that $f(a) \leq t$ for all $a \in A$. In this paper, will use the following simple lemmas. For completeness, we provide the proofs in Appendix A.

Lemma 2. *Let A, S be non-empty sets, where S is either finite or countable, and $(f_s)_{s \in S}$ be a sequence of functions $f_s : A \to \mathbb{R}$. If for all $s \in S$, $\sup_{a \in A} f_s(a)$ exists and if $\sup_{(a_s \in A)_{s \in S}} \sum_{s \in S} f_s(a_s)$ exists then*

$$\sum_{s \in S} \sup_{a \in A} f_s(a) = \sup_{(a_s \in A)_{s \in S}} \sum_{s \in S} f_s(a_s).$$

Lemma 3. *Let A be a non-empty set and $(f_a)_{a \in A}$ be a family of non-decreasing functions $f_a : \mathbb{N} \to \mathbb{R}$. Then:*

$$\lim_{k \to +\infty} \sup_{a \in A} f_a(k) = \sup_{a \in A} \lim_{k \to +\infty} f_a(k)$$

assuming $\sup_{a \in A} \lim_{k \to +\infty} f_a(k)$ and $\lim_{k \to +\infty} f_a(k)$ exist for all $a \in A$.

Lemma 4. *Let $f : \mathbb{N} \times \mathbb{N} \to [0, 1]$ be function such that for all $k, \ell \in \mathbb{N}$: $f(k, \ell) \leq f(k + 1, \ell)$ and $f(k, \ell) \leq f(k, \ell + 1)$. Then, $\lim_{k \to +\infty} \lim_{\ell \to +\infty} f(k, \ell)$ exists and*

$$\lim_{k \to +\infty} \lim_{\ell \to +\infty} f(k, \ell) = \lim_{\ell \to +\infty} \lim_{k \to +\infty} f(k, \ell) = \lim_{k \to +\infty} f(k, k).$$

Lemma 5. *Let S be a non-empty, either finite or countable set and $f : \mathbb{N} \times S \to [0, 1]$ be a function which satisfies $f(k, s) \leq f(k + 1, s)$ and*

$$\sum_{s \in S} f(k, s) \in [0, 1]$$

for all $k \in \mathbb{N}, s \in S$. Then

$$\sum_{s \in S} \lim_{k \to +\infty} f(k, s) = \lim_{k \to +\infty} \sum_{s \in S} f(k, s).$$

2.3 Fully Black-Box Reductions in the Standard Model

We briefly recall the framework on primitives and black-box reductions by Reingold, Trevisan, and Vadhan [22] (RTV). Using their notation, primitive P is a pair $\langle F_P, R_P \rangle$ where F_P is a set of functions $f : \{0,1\}^* \to \{0,1\}^*$ and R_P is a relation over pairs (f, M) for $f \in F_P$ and machine M. One can think of F_P as implementations of a primitive P and R_P as security conditions on F_P.

Then, there is a *fully black-box* reduction from a primitive $P = \langle F_P, R_P \rangle$ to $Q = \langle F_Q, R_Q \rangle$ if there exist PPT machines G, S such that:

- for every function $f \in F_Q$, $G^f \in F_P$,
- for every function $f \in F_Q$ and every adversary $\mathcal{A}$, $(G^f, \mathcal{A}) \in R_P \implies (f, S^{\mathcal{A}}) \in R_Q$.

Informally, G and S are called the *generic construction* and the *reduction* respectively. As mentioned in [22], this definition of reduction does not apply to non-uniform or information-theoretic notions of security. They also define different types of reductions such as semi-black-box or relativizing reductions.

There is a long line of research on formalising (black-box) reductions [1,13,19,16]. In this paper we adapt the recently defined notion of fully black-box reductions by Hofheinz and Nguyen [16]. The main difference to the RTV framework is that the security conditions are represented as a security game instead of a set of relations. Thus, Hofheinz and Nguyen could formally define what is meant by "breaking one primitive with about the same success as the other primitive" in terms of probabilities.

Definition 1 ([16]). *A primitive P is a tuple $\langle \mathbb{P}, F_P, R_P, \sigma \rangle$ where:*

- *$\mathbb{P}$ is a pair of sets (A, B)*
- *F_P is a subset of $\{f : A \to B\}$,*
- *$R_P^{(\cdot,\cdot)}$ is a PPT security algorithm,*
- *$\sigma : \mathbb{N} \to [0,1]$ is a security threshold.*

We say that f is an implementation of P if $f \in F_P$.

Note that usually we define F_P via the use of correctness games.

Since we do not consider primitives in the multi-instance setting as in [16], we already include the setup S_P in the security algorithm R_P. For readability, in this paper we also do not restrict the input space for R_P to call f, i.e. $C = A$ in [16, Definition 4].

There are two main differences between this definition and the one proposed by RTV. Firstly, $\mathbb{P} = (A, B)$ is a pair of sets which describe the domain and the co-domain. This modification enables to characterize implementations which are defined on more abstract mathematical models (e.g. groups, rings) rather than on $\{0,1\}^*$. Secondly, R_P is now an efficient algorithm which expects black-box access to both an implementation f and an adversary $\mathcal{A}$. One can think of R_P as a security game, e.g. one-wayness or IND-CPA game. Here, we want to associate for each pair $(f, \mathcal{A})$ a value in $[0,1]$ which corresponds to the probability of $\mathcal{A}$ winning the R_P game against f. We recall the definition of an advantage from [16].

Definition 2 ([16]). *Let $P = \langle \mathbb{P}, F_P, R_P, \sigma \rangle$ be a primitive. Take $f \in F_P$ and any algorithm $\mathcal{A}$. We define the advantage of $\mathcal{A}$ in breaking f as*

$$\mathsf{Adv}^{\mathrm{P}}_{f,\mathcal{A}}(\lambda) \coloneqq \Pr\left[1 \leftarrow\!\!\$\, R_P^{f,\mathcal{A}}\right] - \sigma(\lambda)$$

where the probability is defined over random coins in the system [7]*.*

We say that $\mathcal{A}$ $P-$breaks f if $\mathsf{Adv}^{\mathrm{P}}_{f,\mathcal{A}}(\lambda) \notin \mathsf{UBnegl}$, i.e. there is no negligible function $\epsilon : \mathbb{N} \to \mathbb{R}$ such that $\mathsf{Adv}^{\mathrm{P}}_{f,\mathcal{A}}(\lambda) \leq \epsilon(\lambda)$ for all $\lambda \in \mathbb{N}$. Primitive P is called secure if there exists an implementation f of P such that there are no PPT algorithms $\mathcal{A}$ that $P-$break f.

Example 1. We define a primitive corresponding to an IND-CPA secure public-key encryption scheme as $\mathsf{PKE} = \langle \mathbb{P}_{\mathsf{PKE}}, F_{\mathsf{PKE}}, R_{\mathsf{PKE}}, \frac{1}{2} \rangle$ where $\mathbb{P}_{\mathsf{PKE}}$ defines the domain and range for the encryption schemes, R_{PKE} is the IND-CPA game and F_{PKE} the set that contains encryption schemes, which we could define via a "encryption scheme correctness" game.

We briefly explain why we want the advantage to be in UBnegl rather than negl. Note that there are certain types of adversaries, for which their advantage is not negligible, and yet they do not win the security game in the usual sense. For instance, consider a decisional game, e.g the IND-CPA game, where the adversary has to guess the bit, and set the security threshold $\sigma(\lambda) = \frac{1}{2}$. Then, an adversary $\mathcal{A}$, which simply aborts/loops, certainly will not win the IND-CPA game (the security game cannot detect $\mathcal{A}$ looping since it is only given black-box access). However, its advantage, as defined in Definition 2, will be $0 - \frac{1}{2} = -\frac{1}{2}$, which is not negligible (but still upper-bounded by a negligible function).

Using the definitions above, Hofheinz and Nguyen formalise fully-black box reductions as follows.

Definition 3 (Fully Black-Box Reductions). *Let $P = \langle \mathbb{P}_1, F_P, R_P, \sigma \rangle$ and $Q = \langle \mathbb{P}_2, F_Q, R_Q, \tau \rangle$ be primitives. Then, there is a fully black-box reduction from P to Q if there exist PPT algorithms $G^{(\cdot)}, S^{(\cdot)}$ such that:*

- *for every implementation f of Q, G^f is an implementation of P,*
- *for every implementation f of Q and every (unbounded) algorithm $\mathcal{A}$, if $\mathcal{A}$ P-breaks G^f then $S^{\mathcal{A}}$ Q-breaks f.*

3 Average of Negligible Functions

In this section we establish several technical properties of negligible functions which will be crucial when proving our main reduction correspondence result. It is a well known that given a finite set of negligible functions $P = \{f_1, f_2, \ldots, f_n\}$, the average $\frac{1}{n}\sum_{i=1}^{n} f_i$ of these functions is also negligible. We provide a similar result in the setting when the set P is countable.

[7] Usually, the security threshold function σ is a constant – either 0 or $\frac{1}{2}$.

Informally, suppose we have a function $P : \mathbb{N} \times \mathbb{N} \to [-1, 1]$ such that for any $f : \mathbb{N} \to \mathbb{N}$, function $P_f(\lambda) := P(\lambda, f(\lambda))$ is negligible in λ. Then, for any discrete distribution $\mathcal{D}$ on $\mathbb{N}$ and an infinite sequence of independent random variables $X_1, X_2, \ldots \leftarrow_{\$} \mathcal{D}$, the function $\mathbb{E}(P) : \mathbb{N} \to [-1, 1]$ defined as

$$\mathbb{E}(P)(\lambda) := \mathbb{E}\left(P(\lambda, X_\lambda)\right)$$

is also negligible. Intuitively, this result says that if a set P [8] consists of only negligible functions then the "expectation of all functions", defined as $\mathbb{E}(P)$ and also called informally as the average of P, is also negligible.

Below we state a generalisation of this result. Roughly speaking and using the language from the previous paragraph, it says the following. Assume there exists a correspondence between negligible functions from set Q to set P. If the expectation of Q is negligible then so is the expectation of P. Clearly, by setting the set Q to only contain the zero functions yields the result described above.

To apply these observations in the context of fully black-box reductions, we work with functions in UBnegl (see Section 2.1) rather than with negligible functions.

Theorem 1. *Let $k \in \mathbb{N}$, S be a (possibly uncountable) set and $(\mathcal{D}_\lambda)_{\lambda \in \mathbb{N}}$ be a sequence of discrete probability distributions $\mathcal{D}_\lambda : S_\lambda \to [0, 1]$ over countable sets $S_\lambda \subseteq S$. Take arbitrary functions $P, Q_1, \ldots, Q_k : \mathbb{N} \times S \to [-1, 1]$. Suppose that for every function $f : \mathbb{N} \to S$, the following holds:*

$$\forall i \in [k], Q_i(\lambda, f(\lambda)) \in \mathsf{UBnegl} \implies P(\lambda, f(\lambda)) \in \mathsf{UBnegl}.$$

Then, for $X_\lambda \leftarrow_{\$} D_\lambda$ we have

$$\forall i \in [k], \mathbb{E}\left(Q_i(\lambda, X_\lambda)\right) \in \mathsf{UBnegl} \implies \mathbb{E}\left(P(\lambda, X_\lambda)\right) \in \mathsf{UBnegl}.$$

Proof. Suppose that each $\mathbb{E}\left(Q_i(\lambda, X_\lambda)\right)$ is upper-bounded by a negligible function. Then, for each $i \in [k]$, we can find an infinite sequence of positive integers $J(i, 1) < J(i, 2) < \ldots$ such that for every $d \in \mathbb{N}$ and any $\lambda \geq J(i, d)$, $\mathbb{E}\left(Q_i(\lambda, X_\lambda)\right) < 1/\lambda^d$. Fix $d \in \mathbb{N}$ and define

$$j_d = \max\{2k + 1, \max_{i \in [k]} J(i, d + 1)\}.$$

We claim that for every $\lambda \geq j_d$, there exists $a \in S_\lambda$ which satisfies:

$$\forall i \in [k], Q_i(\lambda, a) < 1/\lambda^d.$$

First, we fix arbitrary $i \in [k]$ and $\lambda \geq j_d$. Let

$$M_i = \{a : a \in S_\lambda \wedge Q_i(\lambda, a) < 1/\lambda^d\}.$$

We know that $\mathbb{E}\left(Q_i(\lambda, X_\lambda)\right) < 1/\lambda^{d+1}$. Therefore,

$$\frac{\Pr[X_\lambda \in S_\lambda \backslash M_i]}{\lambda^d} \leq \frac{\sum_{a \in S_\lambda \backslash M_i} \Pr[X_\lambda = a]}{\lambda^d} \leq \sum_{a \in S_\lambda} \Pr[X_\lambda = a] \cdot Q_i(\lambda, a) < \frac{1}{\lambda^{d+1}}.$$

[8] Formally, we mean the set of functions $\{P(\lambda, f(\lambda)) : f \in \{g : \mathbb{N} \to \mathbb{N}\}\}$.

In particular, $\Pr[X_\lambda \in S_\lambda \backslash M_i] \leq 1/\lambda$. Then, by the union bound we have:

$$\Pr[\exists i, X_\lambda \in S_\lambda \backslash M_i] \leq k/\lambda \leq 2k/j_d < 1.$$

Hence, $\Pr[\forall i, X_\lambda \in M_i] > 0$ so there exists $a \in S_\lambda$ such that for every $i \in [k], Q_i(\lambda, a) < 1/\lambda^d$. For $\lambda \geq j_d$, let $a(\lambda)$ be the smallest such value.

Next, we prove the following lemma.

Lemma 6. *Let $c \in \mathbb{N}$. Then, there exists a positive integer $d \geq c$, such that there are only finitely many pairs (λ, a) which satisfy the following conditions:*

$$a \in S_\lambda \wedge \forall i, Q_i(\lambda, a) < 1/\lambda^d \wedge P(\lambda, a) \geq 1/\lambda^c. \tag{1}$$

Proof. We prove it by contradiction. Suppose there exists a positive integer c, such that for every $d \geq c$, there are infinitely many pairs (λ, a) which satisfy (1). We construct a function $f : \mathbb{N} \to S$ such that for $i \in [k]$, $Q_i(\lambda, f(\lambda)) \in \mathsf{UBnegl}$ but $P(\lambda, f(\lambda)) \notin \mathsf{UBnegl}$. Then, we get a contradiction.

Fix $d \geq c$. Let us introduce the following notation. First, $L(\ell, d)$ is the smallest $\lambda \geq \ell$ such that there exists an integer a so that (λ, a) satisfies (1). Additionally, denote $R(\ell, d)$ to be the smallest a such that $(L(\ell, d), a)$ satisfies (1). Then, by definition $(L(\ell, d), R(\ell, d))$ satisfy (1). Finally, set $I(c) = j_c$ and $I(d+1) = \max\{j_{d+1}, L(I(d), d) + 1\}$.

We define the function f as follows. For $\lambda < I(c)$, set $f(\lambda) = x$ where x is an arbitrary fixed element in S_λ. Then, for $I(d) \leq \lambda < I(d+1)$, where $d \geq c$, define:

$$f(\lambda) = \begin{cases} R(I(d), d) \text{ if } \lambda = L(I(d), d) \\ a(\lambda) \text{ otherwise.} \end{cases}$$

Recall that $a(\lambda)$ is the smallest value a such that $i \in [k], Q_i(\lambda, a) < 1/\lambda^d$.

We now prove that for each i, $Q_i(\lambda, f(\lambda))$ is upper-bounded by a negligible function. Let $i \in [k]$. By construction, for any $d \in \mathbb{N}$ and $I(d) \leq \lambda < I(d+1)$, we have $Q_i(\lambda, f(\lambda)) < 1/\lambda^d$. Indeed, if $\lambda = L(I(d), d)$ then $(\lambda, f(\lambda))$ satisfies (1). On the other hand, if $\lambda \neq L(I(d), d)$ then since $\lambda \geq I(d) \geq j_d$ we have $Q_i(\lambda, f(\lambda)) = Q_i(\lambda, a(\lambda)) < 1/\lambda^d$.

As a result, for all $\lambda \geq I(d)$ we have $Q_i(\lambda, f(\lambda)) < 1/\lambda^d$. The reason is that for $\lambda \geq I(d)$ there is some $\alpha \geq d$ so that $I(\alpha) \leq \lambda < I(\alpha+1)$. By the observation above, we have $Q_i(\lambda, f(\lambda)) < 1/\lambda^\alpha \leq 1/\lambda^d$. Consequently, $Q_i(\lambda, f(\lambda))$ is upper-bounded by a negligible function.

On the other hand, for all $d \in \mathbb{N}$, we have $P(\lambda, f(\lambda)) \geq 1/\lambda^c$ where $\lambda = L(I(d), d)$. This means that there are infinitely many positive integers λ such that $P(\lambda, f(\lambda)) \geq 1/\lambda^c$. Hence, $P(\lambda, f(\lambda)) \notin \mathsf{UBnegl}$ by Lemma 1. □

Finally, we prove that $\mathbb{E}(P(\lambda, X_\lambda))$ is upper-bounded by a negligible function. Let $c \in \mathbb{N}$ and $c' = c + k + 1$. From Lemma 6 we know that there exists $d \geq c'$ such that there are finitely many pairs (λ, a) satisfying (1). Therefore, there is an integer N, such that for all pairs (λ, a), where $\lambda \geq N$, one of the conditions in (1) does not hold. Now, let $m = \max\{2, j_{2d-1}, N\}$. We claim that for all $\lambda \geq m$, $\mathbb{E}(P(\lambda, X_\lambda)) < 1/\lambda^c$. This would imply that $\mathbb{E}(P(\lambda, X_\lambda)) \in \mathsf{UBnegl}$.

Take any $\lambda \geq m$. Let us compute a lower-bound on $\Pr[X_\lambda \in H : X_\lambda \leftarrow\!\!\$\, D_\lambda]$ where

$$H = \{a \in S_\lambda : \forall i \in [k], Q_i(\lambda, a) < 1/\lambda^d\}.$$

We proceed similarly as before. Let $i \in [k]$. Then, we have $\mathbb{E}\left(Q_i(\lambda, X_\lambda)\right) < 1/\lambda^{2d}$ since $m \geq j_{2d-1}$. Denote $H_i = \{a \in S_\lambda : Q_i(\lambda, a) < 1/\lambda^d\}$. Thus,

$$\frac{\Pr[X_\lambda \in S_\lambda \backslash H_i]}{\lambda^d} \leq \frac{\sum_{a \in S_\lambda \backslash H_i} \Pr[X_\lambda = a]}{\lambda^d} \leq \sum_{a \in S_\lambda} \Pr[X_\lambda = a] \cdot Q_i(\lambda, a) < \frac{1}{\lambda^{2d}}.$$

Therefore, $\Pr[X_\lambda \in S_\lambda \backslash H_i] < 1/\lambda^d$. Hence, by the union bound we get:

$$\Pr[\exists i \in [k], X_\lambda \in S_\lambda \backslash H_i] \leq k/\lambda^d$$

and thus $\Pr[X_\lambda \in H] \geq 1 - k/\lambda^d$.

Note that each pair (λ, a), where $a \in H$, satisfies the first two conditions in (1). Since $\lambda \geq m \geq N$, we get that $P(\lambda, a) < 1/\lambda^{c'}$. Therefore, we can upper-bound $\mathbb{E}\left(P(\lambda, X_\lambda)\right)$ as follows:

$$\begin{aligned}
\mathbb{E}\left(P(\lambda, X_\lambda)\right) &\leq \sum_{a \in S_\lambda} \Pr[X_\lambda = a] \cdot P(\lambda, a) \\
&\leq \sum_{a \in H} \Pr[X_\lambda = a] \cdot P(\lambda, a) + \sum_{a \notin H} \Pr[X_\lambda = a] \cdot P(\lambda, a) \\
&< \sum_{a \in H} \Pr[X_\lambda = a] \cdot \frac{1}{\lambda^{c'}} + \sum_{a \notin H} \Pr[X_\lambda = a] \\
&< \frac{\Pr[X_\lambda \in H]}{\lambda^{c'}} + \Pr[X_\lambda \in S \backslash H] \\
&< 1/\lambda^{c'} + k/\lambda^d \\
&< (k+1)/\lambda^{c+k+1} < 1/\lambda^c.
\end{aligned} \tag{2}$$

Thus, $\mathbb{E}\left(P(\lambda, X_\lambda)\right) \in \mathsf{UBnegl}$. □

4 Setup Assumptions

In this section we formalize, generically, the notion of a setup assumption. Such assumptions are ubiquitous in modern cryptography and include, for instance, popular settings such as the Ideal Cipher model [23], the Common Random String (CRS) [4], the Random Oracle model (ROM) [3]. They allow one to bypass impossibility results or simply yield more efficient schemes.

Before we present our definition, we motivate some of the choices we make. Naively, we could simply attempt to construct a Turing machine that samples a function $X \to Y$ according to some arbitrary distribution, which would encode the expected behaviour of the oracle. The astute reader is soon to notice that several questions arise. How do we pick the domain of the oracle? For example,

in the random oracle model the query domain is $\{0,1\}^*$ which is infinite. What is then our sample space?

We cannot sample eagerly such a function. While one implementation might simply query a small number of polynomial length values in the security parameter λ, we must recall that reductions should also work for unbounded adversaries. Indeed, an unbounded adversary might query the oracle infinitely many times, which raises a conundrum.

4.1 Formal Model for Setup Assumptions

Our formalization is heavily influenced by having to solve the Random Oracle case outlined above. We proceed as follows. We model the use of a random oracle (viewed as an infinite random tape) via a sequence of *finite* setups. Each setup being parameterized by some parameter $\ell \in \mathbb{N}$ – think about this parameter as a restriction on the size of the valid inputs to the random oracle. As ℓ tends to infinity, the setup becomes a better approximation of a random oracle.

For this approach to be meaningful we require a few additional ingredients. We define X_ℓ to be the first ℓ elements of X. Clearly $X_\ell \subseteq X_{\ell+1}$ and X_ℓ is an increasingly better approximation of X, as ℓ grows. Thus, we seek that when ℓ goes to infinity, X_ℓ comes close to X. Here we enforce a total order on X. Since we only consider countable sets X, we model this total order by fixing an arbitrary injection ϕ between X and $\mathbb{N}$; the total order on X is then induced by transporting on X the total order on $\mathbb{N}$.

We may now define a setup assumption as a tuple $(X, Y, \phi, \mathcal{M})$ where X and Y are the domain and range for all possible setup instances. The setup generator $\mathcal{M}$ takes as input the usual security parameter λ and a parameter ℓ as above. For each λ and ℓ, the setup generator defines some distribution on the set of functions with domain X_ℓ and range Y.

More importantly, we demand that the distributions defined by $\mathcal{M}$ are *consistent* across the choices of ℓ. That is, a function sampled from $Y^{X_{\ell+1}}$ according to $\mathcal{M}_{\ell+1,\lambda}$, when restricted to X_ℓ has the same distribution as a function sampled from $\mathcal{M}_{\ell,\lambda}$. The intuition behind this restriction is that the functions output by the setup should "behave" the same on all entries on which they are defined, independent of the size of the domain specified by ℓ.

This requirement is important since, for instance, we do not wish that altering the size of the query space to $X_{\ell+1}$ affects the behavior of participants that only queries the setup with entries from X_ℓ. As it will become clear a bit later in the paper (Definition 5), this property is necessary to meaningfully define the adversary advantage when ℓ goes to infinity.

The following definition formalizes the discussion above.

Definition 4 (Setup Assumptions). *Define setup assumption as a tuple $M = (X, Y, \phi, \mathcal{M})$ where X, Y are non-empty countable sets, ϕ is an injective map from X to $\mathbb{N}$ and $\mathcal{M}_{(\cdot,\cdot)}$ is a probabilistic algorithm with the following properties. Namely, given a "length" parameter $\ell \in \mathbb{N}$ and a security parameter*

$\lambda \in \mathbb{N}$, *it outputs a function* $\mathcal{O} : X_\ell \to Y$ *according to some distribution over* Y^{X_ℓ}, *where*

$$X_\ell := \{x \in X : \phi(x) < \ell\}.$$

Note that this distribution is still discrete. We further call the setup assumption consistent, if for all $\ell \in \mathbb{N}$ *and* $a_1, \ldots, a_{|X_\ell|} \in Y$ *we have:*

$$\Pr\left[\bigwedge_{i=1}^{|X_\ell|} f(x_{\ell,i}) = a_i : f \leftarrow\!\!\$\, \mathcal{M}_{\ell,\lambda}\right] = \Pr\left[\bigwedge_{i=1}^{|X_\ell|} g(x_{\ell,i}) = a_i : g \leftarrow\!\!\$\, \mathcal{M}_{\ell+1,\lambda}\right]$$

where $X_\ell = \{x_{\ell,1}, \ldots, x_{\ell,|X_\ell|}\}$.

Henceforth, we shall simply refer to consistent sampling setup assumptions simply as setup assumption. When working with primitives in the standard model, we will abuse the notation and write $M = \varnothing$.

4.2 Defining Primitives with Setup Assumptions

We build on our notion of a setup assumption defined in the previous section to formalize models for primitives with setup assumptions.

Before we proceed, we introduce the following notation. Suppose the sampler $\mathcal{M}_{(\cdot,\cdot)}$ of M samples a function $\mathcal{O} : X \to Y$ where X and Y are countable and let $\phi : X \to \mathbb{N}$ be a fixed injective map. Then, for an algorithm $\mathcal{A}^{(\cdot)}$ and function $t : \mathbb{N} \to \mathbb{N}$, we denote $\mathcal{A}_t^{(\cdot)}$ to be the algorithm which behaves identically as $\mathcal{A}^{(\cdot)}$ but if a query $x \in X$ is made to the setup assumption, where $\phi(x) > t(\lambda)$, then it automatically aborts. Recall ϕ is our total ordering function. We call t the threshold function. When t is constant, i.e. $t(\lambda) = \ell$ for all λ, then we slightly abuse notation and simply write $\mathcal{A}_\ell^{(\cdot)}$.

We now present the notion of a primitive equipped with a setup assumption. The definition below, refines Definition 1 in two different ways. It introduces as part of the execution model the setup generator and it introduces an explicit correctness notion for the primitive as an additional separate algorithm.

Definition 5. *Primitive* P *with a setup assumption* M *is a tuple* $\langle \mathbb{P}, M, C_P, R_P, \sigma \rangle$ *where:*

- $\mathbb{P}$ *is a pair of sets* (A, B),
- $M = (X, Y, \phi, \mathcal{M})$ *is the setup assumption defining the oracle* $\mathcal{O}$,
- $C_P^{(\cdot,\cdot)}$ *is a correctness algorithm,*
- $R_P^{(\cdot,\cdot)}$ *is a PPT security algorithm (related to* λ*),*
- $\sigma : \mathbb{N} \to [0, 1]$ *is a security threshold.*

We say that $f^{(\cdot)} : A \to B$ *is an implementation of* P *if for all (unbounded) adversaries* $\mathcal{A}^{(\cdot)}$*:*

$$\lim_{k \to +\infty} \lim_{\ell \to +\infty} \Pr\left[1 \leftarrow C_P^{f_k{}^{\mathcal{O}}, \mathcal{A}_\ell^{\mathcal{O}}}\right]$$

is negligible, where probability is over random coins in the environment and especially $\mathcal{O} \leftarrow_{\$} \mathcal{M}_{\max\{k,\ell\},\lambda}$.

For an implementation $f^{(\cdot)}$ *and any algorithm* $\mathcal{A}^{(\cdot)}$*, we define the advantage of* $\mathcal{A}^{(\cdot)}$ *in breaking* $f^{(\cdot)}$ *as*

$$\mathsf{Adv}^{\mathrm{P}}_{f,\mathcal{A}}(\lambda) \coloneqq \lim_{k\to+\infty} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} R_P^{f_k^{\mathcal{O}},\mathcal{A}_\ell^{\mathcal{O}}}\right] - \sigma(\lambda)$$

where the probability is defined over $\mathcal{O} \leftarrow_{\$} M(\max\{k,\ell\},\lambda)$ *and the random coins in the system.*

We say that $\mathcal{A}$ $P-$*breaks* $f^{(\cdot)}$ *if* $\mathsf{Adv}^{\mathrm{P}}_{\mathcal{O}\leftarrow M,f,\mathcal{A}}(\lambda) \notin \mathsf{UBnegl}$*. Furthermore,* $f^{(\cdot)}$ *is called a secure implementation of* P *if there are no PPT algorithms* $\mathcal{A}^{(\cdot)}$ *that* $P-$*break* $f^{(\cdot)}$*.*

A few remarks are in order. First, we argue that the notion of correctness and the adversary advantage are well-defined, in that the limits are guaranteed to exist. This property is established by the following lemma. Its proof (in Appendix A) crucially relies on the consistency property of the setup assumption. Broadly speaking, the property guarantees that the behavior of an adversary (in terms of winning the security game) is monotonic with respect to ℓ. That is, if an adversary wins the game when its query space is X_ℓ (with some probability), then the adversary will win (with at least the same probability) the instance of the game where the query space is $X_{\ell+1}$. This property then gives rise to a monotonically non-decreasing sequence upper-bounded by 1, which implies that the desired limit exists.

Lemma 7. *Let* $f^{(\cdot)}$*,* $\mathcal{A}^{(\cdot)}$ *and* $\mathcal{R}^{(\cdot,\cdot)}$ *be any function, unbounded adversary and PPT machine respectively. Then, for* $M = (X, Y, \phi, \mathcal{M})$*, the following limit exists:*

$$\lim_{k\to+\infty} \lim_{\ell\to+\infty} F(k,\ell),$$

where

$$F(k,\ell) := \Pr\left[1 \leftarrow_{\$} \mathcal{R}^{f_k{}^{\mathcal{O}},\mathcal{A}_\ell^{\mathcal{O}}}\right]$$

and the probability is over $\mathcal{O} \leftarrow_{\$} \mathcal{M}_{\max\{k,\ell\},\lambda}$ *and the random coins in the environment.*

The proof of the lemma is presented in Appendix A.5.

Second, the limits in the definition of a correct implementation can be swapped or merged into a single parameter, as hinted at in the introduction, by Lemma 4. Nonetheless, we prefer to keep the present formulation since it is particularly helpful for proving our main theorem (Theorem 3).

Finally, our definition no longer describes implementations of a primitive as functions from some abstract implementation set. Instead, we identify correct implementations as those for which no efficient adversary can win a correctness game C_P [9]. The reason for this departure is that we need to formalize what is the

[9] One side effect of this change is that Definition 5 does not cover a number of potential oddities which can be represented using previous frameworks [16,22], e.g. a

extension of a notion P to a setup assumption. Indeed, above we have essentially shown how to define a game with an abstract setup assumption. Lifting a notion from the standard model to some particular setup comes down to simply replacing the setup assumption M (which is $\varnothing$ for the standard model), appropriately.

4.3 Fully Black-Box Non-Programmable Reductions

We now introduce a notion of a fully black-box non-programmable reduction between primitives with setup assumptions.

Definition 6. *Let P and Q be primitives with the setup assumption M. We say that there is a fully black-box non-programmable reduction from P to Q in M (written as $P \overset{M}{\hookrightarrow} Q$) if there exist PPT algorithms $G^{(\cdot)}, S^{(\cdot)}$ such that:*

- *for every implementation $f^{(\cdot)}$ of Q, $G^{f^{(\cdot)}}$ is an implementation of P,*
- *for every implementation $f^{(\cdot)}$ of Q and every (unbounded) algorithm $\mathcal{A}^{(\cdot)}$, if $\mathcal{A}^{(\cdot)}$ P-breaks $G^{f^{(\cdot)}}$ then $S^{\mathcal{A}^{(\cdot)}}$ Q-breaks $f^{(\cdot)}$.*

In the literature, S has access to an external oracle $\mathcal{O}$ instead of $\mathcal{A}$. We call this reduction *non-programmable* since we let $\mathcal{A}$ have access to $\mathcal{O}$ via S, meaning that if the adversary wants to query $\mathcal{O}$, it sends the value to S, S passes it to $\mathcal{O}$ and returns to $\mathcal{A}$ what it got from $\mathcal{O}$. Apart from that, S does not query $\mathcal{O}$ at all. From the perspective of $\mathcal{A}$, this is clearly equivalent to $\mathcal{A}$ having access to $\mathcal{O}$, as illustrated in the definition above. There is another type of reduction called *programmable* [13], where S can simulate an oracle on its own. However, we omit the details in this paper.

4.4 Setup Assumption Extensions

In order to describe our main result, we need to define what it means to "naturally extend" the primitive to a setup assumption M. Hence, we define a notion of a $M-$extension of a primitive.

Definition 7. *Let $P = \langle \mathbb{P}_1, \varnothing, C_P, R_P, \sigma \rangle$ be a primitive in the standard model and M be a setup assumption. Then, a $M-$extension $P(M)$ of P is the tuple $P(M) = \langle \mathbb{P}_1, M, C_P, R_P, \sigma \rangle$.*

We can now formalize our main result. Namely, if there exists a fully black-box reduction from P to Q in the standard model, then there also exists a fully black-box reduction from $P(M)$ to $Q(M)$, where M is any setup assumption.

Theorem 2 (Ideal Model Correspondence). *Let P and Q be primitives in the standard model and M be any ideal model. Then, assuming a fully black-box reduction in the standard model implies a fully black-box reduction in the ideal model.*

$$P \hookrightarrow Q \implies P(M) \overset{M}{\hookrightarrow} Q(M).$$

We provide the proof in Section 5.

---

primitive where the set of valid instances is defined as some undecidable set of Turing machines. However, these cases are irrelevant for our purpose.

Remark. Let (G, S) be a reduction from P to Q. Intuitively, (G, S) should also be a correct reduction from $P(M)$ to $Q(M)$. However, in the M model, the adversary as well as the implementation have access to some "shared state" which is the external oracle. This, however, is not the case in the standard model. Indeed, this additional advice might help an adversary break $P(M)$ but not $Q(M)$. In Theorem 2 we show that if (G, S) is a *fully black-box* reduction, then it can be extended to a setup assumption representing some ideal model. However, the open question remains whether the same property holds when (G, S) is not fully black-box anymore.

$$\begin{array}{ccc} \mathcal{A}\ P-\text{breaks}\ G^f & \hookleftarrow_R \longrightarrow & \mathcal{S}^{f,\mathcal{A}}\ Q-\text{breaks}\ f \\ \downarrow & & \downarrow \\ \mathcal{B}^{\mathcal{O}}\ P-\text{breaks}\ G^{f^{\mathcal{O}}} & \hookleftarrow_R \rightarrow & \mathcal{S}^{f^{\mathcal{O}},\mathcal{B}^{\mathcal{O}}}\ Q-\text{breaks}\ f^{\mathcal{O}} \end{array}$$

Diagram 1: We prove a correspondence for fully black-box reductions in the standard model to ideal models.

4.5 Common Instantiations of Setup Assumptions

In this section we present common ideal models in the framework we introduced. We also prove they satisfy the consistent sampling property.

The Random Oracle Model. This model [3] is one of the cornerstones of modern cryptography. A random oracle represents ideal hash functions. When a party queries with a bitstring $\{0,1\}^*$, the random oracle, given a security parameter λ, samples an element from $\{0,1\}^\lambda$ uniformly at random.

Formally, we define a random oracle setup assumption M_{ROM} as the tuple

$$M_{\text{ROM}} = \left(\{0,1\}^*, \{0,1\}^*, \phi, \mathcal{M}^{\text{ROM}}_{\ell,\lambda}\right)$$

Note we can well-order all bit-strings, for instance as follows: $0, 1, 00, 01, \cdot$. Here, ϕ simply outputs the index of the ordering above. Now define Y to be a set of arbitrary bit-strings with cardinality equal to some polynomial of λ. That is $Y_\lambda = \{0,1\}^{\ell_{\text{out}}(\lambda)}$, for some length function ℓ_{out} – see [8]. Then $\mathcal{M}^{\text{ROM}}_{\ell,\lambda}$ samples from $Y_\lambda^{X_\ell}$ which has cardinality $2^{\ell_{\text{out}}(\lambda)\ell}$. Thus the sampler $\mathcal{M}^{\text{ROM}}_{\ell,\lambda}$ iterates over all ℓ inputs at setup, and for each one picks independently uniformly at random with probability $\frac{1}{2^{\ell_{\text{out}}(\lambda)}}$ an element from Y.

Proposition 1. *The above construction M_{ROM} is a consistent setup assumption.*

Proof. We can easily see that on each instantiation of the setup the sampler picks each element independently at random. Thus, the sampler does not depend on elements ordered after i to pick the value of i. □

The Ideal Cipher Model. In the Ideal Cipher Model [23] the participants may access an ideal cipher $\mathsf{enc} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, s.t. enc are random permutations $\{0,1\}^n \to \{0,1\}^n$ that have been independently and uniformly drawn (with replacement for each key). Recall that the Ideal Cipher model is equivalent to the Random Oracle model [11] (original [12]).

We can define the setup assumption for an (k,n)-ideal cipher similarly to ROM above.

$$M_{IC} = \left(\{0,1\}^* \times \{0,1\}^*, \{0,1\}^*, \phi : \{0,1\}^* \times \{0,1\}^* \to \mathbb{N}, \mathcal{M}^{\mathrm{IC}}_{\ell,\lambda}\right).$$

For simplicity we just set k above equal to λ. Note that here we use the normal ordering as described prior of the bitstrings to the naturals ϕ. We order strings similarly. This implies a 2^n period on the domain set: each of the 2^λ keys is paired with 2^n input values. In particular, note that the sampler $\mathcal{M}^{\mathrm{IC}}_{\ell,\lambda}$ has to pick a permutation $\mathsf{enc}(\mathrm{key}) \colon \{0,1\}^n \to \{0,1\}^n$ independently for each key. Note that n might depend on the security parameter λ.

Proposition 2. *The above setup assumption M_{IC} satisfies the consistent sampling property.*

Proof. Assume the normal ordering as discussed above. First for each new key (of the 2^k) we sample a new random permutation. Thus, we need only to show that while sampling a permutation for $0 \leq \ell < 2^n$ the consistent sampling property holds. We can generalize for each of the keyed permutations. Without loss of generality, if $\ell < 2^n - 1$, observe that the sampling process of the ℓ'th query element of $X_{\ell+1}$ does not depend on the $\ell + 1$ value (it depends only on some of the elements of $X_{\ell-1}$ (if $\ell > 0$) – as we sample without replacement. □

The Common Reference String model (CRS) In the Common Reference String model, a generalization of the Common Random String [4] the oracle provides access to a common value that is sampled from some arbitrary desired distribution specific to the protocol. Namely, following the definition of [7] on setup the oracle samples $d \leftarrow\!\!\$\, \mathcal{D}_\lambda$ and sends it to the querying party. For each subsequent query the oracle responds with d.

Formally, for a CRS model with distribution $\mathcal{D}_\lambda$ over a countable set D, we define the following setup assumption M_{CRS}

$$M_{\mathrm{CRS}} = \left(\{0\}, D, \phi : \{0\} \to \mathbb{N}, \mathcal{M}^{\mathrm{CRS}}_{\ell,\lambda}\right)$$

We simply define $\phi(0) = 0$ and define the sampler $\mathcal{M}^{\mathrm{CRS}}_{\ell,\lambda}$ to sample d from $\mathcal{D}_\lambda$ and return $\mathcal{O} : 0 \mapsto d$.

Proposition 3. *The above setup assumption M_{CRS} satisfies the consistent sampling property.*

Proof. It follows immediately from the observation that the sampling process is independent of the parameter ℓ.

□

5 Proof of Theorem 2

We prove our main claim via Theorems 3 and 4. Namely, we show that if (G, S) is a fully black-box reduction from primitives P to Q in the standard model then (i) G is a generic construction in the setup assumption M and (ii) for every implementation $f^{(\cdot)}$ of $Q(M)$ and every (unbounded) algorithm $\mathcal{A}^{(\cdot)}$, if $\mathcal{A}^{(\cdot)}$ $P(M)$-breaks $G^{f^{(\cdot)}}$ then $S^{\mathcal{A}^{(\cdot)}}$ $Q(M)$-breaks $f^{(\cdot)}$.

5.1 Generic Construction Theorem

We first prove a vital lemma about satisfying correctness with functions with access to a bounded oracle random tape.

Lemma 8. *Let P be a primitive in the setup assumption $M = (X, Y, \phi, \mathcal{M})$, $f^{(\cdot)}$ be a function and OA be the set of all unbounded adversaries with oracle access. Then, $f^{(\cdot)}$ is an implementation of P if and only if $\mathsf{Corr}_P(f)$ is negligible, where:*

$$\mathsf{Corr}_P(f)(\lambda) = \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{k\to+\infty} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow\!\!\$\, C_P^{f_k^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right]$$

and $\mathcal{O} \leftarrow\!\!\$\, \mathcal{M}_{\max\{k,\ell\},\lambda}$.

Proof. Clearly, if $\mathsf{Corr}_P(f)(\lambda)$ is negligible then for any adversary $\mathcal{A}$ we have

$$\lim_{k\to+\infty} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow\!\!\$\, C_P^{f_k^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right] \leq \mathsf{Corr}_P(f)(\lambda).$$

Thus, $f^{(\cdot)}$ is an implementation of P.

Now, suppose that $f^{(\cdot)}$ is an implementation of P. Note that by definition of supremum, we can find a sequence of adversaries $\mathcal{A}_1, \mathcal{A}_2, \ldots$ indexed by λ such that for all λ:

$$\mathsf{Corr}_P(f)(\lambda) \leq \lim_{k\to+\infty} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow\!\!\$\, C_P^{f_k^{\mathcal{O}}(\lambda),\mathcal{A}_{\lambda,\ell}^{\mathcal{O}}(\lambda)}\right] + \frac{1}{2^\lambda}.$$

Hence, let us pick an adversary $\mathcal{A}$ which given λ runs $\mathcal{A}_\lambda$. Since $f^{(\cdot)}$ is an implementation of P, we know that $\lim_{k\to+\infty} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow\!\!\$\, C_P^{f_k^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right]$ is negligible and therefore so is $\mathsf{Corr}_P(f)(\lambda)$. □

For convenience, we continue using OA to denote the set of all unbounded adversaries with oracle access henceforth.

Theorem 3. *Let (G, S) be a fully black-box reduction from P to Q in the standard model and $M = (X, Y, \phi, \mathcal{M})$ be a setup assumption. Then, for every implementation $f^{(\cdot)}$ of $Q(M)$, $G^{f^{(\cdot)}}$ is an implementation of $P(M)$.*

Proof. Let us first fix $\lambda \in \mathbb{N}$ and $f^{(\cdot)}$ be an implementation of $Q(M)$. We first prove that there exists a function $t : \mathbb{N} \to \mathbb{N}$ such that f_t is also an implementation of $Q(M)$ and if $G^{f_t^{(\cdot)}}$ is an implementation of $P(M)$ then so is $G^{f^{(\cdot)}}$.

Lemma 9. *For any function $f^{(\cdot)}$, there exists $t : \mathbb{N} \to \mathbb{N}$ which satisfies the following properties:*

- *$f^{(\cdot)}$ is an implementation of $Q(M)$ if and only if $f_t^{(\cdot)}$ is an implementation of $Q(M)$.*
- *$G^{f^{(\cdot)}}$ is an implementation of $P(M)$ if and only if $G^{f_t^{(\cdot)}}$ is an implementation of $P(M)$.*

Proof. We prove the statement by construction. Let $f^{(\cdot)}$ be any function and $\lambda \in \mathbb{N}$. Then, by Lemma 3 we have

$$\begin{aligned}\mathsf{Corr}_{Q(M)}(f)(\lambda) &= \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{k\to+\infty} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_P^{f_k^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right] \\ &= \lim_{k\to+\infty} \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_P^{f_k^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right] \qquad (3) \\ &= \lim_{k\to+\infty} c_k\end{aligned}$$

where

$$c_k = \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_P^{f_k^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right].$$

Therefore, there exists an integer N_1 such that for all $n \geq N_1$

$$|c_n - \mathsf{Corr}_{Q(M)}(f)(\lambda)| < \frac{1}{2^\lambda}.$$

Note that

$$\begin{aligned}c_n &= \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_P^{f_n^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right] \\ &= \lim_{k\to+\infty} \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_P^{f_{\min\{n,k\}}^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right] \qquad (4) \\ &= \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{k\to+\infty} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_P^{f_{\min\{n,k\}}^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right] = \mathsf{Corr}_{Q(M)}(f_n)(\lambda).\end{aligned}$$

Similarly, one can find $N_2 \in \mathbb{N}$ such that for all $n \geq N_2$:

$$|\mathsf{Corr}_{P(M)}\left(G^{f_n}\right)(\lambda) - \mathsf{Corr}_{P(M)}\left(G^{f}\right)(\lambda)| < \frac{1}{2^\lambda}.$$

Let us set $t(\lambda) := \max\{N_1, N_2\}$. Then, the statement holds by construction and Lemma 8. □

Next, we select t as in the lemma above and define the "relevant tape" as a sequence of sets $T_1, T_2, \ldots$ defined as:

$$T_\lambda = X_{t(\lambda)} = \{x \in X : \phi_\lambda(x) < t(\lambda)\}.$$

For simplicity, we index T_λ as follows $T_\lambda = \{x_{\lambda,1}, \ldots, x_{\lambda,|T_\lambda|}\}$. Define:

$$S_\lambda = Y^{|T_\lambda|} \text{ and } S = \bigcup_{\lambda\in\mathbb{N}} S_\lambda.$$

Then, we set the distribution $D_\lambda : S_\lambda \to [0,1]$ as

$$D_\lambda(y_1, \ldots, y_{|T_\lambda|}) := \Pr\left[\forall i \in [|T_\lambda|], \mathcal{O}(x_{\lambda,i}) = y_i : \mathcal{O} \leftarrow\!\!\$\, \mathcal{M}\left(t(\lambda), \lambda\right)\right].$$

Since T_λ is finite, the distribution is discrete. Moreover, by consistency of setup assumptions we get that for all $\ell \geq t(\lambda)$:

$$D_\lambda(y_1, \ldots, y_{|T_\lambda|}) = \Pr\left[\forall i \in [|T_\lambda|], \mathcal{O}(x_{\lambda,i}) = y_i : \mathcal{O} \leftarrow\!\!\$\, \mathcal{M}\left(\ell, \lambda\right)\right].$$

Now, we define

$$Q(\lambda, \vec{y}) = \begin{cases} \sup_{\mathcal{A} \in \mathsf{OA}} \lim_{\ell \to +\infty} \Pr\left[1 \leftarrow\!\!\$\, C_Q^{f_t^{\mathcal{O}}(\lambda), \mathcal{A}_\ell^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] & \text{if } \vec{y} \in S_\lambda \\ 0 \text{ otherwise} \end{cases}$$

and similarly

$$P(\lambda, \vec{y}) = \begin{cases} \sup_{\mathcal{A} \in \mathsf{OA}} \lim_{\ell \to +\infty} \Pr\left[1 \leftarrow\!\!\$\, C_P^{G^{f_t^{\mathcal{O}}}(\lambda), \mathcal{A}_\ell^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] & \text{if } \vec{y} \in S_\lambda \\ 0 \text{ otherwise} \end{cases}.$$

Here, the probabilities are defined over $\mathcal{O} \leftarrow\!\!\$\, \mathcal{M}_{\max\{\ell, t(\lambda)\}, \lambda}$ and the random coins in the system. One argues similarly as in Lemma 7 that functions P and Q are well-defined.

We claim that for any $g : \mathbb{N} \to S$, we have

$$Q(\lambda, g(\lambda)) \in \mathsf{UBnegl} \implies P(\lambda, g(\lambda)) \in \mathsf{UBnegl}.$$

Indeed, suppose that $Q(\lambda, g(\lambda)) \in \mathsf{UBnegl}$. By construction, we have $0 \leq Q(\lambda, g(\lambda)) \leq 1$ for all $\lambda \in \mathbb{N}$ and thus this function is negligible.

We define the function f_t^g with hardwired oracle queries g as follows. Given a security parameter λ, it behaves identically as in $f_t^{(\cdot)}$ but when f "queries an oracle" on input $x_{\lambda,i} \in T_\lambda$, it gets y_i where $g(\lambda) = (y_1, \ldots, y_{|T_\lambda|}) \in S_\lambda$. On the other hand, for λ so that $g(\lambda) \notin S_\lambda$, we set f_t^g to simply abort.

By construction, $\mathsf{Corr}_Q(f_t^g) = Q(\lambda, g(\lambda))$ is negligible and therefore, f_t^g is an implementation of Q. Since G is a generic construction in the standard model, we have that $G^{f_t^g}$ is an implementation of P, i.e. $\mathsf{Corr}_Q(G^{f_t^g})$ is negligible. As a consequence, $P(\lambda, g(\lambda))$ is negligible and in particular $P(\lambda, g(\lambda)) \in \mathsf{UBnegl}$.

We are now ready to apply Theorem 1 for $k = 1$. Note that for $Z_\lambda \leftarrow\!\!\$\, D_\lambda$ we have

$$\begin{aligned} &\mathbb{E}(Q(\lambda, Z_\lambda)) \\ &\quad = \sum_{\vec{y} \in S_\lambda} Q(\lambda, \vec{y}) \cdot \Pr[Z_\lambda = \vec{y}] \\ &\quad = \sum_{\vec{y} \in S_\lambda} \sup_{\mathcal{A} \in \mathsf{OA}} \lim_{\ell \to +\infty} \Pr\left[1 \leftarrow\!\!\$\, C_Q^{f_t^{\mathcal{O}}(\lambda), \mathcal{A}_\ell^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] \cdot \Pr[Z_\lambda = \vec{y}] \end{aligned} \tag{5}$$

which, by Lemma 2, is equal to

$$\sup_{(\mathcal{A}_{\vec{y}})\in\mathsf{OA}^{|S_\lambda|}} \sum_{\vec{y}\in S_\lambda} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_{\vec{y},\ell}^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] \cdot \Pr[Z_\lambda = \vec{y}].$$

We claim that

$$\mathbb{E}(Q(\lambda, Z_\lambda)) = \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right].$$

First, take any unbounded adversary $\mathcal{A} \in \mathsf{OA}$. Then, by Lemma 5:

$$\lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right] \tag{6}$$

$$= \lim_{\ell\to+\infty} \sum_{\vec{y}\in S_\lambda} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] \cdot \Pr[Z_\lambda = \vec{y}] \tag{7}$$

$$= \sum_{\vec{y}\in S_\lambda} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] \cdot \Pr[Z_\lambda = \vec{y}] \tag{8}$$

$$\leq \sum_{\vec{y}\in S_\lambda} \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] \cdot \Pr[Z_\lambda = \vec{y}] \tag{9}$$

$$\leq \mathbb{E}(Q(\lambda, Z_\lambda)). \tag{10}$$

Then, by definition of supremum we have

$$\mathbb{E}(Q(\lambda, Z_\lambda)) \geq \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right].$$

On the other hand, let us select any sequence of adversaries $(\mathcal{A}_{\vec{y}})_{\vec{y}\in S_\lambda}$. We construct an adversary $\mathcal{A}$ which first calls the external oracle $\mathcal{O}$ on all inputs in T_λ and given $\vec{y} = (y_1, \ldots, y_{|T_\lambda|})$, where $\mathcal{O}(x_{\lambda,i}) = y_i$, it runs $\mathcal{A}_{\vec{y}}$. Then, we have

$$\sum_{\vec{y}\in S_\lambda} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_{\vec{y},\ell}^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] \cdot \Pr[Z_\lambda = \vec{y}] \tag{11}$$

$$= \lim_{\ell\to+\infty} \sum_{\vec{y}\in S_\lambda} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_{\vec{y},\ell}^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] \cdot \Pr[Z_\lambda = \vec{y}] \tag{12}$$

$$= \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}_\ell^{\mathcal{O}}(\lambda)}\right]. \tag{13}$$

Consequently, $\mathbb{E}(Q(\lambda, Z_\lambda)) \leq \sup_{\mathcal{A}\in\mathsf{OA}} \lim_{\ell\to+\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda),\mathcal{A}^{\mathcal{O}}(\lambda)}\right]$ and the claim holds. In particular,

$$\mathbb{E}(Q(\lambda, Z_\lambda)) = \mathsf{Corr}_{Q(M)}(f_t)(\lambda) \in \mathsf{UBnegl}.$$

Thus, by Theorem 1, $\mathbb{E}(P(\lambda, Z_\lambda)) \in \mathsf{UBnegl}$. Note that $\mathbb{E}(P(\lambda, Z_\lambda)) \in [0,1]$ for all λ, and consequently this function is also negligible. By arguing similarly as before, we get that $\mathbb{E}(P(\lambda, Z_\lambda)) = \mathsf{Corr}_{Q(M)}(G^{f_t})(\lambda)$ and thus, G^f is an implementation of $P(M)$ by Lemmas 8 and 9. □

5.2 Reduction Theorem

Theorem 4. *Let (G, S) be a fully black-box reduction from P to Q in the standard model and M be an external oracle. Then, for every implementation $f^{(\cdot)}$ of $Q(M)$ and every adversary $\mathcal{A}^{(\cdot)}$, if $\mathcal{A}^{(\cdot)}$ $P(M)$-breaks $G^{f^{(\cdot)}}$ then $S^{\mathcal{A}^{(\cdot)}}$ $Q(M)$-breaks $f^{(\cdot)}$.*

Proof. We prove the statement by contrapositive. First, we will need an extension of Lemma 9.

Lemma 10. *For any implementation $f^{(\cdot)}$ of $Q(M)$ and adversary $\mathcal{A}^{(\cdot)}$, there exists $t : \mathbb{N} \to \mathbb{N}$ such that for all $\lambda \in \mathbb{N}$:*

- *$f_t^{(\cdot)}$ is an implementation of $Q(M)$.*
- *$|\mathsf{Adv}^{\mathrm{P(M)}}_{G^f,\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathrm{P(M)}}_{G^{f_t},\mathcal{A}_t}(\lambda)| < \frac{1}{2^\lambda}$.*
- *$|\mathsf{Adv}^{\mathrm{Q(M)}}_{f,S^{\mathcal{A}}}(\lambda) - \mathsf{Adv}^{\mathrm{Q(M)}}_{f_t,S^{\mathcal{A}_t}}(\lambda)| < \frac{1}{2^\lambda}$.*

Proof. We prove the statement by construction. Let $f^{(\cdot)}$ and $\mathcal{A}^{(\cdot)}$ be any implementation of $Q(M)$ and adversary respectively, and $\lambda \in \mathbb{N}$. First, the proof of Lemma 9 says that there exists $N_0 \in \mathbb{N}$ such that for all $n \geq N_0$:

$$|\mathsf{Corr}_{Q(M)}(f_n)(\lambda) - \mathsf{Corr}_{Q(M)}(f)(\lambda)| < \frac{1}{2^\lambda}.$$

On the other hand, by definition of the advantage and Lemma 4

$$\begin{aligned} \mathsf{Adv}^{\mathrm{Q(M)}}_{f,S^{\mathcal{A}}}(\lambda) &= \lim_{k\to+\infty} \Pr\left[1 \leftarrow\!\!\$\, R_Q^{f_k^{\mathcal{O}}, S^{\mathcal{A}_k^{\mathcal{O}}}}\right] - \sigma_Q(\lambda) \\ &= \lim_{k\to+\infty} c_k \end{aligned} \tag{14}$$

where $\sigma_Q(\lambda)$ is the security threshold for Q and

$$c_k = \Pr\left[1 \leftarrow\!\!\$\, R_Q^{f_k^{\mathcal{O}}, S^{\mathcal{A}_k^{\mathcal{O}}}}\right] - \sigma_Q(\lambda) = \mathsf{Adv}^{\mathrm{Q(M)}}_{f_k,S^{\mathcal{A}_k}}(\lambda)$$

for $k \in \mathbb{N}$. This means there exists $N_1 \in \mathbb{N}$ such that for all $n \geq N_1$:

$$|c_n - \mathsf{Adv}^{\mathrm{Q(M)}}_{f,S^{\mathcal{A}}}(\lambda)| < 1/2^\lambda.$$

One can similarly compute such N_2 for G^f. Let us set $t(\lambda) = \max\{N_0, N_1, N_2\}$. Then, the statement holds by construction. □

Fix an implementation $f^{(\cdot)}$ of $Q(M)$ and adversary $\mathcal{A}^{(\cdot)}$ such that $\mathsf{Adv}^{\mathrm{Q(M)}}_{f,S^{\mathcal{A}}}(\lambda) \in \mathsf{UBnegl}$ is bounded by a negligible function. Let us select t as in the lemma above. Then, $f_t^{(\cdot)}$ is an implementation of $Q(M)$ and $\mathsf{Adv}^{\mathrm{Q(M)}}_{f_t,S^{\mathcal{A}_t}}(\lambda) \in \mathsf{UBnegl}$ as well.

Define the "relevant tape" as a sequence of sets $T_1, T_2, \ldots$ as:

$$T_\lambda = X_{t(\lambda)} = \{x \in \mathcal{X}_\lambda : \phi_\lambda(x) \leq t(\lambda)\}.$$

For simplicity, we write $T_\lambda = \{x_{\lambda,1}, \ldots, x_{\lambda,|T_\lambda|}\}$. Denote

$$S_\lambda = \mathcal{Y}_\lambda^{|T_\lambda|} \text{ and } S = \bigcup_{\lambda \in \mathbb{N}} S_\lambda.$$

Then, we define the distribution $D_\lambda : S_\lambda \to [0,1]$ as

$$D_\lambda(y_1, \ldots, y_{|T_\lambda|}) := \Pr[\forall i \in [|T_\lambda|], \mathcal{O}(x_{\lambda,i}) = y_i : \mathcal{O} \leftarrow_{\$} \mathcal{M}_{t(\lambda),\lambda}].$$

Since $|T_\lambda|$ is finite, the distribution is discrete. As before, consistency of a setup assumption implies that for all $\ell \geq t(\lambda)$:

$$D_\lambda(y_1, \ldots, y_{|T_\lambda|}) = \Pr[\forall i \in [|T_\lambda|], \mathcal{O}(x_{\lambda,i}) = y_i : \mathcal{O} \leftarrow_{\$} \mathcal{M}_{\ell,\lambda}].$$

Next, we introduce the following functions:

$$Q_1(\lambda, \vec{y}) = \begin{cases} \Pr\left[1 \leftarrow_{\$} R_Q^{f_t^{\mathcal{O}}, S^{\mathcal{A}_t^{\mathcal{O}}}} \middle| \forall i \in [|T_\lambda|], \mathcal{O}(x_{\lambda,i}) = y_i\right] - \sigma_Q(\lambda) \text{ if } \vec{y} \in S_\lambda \\ 0 \text{ otherwise} \end{cases},$$

$$Q_2(\lambda, \vec{y}) = \begin{cases} \sup_{\mathcal{B} \in \mathsf{OA}} \lim_{\ell \to +\infty} \Pr\left[1 \leftarrow_{\$} C_Q^{f_t^{\mathcal{O}}(\lambda), \mathcal{B}_\ell^{\mathcal{O}}(\lambda)} \middle| \forall i, \mathcal{O}(x_{\lambda,i}) = y_i\right] \text{ if } \vec{y} \in S_\lambda \\ 0 \text{ otherwise} \end{cases}$$

and similarly

$$P(\lambda, \vec{y}) = \begin{cases} \Pr\left[1 \leftarrow_{\$} R_P^{G^{f_t^{\mathcal{O}}}, \mathcal{A}_t^{\mathcal{O}}} \middle| \forall i \in [|T_\lambda|], \mathcal{O}(x_{\lambda,i}) = y_i\right] - \sigma_P(\lambda) \text{ if } \vec{y} \in S_\lambda \\ 0 \text{ otherwise} \end{cases}.$$

Clearly, for any $\lambda \in \mathbb{N}$ and $\vec{y} \in S$ we have

$$-1 \leq Q_1(\lambda, \vec{y}), Q_2(\lambda, \vec{y}), P(\lambda, \vec{y}) \leq 1.$$

Let $g : \mathbb{N} \to S$. In order to apply Theorem 1 we need to prove that

$$Q_i(\lambda, g(\lambda)) \in \mathsf{UBnegl} \text{ for } i = 1, 2 \implies P(\lambda, g(\lambda)) \in \mathsf{UBnegl}.$$

Similarly as before, we define the function f_t^g with hardwired oracle queries g in the following way. Given a security parameter λ, it behaves identically as

in $f_t^{(\cdot)}$ but when f "queries an oracle" on input $x_{\lambda,i} \in T_\lambda$, it gets y_i where $g(\lambda) = (y_1, \ldots, y_{|T_\lambda|}) \in S_\lambda$. However, for λ so that $g(\lambda) \notin S_\lambda$, we set f_t^g to simply abort. Similarly, we define $\mathcal{A}_t^g$. It is easy to see that

$$Q_1(\lambda, g(\lambda)) = \mathsf{Adv}^{\mathrm{Q(M)}}_{f_t^g, S^{\mathcal{A}_t^g}}(\lambda).$$

Suppose $Q_i(\lambda, g(\lambda)) \in \mathsf{UBnegl}$ for $i = 1, 2$. This implies that (i) $\mathsf{Adv}^{\mathrm{Q(M)}}_{f_t^g, S^{\mathcal{A}_t^g}}(\lambda)$ is upper-bounded by a negligible function and (ii) f_t^g is an implementation of Q in the standard model because

$$Q_2(\lambda, g(\lambda)) = \mathsf{Corr}_{Q(M)}(f_t^g)(\lambda) \in \mathsf{UBnegl}$$

and $\mathsf{Corr}_{Q(M)}(f_t^g)(\lambda) \in [0, 1]$ for all λ. Since (G, S) is a fully black-box reduction from P to Q in the standard model, we have that $\mathsf{Adv}^{\mathrm{Q(M)}}_{G^{f_t^g}, \mathcal{A}_t^g}(\lambda)$ is upper-bounded by a negligible function as well – this is indeed equal to $P(\lambda, g(\lambda))$.

We can now apply Theorem 1 for $k = 2$ with functions defined above. We observe that by the Law of Total Probability, $\mathbb{E}(Q_1(\lambda, Z_\lambda))$ is equal to

$$\sum_{\vec{y} \in S_\lambda} \left(\Pr\left[1 \leftarrow\!\!\$\, R_Q^{f_t^{\mathcal{O}}, S^{\mathcal{A}_t^{\mathcal{O}}}} \,\middle|\, \forall i \in [|T_\lambda|], \mathcal{O}(x_{\lambda,i}) = y_i \right] - \sigma_Q(\lambda) \right) \cdot \Pr[Z_\lambda = \vec{y}].$$

Hence, we get

$$\mathbb{E}(Q_1(\lambda, Z_\lambda)) = \Pr\left[1 \leftarrow\!\!\$\, R_Q^{f_t^{\mathcal{O}}, S^{\mathcal{A}_t^{\mathcal{O}}}} \right] - \sigma_Q(\lambda) = \mathsf{Adv}^{\mathrm{Q(M)}}_{f_t, S^{\mathcal{A}_t}}(\lambda).$$

Since $\mathsf{Adv}^{\mathrm{Q(M)}}_{f_t, S^{\mathcal{A}_t}}(\lambda)$ is upper-bounded by a negligible function, then so is $\mathbb{E}(Q_1(\lambda, Z_\lambda))$. Similarly as in the proof of Theorem 3, one argues that

$$\mathbb{E}(Q_2(\lambda, Z_\lambda)) = \mathsf{Corr}_{Q(M)}(f_t)(\lambda).$$

Since $f_t^{(\cdot)}$ is an implementation of $Q(M)$, this function is negligible. Hence, by Theorem 1, we have $\mathbb{E}(P(\lambda, Z_\lambda)) \in \mathsf{UBnegl}$ which directly implies that $\mathsf{Adv}^{\mathrm{P(M)}}_{G^{f_t}, \mathcal{A}_t}(\lambda)$ is upper-bounded by a negligible function. Finally, $\mathsf{Adv}^{\mathrm{P(M)}}_{G^f, \mathcal{A}}(\lambda) \in \mathsf{UBnegl}$ by Lemma 10 and thus the statement holds. □

Acknowledgments: We would like to thank all the anonymous reviewers for their helpful suggestions which may also guide future work. Work was conducted while Eftychios Theodorakis was at DFINITY U.S. Research. Ngoc Khanh Nguyen was supported by the EU H2020 ERC Project 101002845 PLAZA.

References

1. Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 296–315. Springer, Heidelberg, December 2013.

2. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.
3. Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 232–249. Springer, Heidelberg, August 1994.
4. Manuel Blum, Paul Feldman, and Silvio Micali. Proving security against chosen cyphertext attacks. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 256–268. Springer, Heidelberg, August 1990.
5. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011.
6. 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, Heidelberg, May / June 1998.
7. Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 19–40. Springer, Heidelberg, August 2001.
8. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. Cryptology ePrint Archive, Report 1998/011, 1998. `http://eprint.iacr.org/1998/011`.
9. David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 89–105. Springer, Heidelberg, August 1993.
10. Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 427–444. Springer, Heidelberg, March 2008.
11. Jean-Sébastien Coron, Thomas Holenstein, Robin Künzler, Jacques Patarin, Yannick Seurin, and Stefano Tessaro. How to build an ideal cipher: The indifferentiability of the Feistel construction. *Journal of Cryptology*, 29(1):61–114, January 2016.
12. Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 1–20. Springer, Heidelberg, August 2008.
13. 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, Heidelberg, December 2010.
14. Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 39–56. Springer, Heidelberg, August 2008.
15. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
16. Dennis Hofheinz and Ngoc Khanh Nguyen. On tightly secure primitives in the multi-instance setting. LNCS, pages 581–611. Springer, Heidelberg, 2019.
17. 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, Heidelberg, August 2004.

18. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 44–61, 1989.
19. 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.
20. Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 115–128. Springer, Heidelberg, May 2007.
21. Michael O Rabin. Transaction protection by beacons. *Journal of Computer and System Sciences*, 27(2):256–267, 1983.
22. 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, Heidelberg, February 2004.
23. Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949.
24. 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, Heidelberg, May / June 1998.

A Supporting Proofs

A.1 Proof of Lemma 2

Firstly, we observe that for all $(a_s)_{s\in S} \in A^{|S|}$ we have:

$$\sum_{s\in S} f_s(a_s) \leq \sum_{s\in S} \sup_{a\in A} f_s(a)$$

and by definition of supremum we have

$$\sup_{(a_s)_{s\in S}\in A^{|S|}} \sum_{s\in S} f_s(a_s) \leq \sum_{s\in S} \sup_{a\in A} f_s(a).$$

Now, suppose there exists $\varepsilon > 0$ such that

$$\sum_{s\in S} \sup_{a\in A} f_s(a) = \sup_{(a_s)_{s\in S}\in A^{|S|}} \sum_{s\in S} f_s(a_s) + \varepsilon.$$

Let $\phi : S \to \mathbb{N}$ be an injective map. Then, by definition of supremum, for each $s \in S$ we can find an element $a_s \in A$ such that:

$$\sup_{a\in A} f_s(a) < f_s(a_s) + \varepsilon_{\phi(s)}$$

where ε_i is defined as $\varepsilon_i = (\varepsilon/2) \cdot (1/2)^i$ for $i \in \mathbb{N}$. Hence, we get:

$$\begin{aligned} \sum_{s\in S} \sup_{a\in A} f_s(a) &< \sum_{s\in S} f_s(a_s) + \sum_{s\in S} \varepsilon_{\phi(s)} \\ &< \sup_{(a_s)_{s\in S}\in A^{|S|}} \sum_{s\in S} f_s(a_s) + \sum_{i\in\mathbb{N}} \varepsilon_i \\ &< \sup_{(a_s)_{s\in S}\in A^{|S|}} \sum_{s\in S} f_s(a_s) + \varepsilon \end{aligned} \tag{15}$$

which leads to a contradiction.

A.2 Proof of Lemma 3

Let $\varepsilon > 0$. Then, there exists $\alpha \in A$ such that

$$\sup_{a\in A} \lim_{k\to+\infty} f_a(k) \leq \lim_{k\to+\infty} f_\alpha(k) + \varepsilon/2.$$

Next, there exists $N \in \mathbb{N}$ so that for all $n \geq N$:

$$|\lim_{k\to+\infty} f_\alpha(k) - f_\alpha(n)| < \varepsilon/2.$$

Since f_α is non-decreasing, we get

$$0 \leq \lim_{k\to+\infty} f_\alpha(k) - f_\alpha(n) < \varepsilon/2.$$

Therefore:

$$\sup_{a\in A} \lim_{k\to+\infty} f_a(k) - \varepsilon/2 \leq \lim_{k\to+\infty} f_\alpha(k) < f_\alpha(n) + \varepsilon/2 \leq \sup_{a\in A} f_a(n) + \varepsilon/2.$$

On the other hand, for any n, $\sup_{a\in A} \lim_{k\to+\infty} f_a(k) \geq \sup_{a\in A} f_a(n)$ since f_a is non-decreasing for all $a \in A$. Hence, for $n \geq N$ we have:

$$0 \leq \sup_{a\in A} \lim_{k\to+\infty} f_a(k) - \sup_{a\in A} f_a(n) < \varepsilon/2 + \varepsilon/2 = \varepsilon$$

and consequently, $\lim_{k\to+\infty} \sup_{a\in A} f_a(k) = \sup_{a\in A} \lim_{k\to+\infty} f_a(k)$.

A.3 Proof of Lemma 4

Denote $a_k = \lim_{\ell\to+\infty} f(k,\ell)$ and $b_\ell = \lim_{k\to+\infty} f(k,\ell)$. The monotonocity property and the fact that $f(k,\ell) \leq 1$ for all $k,\ell \in \mathbb{N}$ implies that sequences $(a_k), (b_\ell)$ are well-defined and they are non-decreasing. Moreover, $a_k, b_\ell \leq 1$ for all k,ℓ. Thus, $a = \lim_{k\to+\infty} a_k$ and $b = \lim_{\ell\to+\infty} b_\ell$ do exist. Then, for all $k,\ell \in \mathbb{N}$ we have $f(k,\ell) \leq a_k \leq a$ and hence

$$b_\ell = \lim_{k\to+\infty} f(k,\ell) \leq a$$

for all ℓ. In particular, $b = \lim_{\ell\to+\infty} b_\ell \leq a$. One similarly proves that $a \leq b$.

Lastly, we need to show that $c = a$ where $c := \lim_{k\to+\infty} f(k,k)$. It is easy to see that for $k \in \mathbb{N}$ we have $f(k,k) \leq a_k$ and thus $c = \lim_{k\to+\infty} f(k,k) \leq \lim_{k\to+\infty} a_k = a$. On the other hand, for every k and ℓ we have $f(k,\ell) \leq c$. Thus, $a_k = \lim_{\ell\to+\infty} f(k,\ell) \leq c$ for all k and consequently, $a \leq c$. Hence, $a = b = c$.

A.4 Proof of Lemma 5

The statement is easy to prove when S is finite. Hence, suppose there is a bijective map $\phi : \mathbb{N} \to S$ and define a function $g : \mathbb{N} \times \mathbb{N} \to [0,1]$ as $g(k,\ell) = \sum_{i=0}^{\ell} f(k,\phi(i))$. Note that for all k,ℓ we have $g(k,\ell) \leq g(k+1,\ell)$ and $g(k,\ell) \leq$

$g(k, \ell + 1)$. Then, by Lemma 4 and the fact that the limit of a finite sum is a sum of limits, we have:

$$\begin{aligned}
\lim_{k \to +\infty} \sum_{s \in S} f(k, s) &= \lim_{k \to +\infty} \lim_{\ell \to +\infty} g(k, \ell) \\
&= \lim_{\ell \to +\infty} \lim_{k \to +\infty} g(k, \ell) \\
&= \lim_{\ell \to +\infty} \lim_{k \to +\infty} \sum_{i=0}^{\ell} f(k, \phi(i)) \\
&= \lim_{\ell \to +\infty} \sum_{i=0}^{\ell} \lim_{k \to +\infty} f(k, \phi(i)) \\
&= \sum_{s \in S} \lim_{k \to +\infty} f(k, s).
\end{aligned} \tag{16}$$

A.5 Proof of Lemma 7

Clearly, $F(k, \ell) \in [0, 1]$. We just need to show that for all $k, \ell \in \mathbb{N}$ we have $F(k, \ell) \leq F(k, \ell + 1)$ and $F(k, \ell) \leq F(k + 1, \ell)$. Then, the statement follows directly from Lemma 4.

Let us fix $k, \ell \in \mathbb{N}$. Let us define $\mathcal{B}_{\ell+1}$ which behaves exactly as $\mathcal{A}_{\ell+1}$ but when it queries $x \in S$ such that $\phi(x) = \ell + 1$, it also aborts. Hence, we have

$$\Pr\left[1 \leftarrow_{\$} \mathcal{R}^{f_k{}^{\mathcal{O}}, \mathcal{B}^{\mathcal{O}}_{\ell+1}}\right] \leq \Pr\left[1 \leftarrow_{\$} \mathcal{R}^{f_k{}^{\mathcal{O}}, \mathcal{A}^{\mathcal{O}}_{\ell+1}}\right] = F(k, \ell + 1).$$

Now, by the consistency property of the setup assumption, the view of $\mathcal{B}_{\ell+1}$ given an oracle $\mathcal{O} \leftarrow_{\$} \mathcal{M}_{\max\{k,\ell+1\},\lambda}$ is exactly the same as $\mathcal{A}_{\ell}$ given $\mathcal{O} \leftarrow_{\$} \mathcal{M}_{\max\{k,\ell\},\lambda}$. Therefore

$$F(k, \ell) = \Pr\left[1 \leftarrow_{\$} \mathcal{R}^{f_k{}^{\mathcal{O}}, \mathcal{A}^{\mathcal{O}}_{\ell}}\right] = \Pr\left[1 \leftarrow_{\$} \mathcal{R}^{f_k{}^{\mathcal{O}}, \mathcal{B}^{\mathcal{O}}_{\ell+1}}\right].$$

Similarly, one proves $F(k, \ell) \leq F(k + 1, \ell)$.