Black-Box Uselessness: Composing Separations in Cryptography

Geoffroy Couteau
CNRS, IRIF, Université de Paris, France
couteau@irif.fr

Pooya Farshim
Department of Computer Science, University of York, UK
pooya.farshim@york.ac.uk

Mohammad Mahmoody
Department of Computer Science, University of Virginia, Charlottesville, VA, USA
mohammad@virginia.edu

Abstract

Black-box separations have been successfully used to identify the limits of a powerful set of tools in cryptography, namely those of black-box reductions. They allow proving that a large set of techniques are not capable of basing one primitive $\mathcal{P}$ on another $\mathcal{Q}$. Such separations, however, do not say anything about the power of the *combination* of primitives $\mathcal{Q}_1, \mathcal{Q}_2$ for constructing $\mathcal{P}$, even if $\mathcal{P}$ cannot be based on $\mathcal{Q}_1$ or $\mathcal{Q}_2$ alone.

By introducing and formalizing the notion of *black-box uselessness*, we develop a framework that allows us to make such conclusions. At an informal level, we call primitive $\mathcal{Q}$ black-box useless (BBU) for $\mathcal{P}$ if $\mathcal{Q}$ cannot help constructing $\mathcal{P}$ in a black-box way, even in the presence of another primitive $\mathcal{Z}$. This is formalized by saying that $\mathcal{Q}$ is BBU for $\mathcal{P}$ if for *any* auxiliary primitive $\mathcal{Z}$, whenever there exists a black-box construction of $\mathcal{P}$ from $(\mathcal{Q}, \mathcal{Z})$, then there must already also exist a black-box construction of $\mathcal{P}$ from $\mathcal{Z}$ alone. We also formalize various other notions of black-box uselessness, and consider in particular the setting of *efficient* black-box constructions when the number of queries to $\mathcal{Q}$ is below a threshold.

Impagliazzo and Rudich (STOC'89) initiated the study of black-box separations by separating key agreement from one-way functions. We prove a number of initial results in this direction, which indicate that one-way functions are perhaps also *black-box useless* for key agreement. In particular, we show that OWFs are black-box useless in any construction of key agreement in either of the following settings: (1) the key agreement has perfect correctness and one of the parties calls the OWF a constant number of times; (2) the key agreement consists of a single round of interaction (as in Merkle-type protocols). We conjecture that OWFs are indeed black-box useless for general key agreement.

We also show that certain techniques for proving black-box separations can be lifted to the uselessness regime. In particular, we show that the lower bounds of Canetti, Kalai, and Paneth (TCC'15) as well as Garg, Mahmoody, and Mohammed (Crypto'17 & TCC'17) for assumptions behind indistinguishability obfuscation (IO) can be extended to derive black-box uselessness of a variety of primitives for obtaining (approximately correct) IO. These results follow the so-called "compiling out" technique, which we prove to imply black-box uselessness.

Eventually, we study the complementary landscape of black-box uselessness, namely black-box *helpfulness*. We put forth the conjecture that one-way functions are black-box *helpful* for building collision-resistant hash functions. We define two natural relaxations of this conjecture, and prove that both of these conjectures are implied by a natural conjecture regarding random permutations equipped with a collision finder oracle, as defined by Simon (Eurocrypt'98). This conjecture may also be of interest in other contexts, such as amplification of hardness.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Black-Box Reductions, Separations, One-Way Functions, Key Agreement

12th Innovations in Theoretical Computer Science Conference (ITCS 2021).
Editor: James R. Lee; Article No. 47; pp. 47:1–47:20

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.47

Related Version Full Version: https://eprint.iacr.org/2021/016

Funding *Geoffroy Couteau*: Supported by ERC Project PREP-CRYPTO (724307).
Mohammad Mahmoody: Supported by NSF awards CCF-1910681 and CNS1936799.

1 Introduction

1.1 Background

Black-box reductions are a central tool in Cryptography. They have helped shape a rich landscape of relations between different cryptographic notions, allowing us to develop a better understanding of their powers and limitations.

Roughly speaking, in a (fully) black-box reduction both the design and analysis of a protocol treat the underlying primitives and adversaries in a black-box way, obliviously of their internals. More precisely, we say there is a fully black-box construction of a primitive $\mathcal{P}$ from a primitive $\mathcal{Q}$ if there is an efficient construction P^{Q} that for every implementation Q of primitive $\mathcal{Q}$ implements primitive $\mathcal{P}$, and further, there is an efficient security reduction $\mathsf{S}^{\mathsf{Q},A}$ which for every adversary A^{P} that breaks P, breaks Q. This notion originates in the seminal work of Impagliazzo and Rudich [20], and it was later refined by Reingold, Trevisan, and Vadhan [26] as well as Baecher, Brzuska, and Fischlin [3] who proposed a taxonomy of notions of reducibility between cryptographic primitives.

Impagliazzo and Rudich showed how to attack any key-agreement (KA) protocol in the random-oracle (RO) model with only a polynomial number queries to the random oracle.[1] This result is sufficient to rule out fully black-box reductions, since, roughly speaking, the construction is assumed to work for any OWF oracle f, and in particular for a RO and moreover, the security reduction works for any adversary A, and in particular for those that do not necessarily run in polynomial time but makes a polynomial number of queries to f.

Following this work, a successful and long line of research studying separations between different cryptographic primitives followed (e.g., see [2, 4, 9, 13, 16, 17, 28] and references therein). In this work we will revisit these works and ask if and to what extent their results hold in the presence of other primitives.

1.2 Black-Box Uselessness

Cryptographic constructions typically rely on multiple, incomparable building blocks. This raises the question if, and to what extent, a black-box separation result proves that some primitive is *useless* for building another even with other primitives. Take, for example, the case of key-agreement (KA) and one-way functions (OWFs). Although OWFs on their own are insufficient to build KA, this leaves the possibility that together with some other primitive $\mathcal{Z}$ they do imply KA in a black-box way, even if $\mathcal{Z}$ also does not black-box imply KA.[2] More generally, suppose we have separated primitive $\mathcal{P}$ from primitives $\mathcal{Q}_1$ and $\mathcal{Q}_2$ with respect to black-box reductions. That is, neither $\mathcal{Q}_1$ nor $\mathcal{Q}_2$ can be used to build $\mathcal{P}$. Does it then necessarily follow that $\mathcal{P}$ is also black-box separated from $\mathcal{Q}_1$ and $\mathcal{Q}_2$ put together? More generally, one may ask:

Which black-box impossibility results compose?

[1] More precisely, their attack made $\mathcal{O}((qm)^3)$ RO queries, where q is the number of queries of the protocol to the RO and m is the number of messages exchanged.
[2] Note that constructions of PKE from OWFs plus indistinguishability obfuscation are *non*-black-box [27].

In general, not all black-box impossibility results compose. Indeed, consider a primitive $\mathcal{P}$ that is set to be the "union" of primitives $\mathcal{Q}_1$ and $\mathcal{Q}_2$, where $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are mutually separated (i.e., neither can be based on the other). Then although $\mathcal{P}$ cannot be based on $\mathcal{Q}_1$ or $\mathcal{Q}_2$, it can be trivially based on their union.[3] This situation is somewhat unsatisfying: due to a joint effort of the cryptographic community in the past three decades, we have at our disposal a large number of black-box separations between pairs of primitives, yet we know essentially nothing about whether this situation changes if we are willing to use several primitives in a black-box construction – and of course, black-box separating subsets of primitives from a target primitive would be tedious. This leaves out the possibility that such separations could be obtained more systematically.

In this work, we seek to devise a more efficient strategy, by identifying conditions under which a primitive $\mathcal{P}$ is black-box separated from primitive $\mathcal{Q}$ *in a composable way*. That is, primitive $\mathcal{Q}$ in conjunction with *any* other primitive $\mathcal{Z}$ cannot be used to build $\mathcal{P}$, unless of course $\mathcal{P}$ can be built using $\mathcal{Z}$ alone. Our starting point is the following notion of *black-box uselessness*:[4]

▶ **Definition 1** (Black-box uselessness, informal)**.** *A primitive $\mathcal{Q}$ is black-box useless (BBU) for primitive $\mathcal{P}$ if for any auxiliary primitive $\mathcal{Z}$, whenever there is a black-box construction of $\mathcal{P}$ from $(\mathcal{Q}, \mathcal{Z})$ there also exist a black-box construction of $\mathcal{P}$ from the auxiliary primitive $\mathcal{Z}$ alone.*

A *composition theorem* for black-box uselessness immediately follows: if $\mathcal{Q}_1$ is BBU for $\mathcal{P}$ and $\mathcal{Q}_2$ is BBU for $\mathcal{P}$ then $\mathcal{Q}_1$ and $\mathcal{Q}_2$ put together are BBU for $\mathcal{P}$. Indeed, for any $\mathcal{Z}$, if $(\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Z}) = (\mathcal{Q}_1, (\mathcal{Q}_2, \mathcal{Z}))$ black-box implies $\mathcal{P}$, then $(\mathcal{Q}_2, \mathcal{Z})$ must black-box imply $\mathcal{P}$ (by the black-box uselessness of $\mathcal{Q}_1$). This in turn implies, by the black-box uselessness of $\mathcal{Q}_2$, that $\mathcal{Z}$ alone black-box imply $\mathcal{P}$.

▶ Remark 2. A black-box uselessness result of $\mathcal{Q}$ for $\mathcal{P}$ implies in particular that $\mathcal{Q}$ does not black-box imply $\mathcal{P}$, as long as $\mathcal{P}$ is not a (trivial) primitive that exists unconditionally. Indeed, by the black-box uselessness of $\mathcal{Q}$, if $\mathcal{Q}$ black-box implies $\mathcal{P}$, then taking $\mathcal{Z}$ as the "empty primitive" we get that $\mathcal{P}$ exists unconditionally in the plain model. For instance, although one-way functions are black-box useless for the one-time pad (since the latter exists unconditionally in the presence of any auxiliary oracle), one-way functions black-box imply the one-time pad (for essentially the same reason).

1.3 One-Way Functions and Key Agreement

Perhaps one of the most fundamental questions regarding black-box uselessness is to understand whether or not one-way functions are black-box useless for key agreement. We start with an observation on a natural approach for building key-agreement from one-way functions together with other primitives.

[3] This formal counterexample can be converted into more "natural" ones. Take identity-based encryption (IBE) with *compact* user keys, whose lengths are independent of the length of user identities. One can use a standard IBE and a collision-resistance hash function (CRHF) to build a compact IBE (by simply hashing the identities). Yet, compact IBE cannot be based on CRHFs in a black-box way (since PKE cannot be based on ROs). Furthermore, compact IBE cannot be based on standard IBE, since compact IBE implies CRHF (the keys must be collision-free for otherwise the compact IBE can be broken by finding a collision among keys) and standard IBE does not imply CRHFs [21].

[4] The terminology "a primitive X is useless for black-box constructions of a primitive Y" has been used sometimes in the literature, e.g. in [25], to mean that Y does not black-box reduce to X. Our notion of black-box uselessness should not be confused with this terminology, which only refers to conventional black-box separations.

▶ Remark 3. When looking for candidate primitives that, combined with one-way functions, imply key-agreement, the notion of indistinguishability obfuscation (iO, [27]) might be the first that comes to mind to a reader familiar with the literature on cryptography. As we mentioned, however, the construction of PKE from iO+OWFs in [27] is not black-box. Furthermore, one can observe that this is actually unavoidable: the seminal work of Impagliazzo and Rudich [20], which showed that there is no black-box construction of key agreement from one-way function, actually already shows that there is no black-box construction of key agreement from one-way functions *together with iO*. This is because the result of [20] shows that, relative to a random oracle *and a PSPACE oracle*, there is no key-agreement, yet there are one-way functions. However, relative to these oracles, there is also a perfectly-seure deterministic iO scheme: on input a circuit of a given size, the obfuscation scheme uses the PSPACE oracle to efficiently compute the lexicographically-first functionally equivalent circuit of the same size. This simple observation implies in particular that there is no black-box construction of key-agreement from iO and OWFs.

In this work, we provide a partial answer to the question of whether or not one-way functions are black-box useless for key agreement, by showing that one-way functions are black-box useless in any construction of key agreement satisfying certain restrictions. To describe our result, it is instructive to start with the separation of perfectly correct KA from OWFs by Brakerski, Katz, Segev, and Yerukhimovich [9].

BKSY11 in a nutshell

Given a perfectly correct, two-party KA protocol in the RO model, where Alice and Bob place at most q queries to the RO, consider an attacker Eve that proceeds as follows. Given a transcript T of the protocol, Eve samples coins r'_A and a random oracle RO' for Alice that are consistent with T. Eve then runs Alice on r'_A and RO' and records all oracle queries and the resulting key. It then places all these queries to the real RO to obtain an assignment (a list of query-answer pairs) L. Eve repeats this process, appending to L and storing keys, while ensuring that the sampling of RO' is consistent with L computed so far. Now, if in a sampled run of Alice, there are no queries of Alice in common with those of Bob outside L, by perfect correctness, the correct key will be computed. Otherwise, a query of Bob will be discovered. Since Bob has at most q queries, if Eve executes this procedure $2q + 1$ times, in at least q of the runs no intersection queries will be discovered, and in these runs the correct key will be computed. Thus taking a majority over the set of keys computed, Eve obtains the key with probability one.

Upgrading to BBU

In order to convert this proof into a black-box uselessness result, we make a case distinction based on whether or not one can "lift" the sampling procedure to a Z-relativized world for any oracle Z. Given a construction $\mathsf{KA}^{\mathsf{F},\mathsf{Z}}$ of KA from a OWF F and Z, if the sampling procedure can be successfully carried out in the presence of Z, then we could efficiently break any perfectly correct KA protocol in the presence of Z only (since the attacker computes the correct key with probability one).

Now suppose at some iteration Eve is no longer able to find coins and a random oracle consistent with the transcript while making polynomially many queries to Z. We claim that in this case we can construct a *weak* one-way function. Indeed, consider the function that runs the above attack procedure up to but excluding the stage where Eve fails. Thus, this function starts by running the protocol, then uses the sampler for the first iteration (if

sampling was not already impossible at this stage) to obtain an assignment, and continues with another round of sampling, until it arrives at the stage where sampling is no longer possible. The transcript of the protocol at this stage together with the list of assignments so far constitutes the challenge for which there is no inverter. From a weak one-way function, a full-fledged one-way function follows by the result of [29], as this construction is black-box and hence relativizes with respect to Z.

We may now use the one-way function obtained from Z in place of the original one-way function F to remove reliance on the F all together. Thus, we obtain a KA protocol which relies only on Z, as required. We emphasize that this proof only shows the uselessness of OWFs for (perfect) KA and not that of random oracles, since we only obtain a one-way function using Z.

Note that since we recursively rely on the existence of an inverter, the query complexity (to Z) of the samplers can potentially blow up. Indeed, suppose for some Z, any sampler needs to place $\mathcal{O}(n^2)$ queries to Z to invert a function that places n queries to Z. After the first iteration, we arrive at the construction of a function that places $n + \mathcal{O}(n^2) = \mathcal{O}(n^2)$ queries to Z. Thus, it may well be that a successful inverter at this step needs to place $\mathcal{O}(n^4)$ queries to Z. This in particular implies that the recursive argument above can be applied for only a *constant* number of steps. Since we only need to apply the recursive sampling for *either* Alice *or* Bob, we obtain a BBU result as long as either Alice or Bob makes a constant number of queries to the RO (but polynomially many calls to Z). We formalize this proof in Section 4, where we point out the other subtleties that arise.

Currently, we are not able to extend the proof to arbitrary protocols where both Alice and Bob make a polynomial number of RO queries. Despite this, we can show that OWFs are black-box useless for building *constant-round, imperfect* key agreements when both parties make a constant number of queries to the OWF (and an arbitrary number of queries to the auxiliary oracle), and that OWFs are black-box useless for *one-round* key agreement (without restriction on the queries made by the parties). We defer the details to Section 4.2.

1.4 The Compilation Technique

A number of black-box separation results rely on what we here refer to as the *efficient compiling-out* (or simply compilation) paradigm [1, 7, 10, 12, 13, 15, 16]. At a high-level, here given a construction $\mathsf{G}_1^{\mathsf{P}}$ one compiles out the primitive P via an (oracle-free) simulator Sim which simulates P for the construction in a consistent way and without affecting security. The result is a new construction G_2 that no longer uses P. This proof technique is closely related to black-box uselessness, and as we will see can often be turned into a black-box uselessness result with minor modifications. This in turn highlights the advantage of separation results that are achieved using this technique.

In order to show how this can be done, we briefly discuss this in the context of the work of Canetti, Kalai, Paneth [10], who showed that obfuscation cannot be based on random oracles.[5]

[5] The work of [10] dealt with *virtual-black-box* obfuscation, but as it turned out [8, 22, 23], their proof could also be applied to the case of indistinguishability obfuscation.

CKP and black-box uselessness of random oracles for indistinguishability obfuscation

Consider an obfuscator $\mathsf{Obf}^{\mathsf{RO}}$ that makes use of a random oracle RO. On input a circuit without random oracle gates[6] the obfuscation algorithm outputs a circuit C^{RO}, which may also call the random oracle RO. CKP compile out the random oracle RO from any such construction as follows. First they convert $\mathsf{Obf}^{\mathsf{RO}}$ to an obfuscator in the plain model by simulating the RO for the obfuscator via lazy sampling. Next to ensure that the oracle expected by an obfuscated circuit C^{RO} is consistent with the oracle simulated for obfuscation, CKP execute C on multiple randomly chosen inputs at obfuscation phase while simulating the random oracle consistently, record all query made and answers simulated, and return them together with the obfuscated oracle circuit. Leaking this list cannot hurt security as an adversary in the RO model can also compute such a list given an obfuscated circuit. On the other hand, having this list allows the evaluation of C^{RO} to be consistent with the obfuscation phase with high probability on a *random* input. (This is why the obtained primitive is only an *approximate-correct* IO.)

As can be seen, this proof Z-relativizes in the sense that the simulation of the random oracle RO and the execution of the obfuscated circuit can be both done in the presence of any other oracle Z. By compiling out the random oracle RO in the presence of Z, we obtain a construction that relies on Z. That is, we obtain that random oracles are black-box uselessness for obfuscation.

A number of other impossibility results follow the compilation paradigm and here we observe that they can be lifted to the black-box uselessness regime. In particular, we go over such observations about results from [10, 13, 15, 16] and show how they can be lifted to black-box uselessness. Indeed, as long as compilation is performed for a constant "number of rounds" and each step can be done efficiently we obtain black-box uselessness.[7] As a result we get that approximate IO cannot be obtained from a black-box *combination* of random oracles, predicate encryption, fully homomorphic encryption and witness encryption.[8]

▶ Remark 4. We note that some previous works (e.g., [10]) have described the result of Impagliazzo and Rudich as "compiling out" the random oracle from any key-agreement protocol. This process, however, differs from the compiling-out technique that we study in this paper in two aspects. First, compilation is inefficient and uses the sampling algorithm. Second, the process is carried out adaptively for multiple rounds. The inefficiency of the sampler translates to obtaining BBU for one-way functions only; adaptivity restricts our final result to protocols where one party makes at most a constant number of RO queries. On a similar note, the recent work of Maji and Wang [25] uses the term "black-box useless" as an alternative to proving (traditional) black-box separations. So, despite similarity in the terms, our notion of uselessness is quite different.

1.5 The Case of Collision Resistance

A classic result of Simon [28] separates collision-resistance hash functions (CRHFs) from one-way functions (and indeed one-way permutations). This is done by giving a pair of oracles $(\pi, \mathsf{Coll}^{\pi})$ relative to which one-way permutations (and hence one-way functions) exist,

[6] This restriction only makes the results of CKP stronger.
[7] Lemma 3.25 in [14] shows a similar phenomenon in a context where we completely compile out a sequence of idealized oracles. Here we deal with a setting that an auxiliary oracle remains at the end.
[8] We note that this does not apply in the so-called monolithic model.

but CRHFs don't. Here π implements a random permutation, and Coll^π is an oracle that takes as input a circuit with π-gates and returns a random collision for it (by first computing the circuit on a random point and then picking a random preimage).

Are one-way functions/permutations also black-box useless for collision resistance? One way to answer this question affirmatively would be to extend Simon's result along the lines of what was done for the separation results above. However, in this case we have an oracle separation result, and it is not clear how to "relativize" the proof. Indeed, we conjecture the opposite: OWFs are black-box *helpful* for building CRHFs. To this end, we would need to show that for *any* one-way function F there is a primitive Z, which although on its own is insufficient for building CRHFs, together with F can be used to build CRHFs. We present two possible approaches for proving this conjecture.

One approach follows the recent result of Holmgren and Lombardi [18], who showed how to obtain CRHFs from exponentially secure *one-way product functions* (OWPFs) in a black-box way. Roughly speaking, a OWP is a tuple of one-way functions $(\mathsf{F}_1, \ldots, \mathsf{F}_k)$ where any polynomial-time adversary can invert $(\mathsf{F}_1(x_1), \ldots, \mathsf{F}_k(x_k))$ for random x_i with probability at most $\mathrm{negl}(n)/2^n$ (where n is the security parameter). A good candidate for primitive Z is thus a random permutation π together with Simon's oracle Coll^π for it. To get a positive helpfulness result we need to show that for any one-way function F the pair of functions $(\mathsf{F}, (\pi, \mathsf{Coll}^\pi))$ is product one-way. We intuitively expect this result to hold since π is fully independent of F and essentially all an adversary can do is to invert the F and (π, Coll^π) independently. Formalizing this observation requires handling additional technicalities; we refer the reader to the full version for details. We did not manage to prove this conjecture, and leave it as an interesting open problem which might be of independent interest.[9]

A second approach follows the work of Bauer, Farshim, Mazaheri [5], who defined a new idealized model of computation, known as the backdoored random oracle (BRO) model, whereby an adversary can obtain arbitrary leakage of the function table of the RO. Under a communication complexity conjecture related to the set-intersection problem, BFM show that two independent BROs can be used to build a CRHF by simply xoring their outputs. The leakage oracle defined by BFM is sufficiently powerful to allow implementing Simon's collision-finding oracle. As a result, although a single BRO as an idealized primitive is black-box separated from CRHFs, conjecturally it is not black-box useless for building CRHFs.

Open problems

The central open problem left by our work is that of black-box uselessness of OWFs for arbitrary key agreement protocols. Given our BBU results for special classes of KA protocols, this conjecture may well be within reach. On the other hand, a straightforward generalization seems to require a refined sampler technique with low *adaptivity*. At the moment, however, it seems challenging to reduce the adaptivities of known samplers [4, 9, 20, 24]. Besides OWFs, whether or not CRHFs, or for that matter random oracles, are black-box useless for key agreement remains open. More generally, black-box separation results can be revisited from the point of view of uselessness. In particular, it would be interesting to consider extensions of the recent *monolithic* models to the BBU setting, as these capture certain well-known non-black-box techniques in cryptography.

[9] It might be helpful to consider weaker versions of this problem. For example, given an ε-secure one-way permutation and a random oracle, can an attacker invert both simultaneously with probability better than $\mathrm{negl}(n)/2^n$?

2 Preliminaries

Notation

PPT stands for probabilistic polynomial time. An oracle-aided PPT machine/circuit A^{O} is a PPT machine/circuit with oracle access/gates, such that for any oracle O machine/circuit A^{O} runs in probabilistic polynomial time, where we count each call to the oracle as one step. For any $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, \ldots, n\}$. Given an interactive protocol between two parties Alice and Bob with access to an oracle O, we let $\langle \mathsf{Alice}^{\mathsf{O}}, \mathsf{Bob}^{\mathsf{O}} \rangle$ denote the distribution of triples (T, y_A, y_B) over the randomness of the parties and the oracle, where T denotes the transcript of the protocol, and y_A (resp., y_B) denotes the output of Alice (resp., Bob). We also use the same notation when O comes from a *distribution* over the oracles, in which case the probability distribution of the outputs are also over the randomness of the oracle.

2.1 Black-Box Reductions

The definitions and notions in this section mostly follow those in [26]. Throughout, we use calligraphic letters such as $\mathcal{P}$ or $\mathcal{KA}$ for a cryptographic primitive, sans-serif letters (for example P or KA) for specific implementations, S for the security reduction/proof and P for a "generic" implementation. We denote an auxiliary oracle by Z and an adversary by A.

▶ **Definition 5** (Cryptographic primitive)**.** *A cryptographic primitive $\mathcal{P}$ is a pair $(F_{\mathcal{P}}, R_{\mathcal{P}})$, where $F_{\mathcal{P}}$ is the set of functions* implementing *$\mathcal{P}$, and $R_{\mathcal{P}}$ is a relation. For each $\mathsf{P} \in F_{\mathcal{P}}$, the relation $(\mathsf{P}, A) \in R_{\mathcal{P}}$ means that the adversary A* breaks *the implementation P (according to $\mathcal{P}$). It is required that at least one function $\mathsf{P} \in \mathcal{P}$ is computable by a PPT algorithm.*

▶ **Definition 6** (Fully black-box reduction)**.** *A* fully black-box *reduction of a primitive $\mathcal{P}$ to another primitive $\mathcal{Q}$ is a pair (P, S) of oracle-aided PPT machine such that for any implementation $\mathsf{Q} \in \mathcal{Q}$ the following two conditions hold.*

- **Implementation reduction***: P^{Q} implements $\mathcal{P}$, that is, $\mathsf{P}^{\mathsf{Q}} \in F_{\mathcal{P}}$.*
- **Security reduction***: For any function (adversary) A that $\mathcal{P}$-breaks $\mathsf{P}^{\mathsf{Q}} \in F_{\mathcal{P}}$, i.e., $(\mathsf{P}^{\mathsf{Q}}, A) \in R_{\mathcal{P}}$), it holds that $\mathsf{S}^{\mathsf{Q},A}$ $\mathcal{Q}$-breaks Q, i.e., $(\mathsf{Q}, \mathsf{S}^{\mathsf{Q},A}) \in R_{\mathcal{Q}}$.*

When clear from context, we will refer to fully black-box reductions simply as black-box reductions, to the implementation reduction as the *construction*, and to the security reduction as the *security proof*.

▶ **Definition 7** (Semi and weakly black-box reductions)**.** *We say there is a* semi-black-box *reduction of a primitive $\mathcal{P}$ to primitive $\mathcal{Q}$ if there is an oracle-aided PPT P such that for any implementation $\mathsf{Q} \in \mathcal{Q}$,*

- **Implementation reduction***: $\mathsf{P}^{\mathsf{Q}} \in F_{\mathcal{P}}$.*
- **Security reduction***: If there exists an oracle-aided PPT A such that A^{Q} $\mathcal{P}$-breaks P^{Q}, then there exists an oracle-aided PPT S such that S^{Q} $\mathcal{Q}$-breaks Q.*

*If the order of the quantifiers for the implementation reduction is switched in the sense that for any implementation $\mathsf{Q} \in \mathcal{Q}$ there is an oracle-aided PPT P such that the above two conditions hold, then we say there is a $\forall\exists$-*semi-black-box *reduction of $\mathcal{P}$ to $\mathcal{Q}$.*

Weakly black-box *reductions and a $\forall\exists$ variant thereof are defined analogously with the difference that the security reduction S is a PPT (instead of an oracle-aided PPT) machine and can no longer call Q.*

▶ **Definition 8** (Existence relative to an oracle)**.** *A primitive $\mathcal{P}$ is said to* exist *relative to an oracle O whenever (1) there is an oracle-aided PPT P such that $\mathsf{P}^{\mathsf{O}} \in F_{\mathcal{P}}$; and (2) no oracle-aided PPT machine A^{O} can $\mathcal{P}$-break P^{O}.*

Non-uniform variants of security reductions were formalized in [11]. The following definition extends this to non-uniform implementation reductions.

▶ **Definition 9.** *A fully black-box reduction* (P,S) *of a primitive* $\mathcal{P}$ *from primitive* $\mathcal{Q}$ *is said to have a* non-uniform implementation *if* P *additionally takes as input a polynomial-sized non-uniform advice that can also depend on its oracle* Q*. The reduction is said to have a* non-uniform security reduction *if* S *additionally takes as input a polynomial-sized non-uniform advice that can also depend on its oracles* Q *and* A*.*

2.2 Specific Cryptographic Primitives

▶ **Definition 10** (One-way functions)**.** *A* one-way function F *relative to an oracle* O *is an oracle-aided PPT machine such that for any PPT adversary* A *(modeled as an oracle-aided PPT machine) and any sufficiently large values of the security parameter* λ*, it holds that*

$$\Pr_{x\xleftarrow{\$}\{0,1\}^n}[\mathsf{F}^{\mathsf{O}}(A^{\mathsf{O}}(1^\lambda,\mathsf{F}^{\mathsf{O}}(x)))=\mathsf{F}^{\mathsf{O}}(x)]=\mathrm{negl}(\lambda)\ .$$

If the above is only required to hold for infinitely many values of $\lambda\in\mathbb{N}$*, then* F *is an* infinitely-often *one-way function (*io-$\mathcal{F}$*).*

▶ **Definition 11** (Key agreement)**.** *An oracle-aided* ε*-key agreement with respect to an oracle* O *is an interactive protocol between two oracle-aided PPT machines* Alice *and* Bob *that satisfies the following* ε-correctness *and* security *properties.*

- *(*ε*-correctness) For any* $\lambda\in\mathbb{N}$*,*

$$\Pr[(T,K_A,K_B)\xleftarrow{\$}\langle\mathsf{Alice}^{\mathsf{O}}(1^\lambda),\mathsf{Bob}^{\mathsf{O}}(1^\lambda)\rangle\ :\ K_A=K_B]\geq\varepsilon(\lambda)\ .$$

- *(Security) For any PPT adversary* Eve*, any polynomial* p*, and any sufficiently large* λ*,*

$$\Pr[(T,K_A,K_B)\xleftarrow{\$}\langle\mathsf{Alice}^{\mathsf{O}}(1^\lambda),\mathsf{Bob}^{\mathsf{O}}(1^\lambda)\rangle,K_E\xleftarrow{\$}\mathsf{Eve}^{\mathsf{O}}(1^\lambda,T)\ :\ K_E=K_B]\leq\frac{\varepsilon(\lambda)}{p(\lambda)}\ .$$

If security is only required to hold for infinitely many values of $\lambda\in\mathbb{N}$ *in the sense that for every polynomial* p *and all adversaries, there exists an infinite set of* λ *for which the adversary's winning probability is below* $\varepsilon(\lambda)/p(\lambda)$*, then the construction is called an* infinitely-often *key agreement. If the number of queries to* O *is bounded by a constant for either* Alice *or* Bob*, then we say that the key agreement is* unbalanced *with respect to* O*. We say that the key agreement is a* perfectly correct *when* $\varepsilon(\lambda)\equiv 1$*.*

3 Defining Black-Box Uselessness

To formally define black-box uselessness, we first formalize joint primitives, to simplify statements about black-box constructions from several primitives:

▶ **Definition 12** (Joint primitive)**.** *Given two primitives* $\mathcal{P}=(F_{\mathcal{P}},R_{\mathcal{P}})$ *and* $\mathcal{Q}=(F_{\mathcal{Q}},R_{\mathcal{Q}})$ *the joint primitive* $(\mathcal{P},\mathcal{Q})=(F_{(\mathcal{P},\mathcal{Q})},R_{(\mathcal{P},\mathcal{Q})})$ *is defined by* $F_{(\mathcal{P},\mathcal{Q})}:=F_{\mathcal{P}}\times F_{\mathcal{Q}}$ *and for each* $\mathsf{G}=(\mathsf{P},\mathsf{Q})\in F_{(\mathcal{P},\mathcal{Q})}$ *and any* $A=(A_{\mathcal{P}},A_{\mathcal{Q}})$*, we define* $(\mathsf{G},A)\in R_{(\mathcal{P},\mathcal{Q})}$ *iff* $(\mathsf{P},A_{\mathcal{P}})\in R_{\mathcal{P}}$ *or* $(\mathsf{Q},A_{\mathcal{Q}})\in R_{\mathcal{Q}}$*.*

We are now ready to formally define what it means for a cryptographic primitive $\mathcal{Q}$ to be black-box useless for a primitive $\mathcal{P}$.

3.1 Definition

The following definition is more general than complete black-box uselessness of a primitive $\mathcal{P}$ for obtaining another primitive $\mathcal{Q}$. In particular, the definition states a *set* of primitives $\mathbb{Z}$ such that $\mathcal{P}$ is useless for obtaining $\mathcal{Q}$ in the presence of *any* of the primitives $\mathcal{R} \in \mathbb{Z}$.

▶ **Definition 13** (Black-box uselessness)**.** *Suppose $\mathbb{Z}$ is a set of primitives. A cryptographic primitive $\mathcal{Q}$ is resp., fully, semi, $\forall\exists$-semi* black-box useless *for constructing a primitive $\mathcal{P}$ in the presence of auxiliary primitives $\mathbb{Z}$, if the following holds: For every auxiliary primitive $\mathcal{Z} \in \mathbb{Z}$ whenever there exists a resp., fully, semi, $\forall\exists$-semi black-box reduction of $\mathcal{P}$ to the joint primitive $(\mathcal{Q}, \mathcal{Z})$ there also exists a resp., fully, semi, $\forall\exists$-semi black-box reduction of $\mathcal{P}$ to $\mathcal{Z}$ alone. In the special case where $\mathbb{Z}$ contains* all *primitives, then we simply say that $\mathcal{Q}$ is resp., fully, semi, $\forall\exists$-semi* black-box useless *for constructing a primitive $\mathcal{P}$.*

▶ Remark 14 (Other special cases of Definition 13). Here we point out to two other important special cases of Definition 13 that could be obtained $\mathbb{Z}$ differently. If $\mathbb{Z} = \varnothing$, then black-box uselessness is the same as a traditional black-box separation showing that $\mathcal{Q}$ cannot be black-box reduced to $\mathcal{P}$. In addition, when $\mathbb{Z}$ contains a specific primitive $\mathcal{Z}$, then Definition 13 captures the notion that $\mathcal{P}$ is useless for building $\mathcal{Q}$ when we already assume the existence of $\mathcal{Z}$ as a black-box.

▶ Remark 15 (Other variants of Definition 13). By default we consider the three main cases where the source construction and the target construction in Definition 13 both use the same flavor of black-box reduction, and accordingly use the terms fully, semi, or $\forall\exists$-semi to describe the corresponding notion of black-box uselessness. However, we can (and will) also consider more general notions of black-box uselessness whereby the source construction and the target construction use different notions of black-box reduction. For example, we will write that $\mathcal{Q}$ is [semi $\to$ $\forall\exists$-semi] black-box useless for $\mathcal{P}$ if for every auxiliary primitive $\mathcal{Z}$, whenever there exists a semi-black-box reduction of $\mathcal{P}$ to the joint primitive $(\mathcal{Q}, \mathcal{Z})$ there is a $\forall\exists$-semi-black-box reduction of $\mathcal{P}$ to $\mathcal{Z}$ alone.

3.2 Composition

Given the definition of black-box uselessness, the following composition theorem follows easily. Here, we use the term black-box uselessness to refer to any fixed flavor of black-box uselessness.

▶ **Theorem 16.** *Let $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$ be three cryptographic primitives. If $\mathcal{Q}$ is black-box useless for $\mathcal{P}$ and $\mathcal{R}$ is black-box useless for $\mathcal{P}$ (for the same flavor), then the joint primitive $(\mathcal{Q}, \mathcal{R})$ is black-box useless for $\mathcal{P}$ (for the same flavor).*

Proof. Let $\mathcal{Z}$ be an arbitrary auxiliary primitive. If there is a black-box reduction of $\mathcal{P}$ to the joint primitive $(\mathcal{Z}, (\mathcal{Q}, \mathcal{R})) = ((\mathcal{Z}, \mathcal{Q}), \mathcal{R})$, then by the black-box uselessness of $\mathcal{R}$ for $\mathcal{P}$, there is a black-box reduction of $\mathcal{P}$ to $(\mathcal{Z}, \mathcal{Q})$ (viewing $(\mathcal{Z}, \mathcal{Q})$ as an auxiliary primitive). In turn, using the black-box uselessness of $\mathcal{Q}$ for $\mathcal{P}$, we obtain a black-box construction of $\mathcal{P}$ from $\mathcal{Z}$ alone. Hence, $(\mathcal{Q}, \mathcal{R})$ is black-box useless for $\mathcal{P}$. ◀

We note that a similar composition theorem can be easily established even when $\mathcal{Q}$ and $\mathcal{R}$ do not satisfy the same flavor of black-box uselessness for $\mathcal{P}$, in the following sense: if a primitive $\mathcal{Q}$ is $[X \to Y]$ BBU for $\mathcal{P}$ and a primitive $\mathcal{R}$ is $[X' \to Y']$ BBU for $\mathcal{P}$, where X, Y, X', Y' are flavors of black-box reduction, then $(\mathcal{Q}, \mathcal{R})$ is $[X' \to Y]$ BBU for $\mathcal{P}$ as long as X a stronger flavor than Y' (e.g., X is "fully" and Y' is "semi").

3.3 Restricted Black-Box Uselessness

In many settings, it can be useful to consider a more general notion of black-box uselessness, which restricts the type of primitive (e.g., only infinitely-often variants) or the type of construction for which black-box uselessness is shown to hold. For readability, we will not define cumbersome formal notations for such variants, but instead will simply state the restriction explicitly when needed.

This generalization is especially useful to study the *efficiency* of black-box reductions. Indeed, black-box separations in cryptography are not limited to only showing their nonexistence. A more concrete treatment would make statements of the form "in any black-box construction of $\mathcal{P}$ from $\mathcal{Q}$, any implementation of $\mathcal{P}$ must call an implementation of $\mathcal{Q}$ at least q times," or for interactive primitives states that "any black-box construction of $\mathcal{P}$ from $\mathcal{Q}$ must have at least r rounds". This approach to bounding the efficiency of generic cryptographic construction was initiated in the seminal work of Gennaro, Gertnet, Katz, and Trevisan [16] and has subsequently proven very fruitful.

4 On the Black-Box Uselessness of OWFs for Key Agreement

In this section, we prove black-box uselessness of OWFs for key agreement for several natural special forms of key agreement protocols. We leave the proof of black-box uselessness of OWFs for general key agreement protocols as an intriguing open question.

4.1 Theorem Statement for Perfectly Correct Key Agreement

In this section, we prove the following:

▶ **Theorem 17** (Black-box uselessness of OWFs for perfect unbalanced KA)**.** *Infinitely-often one-way functions are [semi $\to \forall\exists$-semi] black-box useless for infinitely-often perfect key agreement in any construction which is unbalanced with respect to the io-OWF.*

Before proving Theorem 17, let us breakdown its content. The "dream result" here would be to show that one-way functions are black-box useless for any key agreement. Unfortunately, we do not know how to prove this result. Theorem 17 provides a meaningful step in this direction, but it suffers from three limitations:

1. It only applies to *infinitely-often* one-way functions (though it is incomparable in that it shows black-box uselessness for infinitely-often key agreement, which is a weaker primitive).
2. It only applies to constructions where one of the parties makes a *constant* number of queries to the io-OWF oracle, which we call unbalanced key agreement. Note that the key agreement can still make an arbitrary number of queries to the auxiliary primitive.
3. It only applies to *perfectly correct* key agreement.

The first limitation stems from the fact that our proof of Theorem 17 relies on a case distinction based on the existence of one-way functions: if they exist, we get a construction of key agreement, else we get an attack on the candidate construction. However, this attack requires applying a one-way function inverter to several functions at once. But since a one-way function inverter is only guaranteed to succeed on *infinitely many* security parameters, which need not be equal across the different functions that we need to invert (and in fact could be exponentially far apart), this approach fails. To get around this, we rely on an inverter for an infinitely-often OWF, which gives an inverter which is guaranteed to work for *all* sufficiently large security parameters and we can use to simultaneously invert several functions. This,

however, comes at the cost of getting instead a black-box uselessness result for infinitely-often OWFs (for infinitely-often key agreements). Such technicalities are relatively common in cryptography and stem from the asymptotic nature of primitives.

▶ Remark 18. In general, statements of the form "$\mathcal{A}$ and $\mathcal{B}$ black-box imply $\mathcal{C}$" and statements of the form "io-$\mathcal{A}$ and io-$\mathcal{B}$ black-box imply io-$\mathcal{C}$", where io-$\mathcal{X}$ denotes an infinitely-often flavor of a primitive $\mathcal{X}$, can be incomparable for the trivial reason that io-$\mathcal{A}$ and io-$\mathcal{B}$ can never be simultaneously secure on the same security parameters. However, this situation does not arise in the setting of black-box uselessness, since the statement "io-$\mathcal{A}$ is BBU for io-$\mathcal{C}$" refers to the inexistence of black-box construction of io-$\mathcal{C}$ from io-$\mathcal{A}$ together with any other primitive $\mathcal{Z}$ – and not only "infinitely-often" types of primitives. In general, it is easy to show that the statement "$\mathcal{A}$ is BBU for $\mathcal{C}$" is stronger than (i.e., implies) the statement "io-$\mathcal{A}$ is BBU for io-$\mathcal{C}$" for all notions of black-box uselessness.

The second limitation stems from the fact that the proof requires to iteratively define efficient functions F_i, where each F_i builds upon an (efficient) OWF inverter applied to F_{i-1}. The total number of functions can be picked to be the minimum of the number of queries to the OWF made by either of the two parties. However, this argument crucially relies on the fact that the number of functions F_i is a constant; to see this, imagine that we have at our disposal a OWF inverter that would always makes a number of queries quadratic in the number of queries made by the function in the forward direction. Such an inverter would be efficient (i.e., it inverses any poly time function in poly time), yet one cannot obtain a poly time function by iteratively defining a function F_i which invokes $\mathsf{Inv}_{\mathsf{F}_{i-1}}$ unless i is constant, since the complexity of F_i grows as $\mathsf{runtime}(\mathsf{F}_1)^{2^i}$.

Eventually, our result in this section focuses on perfectly correct key agreement. We discuss the case of imperfect key agreement in Section 4.2.

4.1.1 A Helpful Logical Lemma

Below, we state a simple lemma which allows for more direct proofs of [semi $\to \forall\exists$-semi] black-box uselessness.

▶ **Lemma 19.** *Let $\mathcal{P}$ and $\mathcal{Q}$ be two primitives. Then whenever the following statement is established, it implies in particular that $\mathcal{Q}$ is [semi $\to \forall\exists$-semi] black-box useless for $\mathcal{P}$:*

"Fix any primitive $\mathcal{Z}$ and any $\mathsf{Z} \in F_{\mathcal{Z}}$. Assume that there exists an oracle-aided PPT P_1 such that for any $\mathsf{Q} \in F_{\mathcal{Q}}$, $\mathsf{P}_1^{\mathsf{Q},\mathsf{Z}} \in F_{\mathcal{P}}$. Further assume that whenever (Q,Z) is a secure implementation of $(\mathcal{Q},\mathcal{Z})$, then $\mathsf{P}_1^{\mathsf{Q},\mathsf{Z}}$ is a secure implementation of $\mathcal{P}$. Then there exists an efficient implementation $\mathsf{P}_2^{\mathsf{Z}}$ of $\mathcal{P}$ relative to Z, and furthermore, whenever Z is a secure implementation of $\mathcal{Z}$, $\mathsf{P}_2^{\mathsf{Z}}$ is a secure implementation of $\mathcal{P}$."

We prove this lemma in the full version of the paper.

4.1.2 Proof of Theorem 17

Let io-$\mathcal{F}$ be the io-OWF primitive. To prove of Theorem 17, we will prove the following:

▶ **Lemma 20.** *Fix any primitive $\mathcal{Z}$ and any $\mathsf{Z} \in F_{\mathcal{Z}}$. Assume that there exists an oracle-aided PPT KA_1 such that for any implementation ioF of an infinitely-often one-way function, $\mathsf{KA}_1^{\mathsf{ioF},\mathsf{Z}}$ implements an infinitely-often perfect key agreement unbalanced with respect to ioF, relative to $(\mathsf{ioF},\mathsf{Z})$. Assume furthermore that if $(\mathsf{ioF},\mathsf{Z})$ is a secure implementation of $(\mathsf{io}\text{-}\mathcal{F},\mathcal{Z})$, then $\mathsf{KA}_1^{\mathsf{ioF},\mathsf{Z}}$ is a secure implementation of infinitely-often key agreement, unbalanced with respect to ioF. Then there exists an efficient implementation KA_2 of (infinitely-often) key agreement relative to Z, and furthermore, if Z is a secure implementation of $\mathcal{Z}$, then $\mathsf{KA}_2^{\mathsf{Z}}$ is a secure implementation of infinitely-often key agreement.*

The proof of Theorem 17 follows directly from the above lemma by applying Lemma 19. To prove Lemma 20, we rely on the following lemma.

▶ **Lemma 21.** *Let* RO *be a random oracle. For any auxiliary oracle* Z*, if there exists no infinitely-often one-way function relative to* Z*, then there exists no construction* $\mathsf{KA}^{\mathsf{RO},\mathsf{Z}}$ *of a perfect infinitely-often key agreement which is unbalanced with respect to* RO*.*

Given Lemma 21, the proof of Lemma 20 follows from a disjunction argument: fix any auxiliary primitive $\mathcal{Z}$ and any $\mathsf{Z} \in F_{\mathcal{Z}}$. Two complementary cases can occur:

- Either there exists an efficient implementation of an infinitely-often one-way function $\mathsf{ioF}^{\mathsf{Z}}$ relative to Z. By the assumption of Lemma 20, there exists an efficient implementation KA_1 of key agreement relative to $(\mathsf{ioF}', \mathsf{Z})$ for any $\mathsf{ioF}' \in F_{\mathsf{io}\text{-}\mathcal{F}}$. Define the following efficient construction $\mathsf{KA}_2^{\mathsf{Z}}$: $\mathsf{KA}_2^{\mathsf{Z}} := \mathsf{KA}_1^{\mathsf{ioF}^{\mathsf{Z}},\mathsf{Z}}$. By our assumption, this is therefore an efficient implementation of key agreement relative to Z, which is also secure if $(\mathsf{ioF}, \mathsf{Z})$ is secure.
- Or there exists no efficient implementation of an infinitely-often one-way function $\mathsf{ioF}^{\mathsf{Z}}$ relative to Z. By Lemma 21, for a random oracle RO, there must therefore exist an efficient attack on $\mathsf{KA}_1^{\mathsf{RO},\mathsf{Z}}$. By Theorem 5.2 of [20], with measure 1 over the choice of the random oracle RO, RO is a one-way function (and therefore in particular an io-OWF; note that this theorem holds also in the presence of an arbitrary other oracle). Therefore, KA_1 is not a secure implementation of key agreement with respect to any $(\mathsf{ioF}', \mathsf{Z})$ with $\mathsf{ioF}' \in F_{\mathsf{io}\text{-}\mathcal{F}}$, and by the assumptions of Lemma 21, $(\mathsf{RO}, \mathsf{Z})$ is not a secure implementation of $(\mathsf{io}\text{-}\mathcal{F}, \mathcal{Z})$. Since RO is a secure implementation of io-OWF, this implies that Z is not a secure implementation of $\mathcal{Z}$. Therefore, we can define KA_2 to be the trivial protocol in which Alice samples the output key and sends it to Bob. This is an efficient implementation of key agreement (it need not be secure since Z is not a secure implementation of $\mathcal{Z}$).

4.1.3 Proof of Lemma 21

It remains to prove Lemma 21. Consider a candidate construction $\mathsf{KA}^{\mathsf{RO},\mathsf{Z}}$ of key agreement such that one of Alice and Bob makes a constant number of queries to RO. Let $\lambda \in \mathbb{N}$ be the security parameter, and consider a run $(T, K_A, K_B) \xleftarrow{\$} \langle \mathsf{Alice}^{\mathsf{RO},\mathsf{Z}}(1^\lambda), \mathsf{Bob}^{\mathsf{RO},\mathsf{Z}}(1^\lambda) \rangle$ of the construction $\mathsf{KA}^{\mathsf{RO},\mathsf{Z}}$. We will describe an efficient attacker $\mathsf{Eve}^{\mathsf{RO},\mathsf{Z}}$ that breaks $\mathsf{KA}^{\mathsf{RO},\mathsf{Z}}$ for infinitely many λ. The attack closely follows the (inefficient) strategy of [9], but relies on $\mathsf{Inv}^{\mathsf{Z}}$ to make the attack efficient. Without loss of generality, assume that Bob makes a constant number of queries; let q_B be a constant bound on the number of queries made by Bob in any execution of the protocol. Furthermore, let $r_A(\lambda)$ and $r_B(\lambda)$ be (polynomial) bounds on the length of the random tape of Alice and Bob respectively, and let $q(\lambda) = q_A(\lambda) + q_B$ be a (polynomial) bound on the total number of queries to RO made by both parties in any execution.

4.1.3.1 Lazy oracle sampling

Let $q \in \mathbb{N}$. For any string r of length q, and any list L of (query, answer) pairs, we let $\mathsf{SimRO}[L]_q(\cdot; r)$ be a stateful *lazy sampler* for a random oracle consistent with L. Namely, $\mathsf{SimRO}[L]_q(\cdot; r)$ works as follows: it maintains a counter i (initialized to 1) and a list L' of pairs (query, answer), which is initially empty. Each time it receives an input x, $\mathsf{SimRO}[L]_q(x; r)$ first checks whether or not the query belongs to $L \cup L'$, and outputs the corresponding answers if this holds. If the query does not belong to L or L', algorithm $\mathsf{SimRO}[L]_q(x; r)$ defines q_i to be the answer to the query, adds (query, q_i) to L', and sets $i \leftarrow i + 1$. Note that for any interactive protocol Π^{RO} where the parties make less than q queries in total,

and any list L of (query, answer) pairs consistent with RO, the distribution of the views of all parties obtained by sampling a random oracle RO and running Π^{RO} is identical to the distribution of the views of all parties obtained by sampling a q-bit string r and running Π while emulating RO using $\mathsf{SimRO}[L]_q(\cdot; r)$.

4.1.3.2 An inefficient attack

We first describe an inefficient attack on the candidate construction $\mathsf{KA}^{\mathsf{RO},\mathsf{Z}}$, taken almost verbatim from [9]. The attacker $\mathsf{Eve}^{\mathsf{RO},\mathsf{Z}}$, given a transcript T of an execution of $\mathsf{KA}^{\mathsf{RO},\mathsf{Z}}$, maintains a set Q_E of query/answer pairs for Z, and a multi-set of candidate keys $\mathcal{K}$, both initialized to $\emptyset$. Eve runs $2q_B(\lambda)+1$ iterations of the following procedure.

- *Simulation phase:* Eve finds a view of $\mathsf{Alice}^{\mathsf{RO}',\mathsf{Z}}$ with respect to some (possibly different) oracle RO', consistent with the transcript T and all pairs query/answer in Q_E. This view contains a random tape r_A, the set of queries Q_A made by $\mathsf{Alice}^{\mathsf{RO}',\mathsf{Z}}$ (which is consistent with Q_{Eve}, but not necessarily with RO), and the key K_A computed by Alice. Eve adds K_A to $\mathcal{K}$.
- *Update phase:* $\mathsf{Eve}^{\mathsf{RO},\mathsf{Z}}$ makes all queries in Q_A to the true random oracle RO, and adds the results to Q_E.

After running $2q_B(\lambda)+1$ iterations of the above attack, Eve has a multi-set $\mathcal{K}$ of $2q_B+1$ possible keys; Eve outputs the majority value in $\mathcal{K}$. Observe that during each round of the attack, two events can happen:

1. Either one of the new queries (not already contained in Q_E) made by Alice in the simulated run was made by Bob in the real execution of the protocol. In this case, Eve discovers (and adds to Q_E) a new query of Bob.
2. Or none of the new queries of Alice was made by Bob in the real protocol, in which case there exists an oracle RO' which is consistent with the view of Bob in the real protocol, and the view of Alice in the simulated run. By perfect correctness, this means that the key K_A computed by Alice in this run is necessarily the correct key.

Now, since Bob makes at most q_B distinct queries, the first of the two events can happen at most q_B times, hence the second event necessarily happens at least q_B+1 times, which guarantees that the majority value in the multi-set $\mathcal{K}$ is indeed the correct key with probability 1.

The above attack requires $\mathcal{O}(q_A q_B)$ queries to RO. However, it requires to find a view for $\mathsf{Alice}^{\mathsf{RO}',\mathsf{Z}}$ consistent with a given transcript, where RO' is a simulated random oracle, but Z is a "true" auxiliary oracle. In general, this might require exponentially many queries to Z, hence $\mathsf{Eve}^{\mathsf{RO},\mathsf{Z}}$ is not an efficient oracle-aided algorithm. In the following, we show how to make the attack efficient given an inverter for io-OWFs.

4.1.3.3 Lazy protocol emulation

Let L be a list of (query, answer) pairs to RO. Given the construction $\mathsf{KA}^{\mathsf{RO},\mathsf{Z}}$, let $\mathsf{SimC}^{\mathsf{Z}}_L$ be an oracle-aided PPT algorithm that emulates a run of Alice and Bob in $\mathsf{KA}^{\mathsf{RO},\mathsf{Z}}$ that is consistent with L (but not necessarily with the rest of RO). That is, given a random string $r_A||r_B||q$ of length $r_A(\lambda)+r_B(\lambda)+q(\lambda)$, $\mathsf{SimC}^{\mathsf{Z}}_L(1^\lambda; r_A||r_B||q)$ does the following: it runs Alice and Bob with input 1^λ and respective random tapes r_A and r_B, while using $\mathsf{SimRO}[L]_q(\cdot; q)$ to lazily emulate the random oracle RO. After completion of the protocol, SimC outputs the transcript T of the interaction, the lists (Q_A, Q_B) of all queries to RO made by Alice and Bob during the emulation of the protocol (together with their answers), and the outputs (K_A, K_B) of both parties. Observe that SimC corresponds to a valid interaction between $\mathsf{Alice}^{\mathsf{RO}'}(1^\lambda; r_A)$ and $\mathsf{Bob}^{\mathsf{RO}'}(1^\lambda; r_B)$ with respect to a random oracle RO' sampled uniformly at random, conditioned on being consistent with L.

4.1.3.4 The inverter

Since there is no infinitely-often OWF relative to Z, there exists an efficient inverter for any efficient oracle-aided function:

▶ **Lemma 22.** *For any oracle-aided PPT function F^{Z} and any polynomial p, there exists an oracle-aided PPT inverter $\mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F},p}$ such that for all large enough $n \in \mathbb{N}$, it holds that*

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n}[\mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F},p}(\mathsf{F}^{\mathsf{Z}}(x), 1^n) \in (\mathsf{F}^{-1})^{\mathsf{Z}}(\mathsf{F}^{\mathsf{Z}}(x))] \geq 1 - \frac{1}{p(n)} .$$

Proof. This follows directly from the fact that the inexistence of io-OWFs relative to Z implies the inexistence of *weak* io-OWFs relative to Z (a weak OWF is a OWF where security is relaxed by saying that there exists a polynomial p such that no efficient adversary can invert with probability better than $1 - 1/p(n)$). The latter follows from standard hardness amplification methods as was initially proven by Yao [29]. ◀

4.1.3.5 The sequence of functions

Let $p : \lambda \to 1/(6q_B + 3)$ be a constant polynomial. We iteratively define a sequence of $2(q_B + 1)$ oracle functions $(\mathsf{F}^{\mathsf{Z}}_0, \mathsf{G}^{\mathsf{Z}}_0), \cdots, (\mathsf{F}^{\mathsf{Z}}_{q_B}, \mathsf{G}^{\mathsf{Z}}_{q_B})$ as follows.

- $\mathsf{F}^{\mathsf{Z}}_0$ gets as input a string $(r_A||r_B||q_0)$ of length $r_A(\lambda) + r_B(\lambda) + q(\lambda)$, computes
$$(T, Q_A, Q_B, K_A, K_B) \leftarrow \mathsf{SimC}^{\mathsf{Z}}_{\emptyset}(1^\lambda; r_A||r_B||q) ,$$
and outputs T. The function $\mathsf{G}^{\mathsf{Z}}_0$ is defined similarly, but outputs (T, Q_A, Q_B, K_A).
- $\mathsf{F}^{\mathsf{Z}}_1$ gets as input a string $(r_A||r_B||q_0||q_1)$ of length $r_A(\lambda)+r_B(\lambda)+2q(\lambda)$. First, it computes $(T, Q_A, Q_B, K_A) \leftarrow \mathsf{G}^{\mathsf{Z}}_0(r_A||r_B||q_0)$. Second, it sets $n \leftarrow r_A(\lambda) + r_B(\lambda) + q(\lambda)$ and runs $(r'_A||r'_B||q'_0) \leftarrow \mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}_0,p}(T, 1^n)$. Third, it computes $(T', Q'_A, Q'_B, K'_A) \leftarrow \mathsf{G}^{\mathsf{Z}}_0(1^\lambda; r'_A||r'_B||q'_0)$. Eventually, it uses $\mathsf{SimRO}[Q_A \cup Q_B](\cdot; q_1)$ to lazily sample the answers to all queries contained in Q'_A, and stores the results in a set Q_E of pairs query/answer. $\mathsf{F}^{\mathsf{Z}}_1$ outputs (T, Q_E). We also define $\mathsf{G}^{\mathsf{Z}}_1$ to be the function defined as $\mathsf{F}^{\mathsf{Z}}_1$ except that it additionally outputs (Q_A, Q_B, K'_A).
- $\mathsf{F}^{\mathsf{Z}}_i$ gets as input a string $(r_A||r_B||q_0||\cdots||q_i)$ of length $r_A(\lambda) + r_B(\lambda) + (i+1) \cdot q(\lambda)$. First, it computes $(T, Q_E, Q_A, Q_B) \leftarrow \mathsf{G}^{\mathsf{Z}}_{i-1}(r_A||r_B||q_0||\cdots||q_{i-1})$. Second, it sets $n \leftarrow r_A(\lambda) + r_B(\lambda) + i \cdot q(\lambda)$ and runs $(r'_A||r'_B||q'_0||\cdots||q'_{i-1}) \leftarrow \mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}_{i-1},p}((T, L_{i-1}), 1^n)$. Third, it computes $(T', Q'_A, Q'_B) \leftarrow \mathsf{G}^{\mathsf{Z}}_0(1^\lambda; r'_A||r'_B||q'_0)$. Eventually, it uses $\mathsf{SimRO}[Q_A \cup Q_B](\cdot; q_i)$ to lazily sample the answers to all queries contained in Q'_A, and add the results to Q_E. $\mathsf{F}^{\mathsf{Z}}_2$ outputs (T, Q_E). We also define $\mathsf{G}^{\mathsf{Z}}_i$ to be the function defined as $\mathsf{F}^{\mathsf{Z}}_i$ except that it additionally outputs (Q_A, Q_B, K'_A).

For readability, we also provide a pseudocode for the function $\mathsf{F}^{\mathsf{Z}}_i$ below:

function $\mathsf{F}^{\mathsf{Z}}_i(r_A||r_B||q_0||\cdots||q_i)$ ▷ $(r_A||r_B||q_1||\cdots||q_i)$ is of length $r_A(\lambda) + r_B(\lambda) + (i+1) \cdot q(\lambda)$
  $(T, Q_E, Q_A, Q_B, K_A) \leftarrow \mathsf{G}^{\mathsf{Z}}_{i-1}(r_A||r_B||q_1||\cdots||q_{i-1})$
  $n \leftarrow r_A(\lambda) + r_B(\lambda) + i \cdot q(\lambda)$
  $(r'_A||r'_B||q'_0||\cdots||q'_{i-1}) \leftarrow \mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}_{i-1},p}((T, Q_E), 1^n)$ ▷ $p : \lambda \to 1/(6q_B + 3)$
  $(T', Q'_A, Q'_B, K'_A) \leftarrow \mathsf{G}^{\mathsf{Z}}_1(1^\lambda; r'_A||r'_B||q'_0)$
  for $(x, y) \in Q'_A$ **do**
    $Q_E \leftarrow Q_E \cup (x, \mathsf{SimRO}[Q_A \cup Q_B](x; q_i))$
  end for
  return (T, Q_E) ▷ $\mathsf{G}^{\mathsf{Z}}_i$ is similar but additionally outputs (Q_A, Q_B, K'_A)
end function

4.1.3.6 Making the [9] attack efficient with Inv

To overcome the inefficiency of the attack of [9], we leverage the fact that, by assumption, there exists no infinitely-often one-way function relative to Z. As before, the attacker $\mathsf{newEve}^{\mathsf{RO},\mathsf{Z}}$, given a transcript T of an execution of $\mathsf{KA}^{\mathsf{RO},\mathsf{Z}}$, maintains a set Q_E of query/answer pairs for Z, and a multi-set of candidate keys $\mathcal{K}$, both initialized to $\emptyset$. Let $p : \lambda \to 1/(6q_B + 3)$ be a constant polynomial. newEve runs $2q_B + 1$ iterations of the following attack.

- *Simulation phase:* During the i-th round of attack, newEve does the following:
  1. *Finding a view of* Alice *consistent with* T *and* Q_E*:* newEve sets $n \leftarrow r_A(\lambda) + r_B(\lambda) + i \cdot q(\lambda)$ and computes $(r'_A||r'_B||q'_0|| \cdots ||q'_{i-1}) \leftarrow \mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}_{i-1},p}((T, Q_E); 1^n)$.
  2. *Simulating the run of Alice with the view above:* newEve computes $(T', Q'_A, Q'_B, K'_A) \leftarrow \mathsf{G}^{\mathsf{Z}}_1(1^\lambda; r'_A||r'_B||q'_0)$.
  3. *Storing the key:* newEve adds K'_A to $\mathcal{K}$.
- *Update phase:* $\mathsf{Eve}^{\mathsf{RO},\mathsf{Z}}$ makes all queries in Q'_A to the true random oracle RO, and adds the results to Q_E.

After running $2q_B + 1$ iterations of the above attack, Eve has a multi-set $\mathcal{K}$ of $2q_B + 1$ possible keys; Eve outputs the majority value in $\mathcal{K}$. We now analyze the success probability of the attack.

▷ Claim 23. newEve outputs the correct key with probability at least $2/3$.

First, observe that by definition of $\mathsf{F}^{\mathsf{Z}}_0$, the distribution of transcripts T of the real execution of the protocol (which newEve gets as input) is distributed exactly as $\mathsf{F}^{\mathsf{Z}}_0(r_A||r_B||q_0)$ for uniformly random (r_A, r_B, q_0), where r_A (resp., r_B) is the real random tape of $\mathsf{Alice}^{\mathsf{RO},\mathsf{Z}}$ (resp., $\mathsf{Bob}^{\mathsf{RO},\mathsf{Z}}$) and q_0 is the ordered string of all answers of RO to distinct queries from Alice and Bob. Therefore, by definition of $\mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}_0,p}$, the tuple $(r'_A||r'_B||q'_0)$ computed in the first iteration of the attack is consistent with the real transcript T with probability at least $p = 1/(6q_B + 3)$.

Consider now the set Q_E of queries obtained by newEve after the update phase of the first iteration. (T, Q_E) is distributed exactly as $\mathsf{F}^{\mathsf{Z}}_1(r_A||r_B||q_0||q_1)$ for uniformly random (r_A, r_B, q_0, q_1). This is because Q_E in $\mathsf{F}^{\mathsf{Z}}_1$ is computed by lazily sampling the answers of a random oracle, conditioned on being consistent with all queries made by Alice and Bob in the execution of $\mathsf{F}^{\mathsf{Z}}_0(r_A||r_B||q_0)$. Since the real run of the protocol corresponds to an execution of $\mathsf{F}^{\mathsf{Z}}_0$ on a random input $(r_A||r_B||q_0)$, and making the queries in Q'_A to RO is identical to lazily sampling RO while being consistent with q_0 (i.e., the query/answer pairs obtained by Alice and Bob in the real execution). Therefore, the tuple $(r'_A||r'_B||q'_0||q'_1)$ which newEve computes in the second iteration of the attack is consistent with the real transcript T with probability at least $p = 1/(6q_B + 3)$.

More generally, the distribution of (T, Q_E) obtained by newEve during the i-th iteration of the attack after receiving the transcript T of a real execution of the protocol is distributed exactly as $\mathsf{F}^{\mathsf{Z}}_i(r_A||r_B||q_0|| \cdots ||q_i)$ for uniformly random $(r_A, r_B, q_0, \cdots, q_i)$, hence the tuple $(r'_A||r'_B||q'_0|| \cdots ||q'_i)$ which newEve computes in the $(i+1)$-th iteration of the attack is consistent with the real transcript T with probability at least $p = 1/(6q_B + 3)$.

Putting everything together, after finishing the attack, by a straightforward union bound, all simulated views computed by newEve during the attack are consistent with T with probability at least $1 - (2q_B + 1) \cdot 1/(6q_B + 3) = 2/3$. When this happens, by the same argument as for the inefficient attack, the majority key in $\mathcal{K}$ is necessarily the correct key, and the claim follows.

▷ Claim 24. The number of queries made by newEve to RO and Z is bounded by a polynomial.

As in the inefficient attack, newEve makes at most $\mathcal{O}(q_A q_B) = \mathcal{O}(q_A)$ queries to RO. The polynomial bound on the number of queries to Z follows from the efficiency of Inv: $\mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}_0,p}$ is efficient by definition, hence $\mathsf{F}^{\mathsf{Z}}_1$ (which invokes $\mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}_0,p}$ internally) is an efficient function, from which we get that $\mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}_1,p}$ is also efficient, and so on, and the claim follows.

4.2 Black-Box Uselessness of OWFs for Imperfect KA

In this section, we discuss how our result of the previous section can be extended to the case of imperfect key agreement. More precisely, we provide a sketch of how to modify our previous proof to show the following:

▶ **Theorem 25.** *Infinitely-often one-way functions are [semi $\to \forall\exists$-semi] black-box useless for infinitely-often perfect key agreement in any construction where:*

- *Both parties make a constant number of queries to the io-OWF (but any polynomial number of queries to the auxiliary oracle)*
- *The key agreement protocol has a constant number of rounds.*

Proof Sketch. The natural approach to extend our result is to replace the attack of [9] by an attack that applies to any imperfect key agreement protocol in the random-oracle model, such as those of Impagliazzo and Rudich [20] or Barak and Mahmoody [4], making the same case distinction based on the existence of io-OWFs to make the attack efficient. The structure of these attacks are very similar, though more involved than the attack of [9]: they proceed in a sequence of steps, where each step has a simulation phase, in which the attacker samples views consistent with (a portion of) the transcript and a set of queries, and an update phase, where the attacker makes some queries based on the simulated run.

Two important technicalities arise when modifying our previous proof with the attacks of [4, 20]:

First, in the simpler attack of [9], the perfect correctness guarantees that finding *any* consistent view is sufficient; in the attacks of [4, 20], however, the attacker is required to *sample* views from a distribution close to the uniform distribution over views conditioned on a transcript and a set of queries. This can still be achieved assuming only the inexistence of io-OWFs: the inexistence of io-OWFs further entails the inexistence of *distributional* io-OWFs [19]. Namely, we must rely on the following lemma, a proof of which can be found in [6].

▶ **Lemma 26.** *Assume that there exists no infinitely-often one-way function relative to* Z*. Then for any efficient oracle-aided function* F^{Z} *and any polynomial* p*, there exists an inverter* $\mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}}$ *such that for all large enough* $n \in \mathbb{N}$*,*

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n, y \leftarrow \mathsf{F}^{\mathsf{Z}}(x)} \left[\mathsf{SD}\left((\mathsf{F}^{-1})^{\mathsf{Z}}(y), \mathsf{Inv}^{\mathsf{Z}}_{\mathsf{F}}(y, 1^n) \right) > \frac{1}{p(n)} \right] \leq \frac{1}{p(n)} \ ,$$

where SD *denotes the statistical distance between the two distributions.*

Second, the attacks of [4, 20] proceed in a number of steps that grows with the number of queries of *both* parties (to the random oracle) *and* the round complexity of the protocol. More precisely, the attack requires executing several simulation and updates phases (of the order of $\tilde{\mathcal{O}}(q_A q_B)$, where q_A, q_B bound the respective number of queries of Alice and Bob to the random oracle) for each round of the protocol, where the simulation phase for the round i inverse samples views consistent with the set of queries made by the attacker so far and the transcript of the protocol *up to round* i. This means that to be carried efficiently, the key agreement must be constant round, and both parties must make a constant number of queries to the random oracle.

From here, a proof of Theorem 25 follows by fixing a (constant) bound B on the total number of steps of the (inefficient) attacker, and using the inverse sampler guaranteed by Lemma 26 for statistical distance $1/(10B)$ to make it efficient, similarly as in our previous proof. By a union bound over all steps, the B steps of the inverse sampling will simultaneously guarantee that, with probability at least $1/10$, at any round i, no intersection query between Alice and Bob made prior to round i has been missed by the attacker. This allows us to conclude that the overall success probability of the efficient attacker (which uses the inverse sampler) is at most a tenth of the success probability of the inefficient attacker described in [4, 20].

With these technicalities in mind, this proof shows that io-OWFs are [semi $\to \forall\exists$-semi] black-box useless for constant-query constant-round constructions of imperfect key agreement. ◀

References

1 Shashank Agrawal, Venkata Koppula, and Brent Waters. Impossibility of simulation secure functional encryption even with random oracles. In *Theory of Cryptography Conference*, pages 659–688. Springer, 2018.

2 Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 191–209, Berkeley, CA, USA, October 17–20 2015. IEEE Computer Society Press. doi:10.1109/FOCS.2015.21.

3 Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 296–315, Bengalore, India, December 1–5 2013. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-42033-7_16.

4 Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 374–390, Santa Barbara, CA, USA, August 16–20 2009. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-03356-8_22.

5 Balthazar Bauer, Pooya Farshim, and Sogol Mazaheri. Combiners for backdoored random oracles. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 272–302, Santa Barbara, CA, USA, August 19–23 2018. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-96881-0_10.

6 Itay Berman, Iftach Haitner, and Aris Tentes. Coin flipping of *any* constant bias implies one-way functions. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 398–407, New York, NY, USA, May 31 – June 3 2014. ACM Press. doi:10.1145/2591796.2591845.

7 Nir Bitansky, Huijia Lin, and Omer Paneth. On removing graded encodings from functional encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–29. Springer, 2017.

8 Zvika Brakerski, Christina Brzuska, and Nils Fleischhacker. On statistically secure obfuscation with approximate correctness. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 551–578, Santa Barbara, CA, USA, August 14–18 2016. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-53008-5_19.

9 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: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 559–578, Providence, RI, USA, March 28–30 2011. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-19571-6_34.

10 Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On obfuscation with random oracles. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 456–467, Warsaw, Poland, March 23–25 2015. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-46497-7_18.

11 Kai-Min Chung, Huijia Lin, Mohammad Mahmoody, and Rafael Pass. On the power of nonuniformity in proofs of security. In Robert D. Kleinberg, editor, *ITCS 2013: 4th Innovations in Theoretical Computer Science*, pages 389–400, Berkeley, CA, USA, January 9–12 2013. Association for Computing Machinery. doi:10.1145/2422436.2422480.

12 Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ameer Mohammed. Limits on the power of garbling techniques for public-key encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 335–364, Santa Barbara, CA, USA, August 19–23 2018. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-96878-0_12.

13 Sanjam Garg, Mohammad Mahmoody, and Ameer Mohammed. Lower bounds on obfuscation from all-or-nothing encryption primitives. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 661–695, Santa Barbara, CA, USA, August 20–24 2017. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-63688-7_22.

14 Sanjam Garg, Mohammad Mahmoody, and Ameer Mohammed. Lower bounds on obfuscation from all-or-nothing encryption primitives. In *Draft of the full version*, 2017. URL: http://www.cs.virginia.edu/~mohammad/files/papers/IO-all-or-nothing.pdf.

15 Sanjam Garg, Mohammad Mahmoody, and Ameer Mohammed. When does functional encryption imply obfuscation? In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 82–115, Baltimore, MD, USA, November 12–15 2017. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-70500-2_4.

16 Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM journal on Computing*, 35(1):217–246, 2005.

17 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 Annual Symposium on Foundations of Computer Science*, pages 669–679, Providence, RI, USA, October 20–23 2007. IEEE Computer Society Press. doi:10.1109/FOCS.2007.27.

18 Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 850–858, Paris, France, October 7–9 2018. IEEE Computer Society Press. doi:10.1109/FOCS.2018.00085.

19 Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th Annual Symposium on Foundations of Computer Science*, pages 230–235, Research Triangle Park, NC, USA, October 30 – November 1 1989. IEEE Computer Society Press. doi:10.1109/SFCS.1989.63483.

20 Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st Annual ACM Symposium on Theory of Computing*, pages 44–61, Seattle, WA, USA, May 15–17 1989. ACM Press. doi:10.1145/73007.73012.

21 Mohammad Mahmoody and Ameer Mohammed. On the power of hierarchical identity-based encryption. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 243–272, Vienna, Austria, May 8–12 2016. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-49896-5_9.

22 Mohammad Mahmoody, Ameer Mohammed, Soheil Nematihaji, Rafael Pass, and abhi shelat. Lower bounds on assumptions behind indistinguishability obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 49–66, Tel Aviv, Israel, January 10–13 2016. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-49096-9_3.

23 Mohammad Mahmoody, Ameer Mohammed, Soheil Nematihaji, Rafael Pass, and abhi shelat. A note on black-box separations for indistinguishability obfuscation. Cryptology ePrint Archive, Report 2016/316, 2016. URL: http://eprint.iacr.org/2016/316.

24 Mohammad Mahmoody, Tal Moran, and Salil P. Vadhan. Time-lock puzzles in the random oracle model. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 39–50, Santa Barbara, CA, USA, August 14–18 2011. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-22792-9_3.

25 Hemanta K. Maji and Mingyuan Wang. Black-box use of one-way functions is useless for optimal fair coin-tossing. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 593–617, Santa Barbara, CA, USA, August 17–21 2020. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-56880-1_21.

26 Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20, Cambridge, MA, USA, February 19–21 2004. Springer, Heidelberg, Germany. doi:10.1007/978-3-540-24638-1_1.

27 Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 475–484, New York, NY, USA, May 31 – June 3 2014. ACM Press. doi:10.1145/2591796.2591825.

28 Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345, Espoo, Finland, May 31 – June 4 1998. Springer, Heidelberg, Germany. doi:10.1007/BFb0054137.

29 Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, Chicago, Illinois, November 3–5 1982. IEEE Computer Society Press. doi:10.1109/SFCS.1982.45.