On Composing Oracle Separations

Willy Quach and Felix Rohrbach

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
{willy.quach, felix.rohrbach}@cispa.de

Abstract

Black-box separations constitute very powerful technical barriers against constructing primitives from specific building blocks. However, they give surprisingly little indication on the usefulness of *combining* several such building blocks in a single construction.

We propose a set of technical conditions, under which (fully) black-box separations compose with each other. That is, if two black-box separations $\mathcal{P}_1 \overset{BB}{\not\Rightarrow} \mathcal{Q}$ and $\mathcal{P}_2 \overset{BB}{\not\Rightarrow} \mathcal{Q}$ satisfy our conditions, then $\mathcal{Q}$ is also separated from black-box combinations of $\mathcal{P}_1$ and $\mathcal{P}_2$. When applicable, these conditions can be established without unpacking the full technical details of the base separations.

As our main application, we show that combining indistinguishability obfuscation (iO) with collision resistant hash functions is unlikely to yield hard problems in the complexity class SZK, and thus unlikely to suffice for primitives such as lossy functions or fully homomorphic encryption. Previous lower bounds on iO only applied to combinations of iO with one-way permutations (Asharov-Segev, FOCS '15 and Bitansky-Degwekar-Vaikuntanathan, SIAM Journal on Computing '21).

1 Introduction

Black-box impossibility results are central to our understanding of cryptography. Because of our inability to formally prove cryptographic assumptions, they provide rare insights into how various cryptographic notions are likely *hard to build* (at least given current techniques) or *differ* from each other.

Oracle separations (which we will also call black-box separations throughout this paper) are arguably the most common kind of such lower bounds

in cryptography. In a nutshell, an oracle separation of a "target" primitive $\mathcal{Q}$ (such as key agreement) from a "source" primitive $\mathcal{P}$ (such as one-way functions) rules out black-box constructions of $\mathcal{Q}$ from $\mathcal{P}$.[1] Since a large majority of existing cryptographic constructions are black-box, such separations often indicates strong technical barriers against building $\mathcal{Q}$ from $\mathcal{P}$. In fact, most of these technical barriers have persisted to this day.[2]

This methodology has been remarkably successful. The seminal work of Impagliazzo and Rudich [IR89] separated key-agreement from one-way functions, thus indicating that "Minicrypt" and "Cryptomania" are plausibly distinct worlds in the sense of Impagliazzo [Imp95]. Since then, many further separations have developed a rich web of black-box separations between cryptographic primitives, e.g. [Sim98, GT00, HR04, HHRS07, BKSY11, AS15, BMM24] just to cite a few.

One drawback of oracle separations is that they are typically very *monolithic*: they separate a specific target primitive $\mathcal{Q}$ from a specific source primitive $\mathcal{P}$. Consequently, whenever a target primitive $\mathcal{Q}$ is independently separated from two source primitives $\mathcal{P}_1$ and $\mathcal{P}_2$, it is not clear in general whether the latter can be meaningfully *combined* in a black-box way to overcome the individual separations. In fact, modern cryptographic constructions provide ample evidence that combining different primitives often enables powerful applications. This overall points to a potential blind spot in these impossibility results, and raises the following question:

Can black-box separations be overcome by combining *building blocks in a black-box manner?*

Unsurprisingly, the answer depends on the setting. For instance, whenever $\mathcal{P}_1$ and $\mathcal{P}_2$ can be constructed in a black-box way from a primitive $\mathcal{P}$, and $\mathcal{Q}$ is separated from $\mathcal{P}$, then $\mathcal{Q}$ is also separated from black-box combinations of $\mathcal{P}_1$ and $\mathcal{P}_2$. For example, key-agreement is separated from black-box combinations of PRFs and signatures, as a consequence of being separated from one-way functions [IR89].

Conversely, whenever $\mathcal{P}_1$ and $\mathcal{P}_2$ are mutually separated from each other, then the "joint primitive" combining $\mathcal{P}_1$ and $\mathcal{P}_2$ is also separated from $\mathcal{P}_1$ and $\mathcal{P}_2$ taken individually.[3] However, "joining" $\mathcal{P}_1$ and $\mathcal{P}_2$ together trivially yields the joint primitive in a black-box way. More concretely, the

[1] There exist several distinct notions of oracle separations and black-box constructions [RTV04, BBF13], which we will ignore for the moment.

[2] In some rare cases, these technical barriers have been overcome. We refer to the related work section for more discussion.

[3] Here, we refer to the "joint" primitive as consisting of the algorithms of both $\mathcal{P}_1$

joint primitive combining collision-resistant hashing (CRH) and public-key encryption (PKE) is respectively separated from public-key encryption (as CRH is separated from PKE [AS15]) and from collision-resistant hashing (as PKE is separated from CRH [IR89]). Nevertheless, joining them together trivially yields the joint primitive, and thus overcomes the two individual separations [CFM21].

Intriguingly, our understanding beyond these simple cases is fairly poor. To the best of our knowledge, the only explicit progress on this question is due to Couteau, Farshim and Mahmoody [CFM21], who introduce a strong notion of black-box separations called black-box uselessness. It ensures that some primitive $\mathcal{P}_1$ cannot be meaningfully combined with *any* other primitive $\mathcal{P}_2$ in a black-box way to build $\mathcal{Q}$. In other words, no separation $\mathcal{P}_2 \overset{BB}{\not\Rightarrow} \mathcal{Q}$ can be overcome with the help of a primitive $\mathcal{P}_1$ that is black-box useless for $\mathcal{Q}$. This can be viewed as strong form of *composability* of the separation $\mathcal{P}_1 \overset{BB}{\not\Rightarrow} \mathcal{Q}$. Unfortunately, the notion of black-box uselessness seems very strong and hard to achieve. Thus, establishing lower bounds for combining primitives in constructions remains broadly open:

In which settings do black-box separations compose with each other?

1.1 Our Contributions

1.1.1 A Criterion to Compose (Fully) Black-Box Separations.

Our main contribution is a set of conditions under which specific black-box separations compose with each other. In other words, if a target primitive $\mathcal{Q}$ is black-box separated from source primitives $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively, and moreover both black-box separations satisfy these technical conditions, then $\mathcal{Q}$ is also black-box separated from black-box combinations of $\mathcal{P}_1$ and $\mathcal{P}_2$.

As previously discussed, because black-box separations do not always compose, some restrictions on the base separations are necessary. From this perspective, our contribution is to identify sufficient conditions that capture the intuition that different black-box separations are compatible with each other.

We now state these technical conditions in more detail. They apply to *fully black-box separations* [RTV04] proven using the so-called "'two-oracle" technique [HR04]. This is a technique used to separate a target primitive $\mathcal{Q}$ from a source primitive $\mathcal{P}$, by constructing a constructive oracle Const and a

and $\mathcal{P}_2$, along with their individual security properties; see Definition 2.2 for a formal definition.

breaker oracle Break, such that intuitively, Break breaks any instantiation of $\mathcal{Q}$ (in an oracle world with Const and Break), while Const provides a secure realization of $\mathcal{P}$ even in the presence of Break. We can now state our main result:

Theorem 1.1 (Informal)**.** *Suppose $\mathcal{Q}$ is black-box separated from $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) using the two-oracle technique, with oracles Const_1 and Break_1 (resp. Const_2 and Break_2). Suppose the two following conditions hold:*

1. Identical, Relativizing Break Oracles: *both separations use the same breaker oracle, namely $\mathsf{Break}_1 = \mathsf{Break}_2$, and moreover, it breaks $\mathcal{Q}$ in a relativizing manner;*

2. Simulatable Const Oracles (Simplified):[4] *any (relativizing) algorithm $\mathcal{A}^{\mathsf{Const}_1}$ with oracle access to Const_1 (resp. $\mathcal{A}^{\mathsf{Const}_2}$ having access to Const_2) can be efficiently simulated by a (relativizing) algorithm $\mathcal{B}$ without access to Const_1 (resp. Const_2),*

then $\mathcal{Q}$ is separated from black-box combinations of $\mathcal{P}_1$ and $\mathcal{P}_2$.

Moreover, the latter separation is itself a two-oracle separation with oracles $(\mathsf{Const}_1, \mathsf{Const}_2)$ and Break that satisfies conditions 1 and 2.

We give now a brief overview of these conditions, and refer to the technical overview for more details. First, the condition that $\mathsf{Break}_1 = \mathsf{Break}_2$ are identical is the core conceptual ingredient capturing that the two separations from $\mathcal{P}_1$ and $\mathcal{P}_2$ are "compatible": it is the only condition in Theorem 1.1 that relates both base separations. The relativization condition for Break helps formalize that "Break strongly breaks $\mathcal{Q}$". This is both a useful technical condition in our setting, and helps formalize the role of the breaker oracle. In particular, this prevents setting the constructive oracle as $(\mathsf{Const}, \mathsf{Break})$ and the breaker oracle as the empty oracle, and consequently makes this first condition non-trivial.

We view this first condition as constraining, but fairly natural. Indeed, separations proven via the two-oracle technique for a given target primitive $\mathcal{Q}$ typically use the same breaker oracle which breaks $\mathcal{Q}$ in a relativizing manner. As important examples, separations to collision resistance most often use a breaker oracle $\mathsf{ColFind}$ that outputs random collisions for its input circuit [Sim98, AS15, HL18], and separations to the hardness of the

[4] Theorem 1.1 states a simplified condition 2. Technically, we additionally need to compile out oracle calls made by Break to Const, but omit this here for simplicity. We refer to the body of the paper for more details.

complexity class SZK use a breaker oracle SDO that decides whether the statistical distance between two input circuits is large or small [BDV17, BD19]. In both cases, the breaker oracles naturally relativize to handle oracle-aided circuits.

The second condition is slightly more technical, and is reminiscent of the existing technique of "compiling out oracles" used in e.g. [GGKT05, CKP15, GMM17, GHMM18, CFM21]. The main advantage of our formulation is that prior works often used this technique to compile out oracles *in constructions*, and thus needed to additionally ensure that doing so also preserved various *security and efficiency properties*. In contrast, our simulation is only required to preserve correctness of the algorithm and a weaker notion of efficiency. Looking ahead, this is because, in our proof, we only compile oracles out of *adversaries* instead of constructions. Most notably, our simulation is only required to preserve relativized query complexity, as opposed to the standard running time.[5] On the technical side, we leverage this feature to hard-code, for any fixed oracle Const, its whole truth table in $\mathcal{B}$.[6]

Overall, we believe that these two conditions are commonly satisfied for natural black-box separations to a given target primitive $\mathcal{Q}$ established using a two-oracle approach.

An additional attractive feature is that our conditions are easy to establish: for the most part, they only require knowledge of the two oracles used in the base separations, and no further details from the underlying proofs. The main exception is the condition that Break breaks $\mathcal{Q}$ in a relativizing way. Still, most often, the breaker oracle Break breaks $\mathcal{Q}$ generically given any input candidate construction, and thus readily relativizes.

Additionally, our result is not limited to a single composition: because the separation resulting from Theorem 1.1 satisfies conditions 1 and 2, it can be further composed with other separations involving the same breaker oracle Break.

As a first application, we show that the two following separations smoothly compose via our conditions: the separation of hardness in SZK from one-way permutations [BDV17] and the separation of hardness in SZK from collision-resistant hash functions [BD19].

[5] This is an important difference with the work of [CFM21]; see Section 1.2 for a more detailed comparison.

[6] This still doesn't make simulation entirely trivial, due to the presence of Break which can also make Const calls; see Footnote 4.

Theorem 1.2 (Informal)**.** *There are no fully-black-box reductions of (worst-case) hard problems in* SZK *to the existence of one-way permutations and collision-resistant hash functions.*

1.1.2 Main Application: Separating Hardness of SZK from iO & CRH.

Indistinguishability obfuscation (iO) is often considered a "crypto-complete" primitive: most cryptographic applications are known to be realizable with the help of iO. However, because iO exists if $\mathsf{P} = \mathsf{NP}$, iO needs a "cryptographic crutch" for most cryptographic applications, typically via the existence of one-way functions. Since then, black-box separations have helped paint a finer-grained picture: combining iO and one-way permutations is unlikely to give constructions of collision resistant hashing [AS15, HL18] nor lossy functions or fully homomorphic encryption (FHE) [BDV17]. Put differently, in order to build most advanced cryptographic primitives, iO likely needs to be paired with cryptography beyond plain one-way functions (or permutations).

This leaves the following intriguing question broadly open:

What minimal cryptographic assumptions are required to unlock the full power of iO?

Given the state of the art on black-box separations involving iO, it is plausible that combining iO with collision-resistant hashing, or even distributional collision resistant hashing [DI06], would be sufficient to obtain most known applications of iO.

As our main application of our composition framework, we rule out this possibility, and show that combining iO and collision resistant hashing is unlikely to yield constructions of FHE or lossy functions:

Theorem 1.3 (Informal)**.** *There are no fully-black-box reductions of (worst-case) hard problems in* SZK *to the existence of* iO *and collision-resistant hash functions.*

Here, black-box is in the sense of Asharov-Segev [AS15]: iO is allowed to obfuscate and evaluate circuits with CRH gates, and the construction is otherwise (fully) black-box in the standard sense.

We obtain Theorem 1.3 by composing two black-box separations based on the two-oracle technique: one separating the hardness of SZK from collision-resistant hashing [BD19], and one separating the hardness of SZK from iO (and one-way permutations) [BDV17]. To do so, we generalize the

conditions of Theorem 1.1 to allow one of the base primitives $\mathcal{P}_1$ to handle and evaluate the other primitive $\mathcal{P}_2$ in a slightly non-black-box way, which is crucial to allow the iO to obfuscate the CRH. This leads to slightly more technical versions of Condition 2 than in Theorem 1.1. We refer to Section 5 for more details.

As an additional demonstration of the flexibility of our framework, we further compose the separation of Theorem 1.3 to obtain the following:

Theorem 1.4 (Informal)**.** *There are no fully-black-box reductions of (worst-case) hard problems in* SZK *to the existence of* iO*, collision-resistant hash functions and (unkeyed) one-way permutations.*[7]

Again, black-box here is in the sense of Asharov-Segev, where iO can obfuscate circuits with now both CRH and OWP gates.

Overall, we hope that Theorem 1.3 (and Theorem 1.4) reinforce the understanding that while iO is extremely strong *when combined with other cryptographic primitives*, achieving strong applications such as lossy functions or FHE requires the use of adequately strong assumptions, at least stronger than CRH. For comparison, known constructions of FHE from iO [CLTV15] build on (a strong form of) rerandomizable encryption. We leave the possibility of building FHE from iO and weaker assumptions than in [CLTV15], or ruling out such possibility, as an exciting open problem.

We believe our work also raises the intriguing possibility that Theorem 1.3 and Theorem 1.4 could be part of a larger phenomenon, namely, that iO is broadly unable to help circumvent two-oracle-based black-box separations, beyond what is achievable from iO and OWPs alone. Put differently, iO (along with OWPs) could plausibly be "black-box useless" when restricted to two-oracle-based separations in the following sense: if iO and a primitive $\mathcal{P}$ can be combined to circumvent such a separation $\mathcal{P} \overset{BB}{\not\Rightarrow} \mathcal{Q}$, then iO and OWPs suffice to build $\mathcal{Q}$ without $\mathcal{P}$. We leave finding a counterexample to this phenomenon (or proving it!) as an interesting open question.

1.2 Related Work.

1.2.1 Comparison with Black Box Uselessness.

Couteau, Farshim and Mahmoody [CFM21] introduced the notion of black-box uselessness, which captures that a primitive $\mathcal{P}$ cannot be useful, even in

[7] A weaker statement restricting the one-way permutation to be keyed would follow directly from Theorem 1.3 and [BPW16, SZ25], who prove that (sub-exponential) iO and one-way functions imply (structured) keyed one-way permutations.

(black-box) combination with other primitives, to build another primitive $\mathcal{Q}$. Formally, they define $\mathcal{P}$ as being black-box useless for $\mathcal{Q}$ as follows. If, for any third primitive $\mathcal{Z}$, there exists a construction of $\mathcal{Q}$ from black-box combinations of $\mathcal{P}$ and $\mathcal{Z}$, then there also exists a construction of $\mathcal{Q}$ from $\mathcal{Z}$ alone. With this definition, they show that one-way functions are black-box useless for constant-round key agreement whenever each party calls the one-way function only a constant number of times, and that one-way functions are black-box useless for approximate iO.

Black-box uselessness is significantly stronger than our composition notion. It shows that a separation can be composed with *any* other black-box separation, even if that other separation is not black-box useless itself. In contrast, we only show how to compose specific black-box separations that are "compatible" with each other. We believe this difference makes our notion, in fact, substantially more broadly applicable to known oracle separations.

Another similarity with [CFM21] is the technical idea of compiling out oracles to prove that separations compose. Here, the main conceptual difference is that [CFM21] propose to compile out oracles from *constructions*, while our analogue to compilation is only applied to *adversaries*. This makes the constraints on our simulator much weaker, and in particular the latter only needs to preserve the query complexity of the underlying adversary to be compiled.

1.2.2 Overcoming Black-Box Separations.

In some rare cases, oracle separations have been frontally overcome using non-black-box techniques. For instance, the work of Mahmoody and Pass [MP12] shows an oracle separation from non-interactive commitments to one-way functions, despite known non-black-box constructions from derandomization assumptions [BOV03]. Döttling and Garg [DG17] build identity-based encryption from the (Computational) Diffie-Hellman assumption, using garbling techniques to overcome impossibility results [BPR$^+$08, PRV12]. Another example is the construction of collision-resistance from multi-collision resistance [BT24], despite known black-box impossibility results [KNY14, MZ25]. The latter construction is non-explicit, and depends on a hypothetical adversary against collision-resistance in a non-black-box way.

In other cases, the technical barriers induced by *conditional* oracle separations have been circumvented. Often, the latter stated that no black-box construction satisfying some property P exist, and were overcome by constructions that do *not* satisfy P, e.g. [GMM07].

In our work, we explore a different avenue to circumvent black-box separations, through the combination of different individually separated primitives.

1.3 Technical Overview

We provide here more intuition about the conditions underlying Theorem 1.1, along with our applications.

Recall the setup: starting with two oracle separations $\mathcal{P}_1 \overset{BB}{\not\Rightarrow} \mathcal{Q}$ and $\mathcal{P}_2 \overset{BB}{\not\Rightarrow} \mathcal{Q}$, our goal is to understand whether black-box combinations of $\mathcal{P}_1$ and $\mathcal{P}_2$ can realize $\mathcal{Q}$.

In general, if $\mathcal{Q}$ is separated from both $\mathcal{P}_1$ and $\mathcal{P}_2$ because of "very different reasons", then $\mathcal{P}_1$ and $\mathcal{P}_2$ could in principle non-trivially complement each other and, combined, plausibly overcome the individual separations. More concretely, going back to the example of collision resistant hash functions and public-key encryption, the "joint primitive" combining PKE and CRH is separated from both PKE and CRH taken individually. However, combining PKE and CRH overcomes these individual separations. Here, CRH (resp. PKE) overcomes the deficiency of PKE (resp. CRH) in constructing CRH (resp. PKE).

Our main conceptual contribution is to identify a setting where the primitive $\mathcal{Q}$ is separated from both $\mathcal{P}_1$ and $\mathcal{P}_2$ "for a similar reason". We thus proceed to explore to what extent the shortcomings of $\mathcal{P}_1$ and $\mathcal{P}_2$ in building $\mathcal{Q}$ can be correlated.

Let us attempt to translate this intuition into the world of oracle separations. To prove an impossibility result, we wish to build an oracle $\mathcal{O}$ relative to which the two following conditions hold:

(1) there exist secure constructions of both $\mathcal{P}_1$ and $\mathcal{P}_2$ relative to $\mathcal{O}$,

(2) every (efficient) construction of $\mathcal{Q}$ relative to $\mathcal{O}$ is broken.

A natural candidate, and useful example to keep in mind, is the concatenation $\mathcal{O}$ of both oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ underlying the base black-box separations: $\mathcal{O} = (\mathcal{O}_1, \mathcal{O}_2)$.

As discussed above, in some settings, $\mathcal{P}_1$ and $\mathcal{P}_2$ can in fact be combined in a black-box manner to build $\mathcal{Q}$. Reinterpreting this as a condition on $\mathcal{O}$, this means that, in these settings, every oracle $\mathcal{O}$ fails to satisfy either condition (1) or condition (2).

Our intuition is that condition (1) failing roughly captures that "the primitives $\mathcal{P}_1$ and $\mathcal{P}_2$ get in each other's way", while condition (2) failing

reflects the possibility of combining $\mathcal{P}_1$ and $\mathcal{P}_2$ to overcome their individual shortcomings in building $\mathcal{Q}$. We view the failure of (1) more as a technical shortcoming of the oracle separations, while the failure of (2) reflects a more fundamental possibility of overcoming the black-box separations by combining primitives. Consequently, we now focus on building an oracle $\mathcal{O}$ that satisfies condition (2).

Because establishing this condition (2) is hard in general (given the counter-example above), we instead turn our attention to specific black-box separations where arguing that $\mathcal{Q}$ is broken is particularly easy. This is typically the case for separations resulting from the so-called *two-oracle technique* [Sim98, HR04]. In these oracle separations, the oracle is defined as $\mathcal{O} = (\mathsf{Const}, \mathsf{Break})$. There, Break is a breaker oracle that generically breaks the security of $\mathcal{Q}$ for any candidate construction given as input. On the other hand, the constructive oracle Const enables the existence of $\mathcal{P}$. The technical difficulty of proving these separations usually lies in arguing that Const securely instantiates $\mathcal{P}$ even given access to Break.

Our observation is that, separations with a given target primitive $\mathcal{Q}$ proven in the two-oracle technique typically feature the *same* breaker oracle Break. For instance, separations where $\mathcal{Q}$ denotes collision-resistant hash functions typically involve an oracle $\mathsf{ColFind}$ that outputs, given any (oracle-aided) input circuit, a random collision [Sim98, HHRS07, AS15, BDV17, BD19]. Importantly for us, we interpret such a phenomenon as indicating that all these impossibility results for $\mathcal{Q}$ face *the same technical barrier*, namely, that all the underlying constructive oracles are compatible with this specific oracle Break breaking $\mathcal{Q}$.

This will conceptually constitute our main setting where our composition criterion applies: we will require $\mathcal{O}_1 = (\mathsf{Const}_1, \mathsf{Break}_1)$ and $\mathcal{O}_2 = (\mathsf{Const}_2, \mathsf{Break}_2)$ to use the *same* breaker oracle: $\mathsf{Break}_1 = \mathsf{Break}_2$.

Our goal is now to show that $\mathcal{Q}$ is separated from black-box combinations of $\mathcal{P}_1$ and $\mathcal{P}_2$. We will naturally do so using a two-oracle-based separation, using as the constructive oracle $\mathsf{Const} = (\mathsf{Const}_1, \mathsf{Const}_2)$ and as the breaker oracle $\mathsf{Break} = \mathsf{Break}_1 = \mathsf{Break}_2$.

As expected, proving condition (2), namely that $\mathcal{Q}$ is broken, is now fairly direct: as long as Break breaks $\mathcal{Q}$ for any oracle-aided candidate, adding more constructive oracles will not help fixing $\mathcal{Q}$. This translates to the first condition of Theorem 1.1.

It remains to prove condition (1), namely that $\mathcal{P}_1$ and $\mathcal{P}_2$ still exist in this relativized world. By symmetry, let us focus on arguing that $\mathcal{P}_1$ exists in the presence of $\mathcal{O} = (\mathsf{Const}_1, \mathsf{Const}_2, \mathsf{Break})$. Our strategy will be to reduce this statement to the existence of $\mathcal{P}_1$ relative to $(\mathsf{Const}_1, \mathsf{Break})$. Fix any candi-

date construction for $\mathcal{P}_1$ having access to $(\mathsf{Const}_1, \mathsf{Break})$. Suppose we could turn any adversary $\mathcal{A}$ with oracle access to $\mathcal{O} = (\mathsf{Const}_1, \mathsf{Const}_2, \mathsf{Break})$, into an adversary $\mathcal{B}$ that only uses $\mathsf{Const}_1, \mathsf{Break}$. Then, the conclusion would follow from the fact that $\mathcal{P}_1$ exists relative to $(\mathsf{Const}_1, \mathsf{Break})$.

To build $\mathcal{B}$, we take inspiration from prior work on black-box separations which "compile out" various oracles. Here, it suffices to "compile out" Const_2 from $\mathcal{A}$; this is exactly what condition 2 of Theorem 1.1 ensures. Importantly, here, we only need to produce an efficient adversary in a relativized world. In particular, we only need to ensure that (i) the output of the compiled adversary is close to the output of the original one, and (ii) the efficiency *in terms of query complexity* is preserved. This constitutes the second condition for our composition.

The latter condition is a noticeable relaxation compared to prior applications of the compilation techniques in the context of black-box separations, where the goal is typically to compile out *constructions*. In these cases, the compilation needs to preserve both security properties out of the constructions, and preserve its runtime. In contrast, in our setting, we can even afford to hard-code the truth table of the constructions we wish to compile out. Still, this condition is not trivial: Break breaks $\mathcal{Q}$ relative to $\mathcal{O}$ only when given oracle access to $\mathsf{Const}_1, \mathsf{Const}_2$. So, the compilation also needs to compile out Const_2 out of Break.

We show that this condition broadly holds for separations to CRH using the collision finder $\mathsf{ColFind}$ of Simon [Sim98, AS15], or the statistical distance oracle SDO of Bitansky, Degwekar and Vaikuntanathan [BDV17, BD19]. In a nutshell, we leverage the fact that the oracles $\mathsf{ColFind}$ and SDO can accept exponentially large circuits as inputs, while the efficiency of an adversary is only counted in terms of query complexity.[8] This allows, in order to compile out a fixed oracle Const_2, to hard-code its whole truth table in (now exponential-sized) equivalent circuits, which are fed as an input to $\mathsf{ColFind}$ or SDO.

Composing Asharov-Segev-Style Separations. Last, we briefly discuss extending our framework to encompass Asharov-Segev-style separations, where we wish to rule out constructions where one "outer" primitive, typically iO, calls another "inner" primitive $\mathcal{P}$ in a somewhat non-black-box way. For instance, to capture these combinations, we allow iO to take as input $\mathcal{P}$-oracle-aided circuits, and to evaluate $\mathcal{P}$ itself.

We show that our framework extends to this setting in a conceptually

[8] This idea is a crucial component in the proofs of [BDV17, BD19].

natural way, though the resulting abstraction is arguably not as clean. One reason is that the compilation requirements from the base separations now slightly differ between the inner and the outer oracles.

The main technical drawback of our current approach is that we currently additionally require the proof of existence of the "outer primitive" (in our case, iO), which takes as input oracle-aided circuits, to not depend on how the latter "inner" oracle is instantiated. While this fortunately holds for separations from iO to CRH and hardness of SZK [BDV17, BD19], this new condition unfortunately does require a deeper technical understanding of the base separations, compared to our "plain composition" conditions. We defer removing this technical restriction to future work.

1.3.1 Additional Technical Discussions

We briefly discuss here some technical aspects of our results.

Restriction to Two-Oracle Separations. Our framework fundamentally only applies to separations proven using a two-oracle technique. This raises the intriguing possibility that separations resulting from a two-oracle technique are conceptually different than "single-oracle" separations, such as [IR89]. More concretely, *is it possible to compose separations where the brokenness of the target primitive $\mathcal{Q}$ does not relativize*?[9] Unfortunately, we currently do not see a way to extend our results to that setting. Given the trove of separations (starting with as [IR89]) that could be composed in that setting, we leave this possibility as a fascinating question for future research.

Security Parameters. The overview implicitly assumed, in our composition conditions, that adversaries always used the underlying primitives on at most one security parameter, and further, if we combine two primitives, that they would use the same security parameter for both primitives. This is because we implicitly assumed $\mathsf{Const}, \mathsf{Break}$ to be single oracles associated with a single security parameter. In the technical body, we extend our definitions to allow algorithms to call the oracles on many different security parameters, as long as these parameters are upper bounded by a polynomial (that depends on the algorithm at hand and the current security parameter), and our main results extend readily to this setting. On the technical side, this

[9] For separations broken using a generic PSPACE oracle, the brokenness usually does not relativize.

requires to keep track of such security parameters during the composition; an additional complication is that for oracles taking as input oracle-aided circuits (such as $\mathsf{ColFind}$ or SDO), we also need to restrict the security parameters used in these circuits, and argue that our composition theorem maintains this invariant.

2 Preliminaries

A function $f : \mathbb{N} \to \mathbb{R}^+$ is *negligible*, which we will denote by $f(n) \in \mathsf{negl}(n)$, if for every constant $c > 0$, there exists $n_C \in \mathbb{N}$ such that for all $n \leq n_C$, $f(n) \leq n^{-c}$.

Conventions on Oracles. Throughout this paper, we consider *oracle-aided* algorithms (resp. circuits), which have oracle access (resp. gates) to an oracle $\mathcal{O}$. We more generally consider infinite ensembles of oracles $\mathcal{O} = \{\mathcal{O}_n\}_{n\in\mathbb{N}}$, where n denotes a security parameter. More generally, we might consider (ensembles of) *distributions* of oracles $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$, which defines $\mathcal{O}_n \leftarrow \mathcal{D}_n$. For an ensemble $\mathcal{O} = \{\mathcal{O}_n\}_{n\in\mathbb{N}}$ of oracles, we abuse notation and denote oracle-aided algorithms as $\mathcal{A}^{\mathcal{O}}$, where $\mathcal{A}$ is allowed to make oracle queries $(1^n, x)$, to receive $\mathcal{O}_n(x)$ for various security parameters n.

An oracle-aided (ensemble of) algorithm is said to be PPT if it runs in probabilistic polynomial time. We say that it is *probabilistic polynomial-query* (PPQ) if it makes a polynomial number of oracle queries, but without further restrictions on running time. For all PPQ algorithms $\mathcal{A}$ we consider, we suppose that there exists a polynomial ℓ such that on input of size n, $\mathcal{A}$ only makes oracle calls to $\mathcal{O}_m$, such that $m \leq \ell(n)$. Moreover, whenever $\mathcal{A}$ internally computes oracle-aided circuits, these circuits will also only have oracle gates $\mathcal{O}_m$, such that $m \leq \ell(n)$.

We also extend these definitions to *multi-stage, stateful, interactive* algorithms, which we call (oracle-aided) *adversaries*. For any (oracle-aided, possibly multi-stage, stateful, and interactive) algorithms $\mathcal{A}, \mathcal{A}'$, we write $\mathcal{A} \equiv \mathcal{A}'$ if $\mathcal{A}$ and $\mathcal{A}'$ are functionally equivalent (for all possible oracles). We denote oracle-aided circuits $C[\cdot]$, and whenever the oracle is instantiated by an oracle Z, write $C[\mathsf{Z}]$. For two oracle-aided circuits $C_1[\mathsf{Z}], C_2[\mathsf{Z}]$ with n-bit inputs and m-bit outputs, we denote by $s = \mathsf{SD}(C_1[\mathsf{Z}], C_2[\mathsf{Z}])$ their statistical distance.

Next, we recall a broad definition of cryptographic primitives, which we specialize to the setting of efficient adversaries. We further restricts this notion to be *black-box*: we require that two functionally equivalent

adversaries with similar efficiency either both break the primitive, or neither of them do.

Definition 2.1 (Cryptographic Primitive [RTV04])**.** *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 $(\mathsf{P}, \mathcal{A}) \in R_{\mathcal{P}}$ means that the adversary $\mathcal{A}$ breaks the implementation P (according to $\mathcal{P}$). It is required that at least one function $\mathsf{P} \in F_{\mathcal{P}}$ is implementable by a PPT algorithm.*

In the plain setting, we additionally require that for any $\mathcal{A}$ running in time T such that $(\mathsf{P}, \mathcal{A}) \in R_{\mathcal{P}}$, if an adversary $\mathcal{A}'$ is functionally equivalent to $\mathcal{A}$ and runs in time $\mathsf{poly}(n)(T)$, then $(\mathsf{P}, \mathcal{A}) \in R_{\mathcal{P}}$.

In the oracle-aided setting, we require that for any oracle-aided $\mathcal{A}^{(\cdot)}$ such that $(P, \mathcal{A}) \in R_{\mathcal{P}}$, if an oracle-aided adversary $\mathcal{A}'^{(\cdot)}$ is functionally equivalent to $\mathcal{A}$ and makes polynomially many more queries than $\mathcal{A}$, then $(\mathsf{P}, \mathcal{A}) \in R_{\mathcal{P}}$.

In the rest of the paper, we will alternatively say, whenever $\mathsf{P} \in F_{\mathcal{P}}$ and $(\mathsf{P}, \mathcal{A}) \in R_{\mathcal{P}}$, that $\mathcal{A}$ *breaks* P.

We recall the notion of a joint primitive, taken verbatim from [CFM21].

Definition 2.2 (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 setting $F_{(\mathcal{P},\mathcal{Q})} := F_{\mathcal{P}} \times F_{\mathcal{Q}}$, and for each $\mathsf{G} = (\mathsf{P}, \mathsf{Q}) \in F_{\mathcal{P},\mathcal{Q}}$ and any $\mathcal{A} = (\mathcal{A}_{\mathcal{P}}, \mathcal{A}_{\mathcal{Q}})$ defining $(\mathsf{G}, \mathcal{A}) \in R_{(\mathcal{P},\mathcal{Q})}$ iff $(\mathsf{P}, \mathcal{A}_{\mathcal{P}}) \in R_{\mathcal{P}}$ or $(\mathsf{Q}, \mathcal{A}_{\mathcal{Q}}) \in R_{\mathcal{Q}}$.*

Next, we define the notion of a black-box separation from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$, which we specialize to the setting of oracle separations proven using the so-called two-oracle technique [Sim98, HR04].

Definition 2.3 (Two-Oracle-Based Black-Box Separation)**.** *Let $\mathsf{Const} = \{\mathsf{Const}_n\}_{n\in\mathbb{N}}$ be an ensemble of distributions of oracles, and Break be an oracle. We say that $(\mathsf{Const}, \mathsf{Break})$ implements a two-oracle separation from $\mathcal{P}$ to $\mathcal{Q}$ if the following two properties hold:*

Insecurity of $\mathcal{Q}$: *For any candidate PPT relativized construction $\mathsf{Q}^{(\cdot)}$ of $\mathcal{Q}$, there exists a polynomial-query adversary $\mathcal{A} = \{\mathcal{A}\}_n$ such that for all $n \in \mathbb{N}$:*

$$\Pr_{\mathsf{C}\leftarrow\$\mathsf{Const}}\left[\mathcal{A}^{\mathsf{C},\mathsf{Break}^{\mathsf{C}}} \text{ breaks } \mathsf{Q}^{\mathsf{C}}\right] \notin \mathsf{negl}(n).$$

Existence of $\mathcal{P}$: *There exists a PPT relativized construction $\mathsf{P}^{(\cdot)}$ of $\mathcal{P}$ such that for any polynomial-query adversary $\mathcal{A}$ and all $n \in \mathbb{N}$:*

$$\Pr_{\mathsf{C}\leftarrow\$\mathsf{Const}}\left[\mathcal{A}^{\mathsf{C},\mathsf{Break}^{\mathsf{C}}} \text{ breaks } \mathsf{P}^{\mathsf{C}}\right] \in \mathsf{negl}(n).$$

Note that we give the breaker oracle Break oracle access to the constructive oracle C in order to break constructions Q^{C}.

By [HR04], if there exists a two-oracle separation from $\mathcal{P}$ to $\mathcal{Q}$, then there are no fully black-box constructions of $\mathcal{Q}$ from $\mathcal{P}$, namely where both the construction calls $\mathcal{P}$ as a black-box, and the security reduction calls the adversary as a black-box [RTV04]. We will denote this by $\mathcal{P} \overset{BB}{\not\Rightarrow} \mathcal{Q}$.

Definition 2.4 (Indistinguishability Obfuscation)**.** *A probabilistic polynomial-time algorithm* iO *is an* indistinguishability obfuscator *relative to an oracle ensemble* Z *for a class* $C = \{C_n\}_{n\in\mathbb{N}}$ *of oracle-aided circuits if the following conditions are satisfied:*

Functionality *For all* $n \in \mathbb{N}$ *and all* $C \in C_n$*, it holds that*

$$\Pr\Big[C^Z \equiv \tilde{C}^Z : \tilde{C} \leftarrow \mathsf{iO}^Z(1^n, C)\Big] = 1$$

Indistinguishability *For any probabilistic polynomial-time distinguisher* $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$*,*

$$\mathcal{A}^{\mathrm{iO}}_{Z,\mathsf{iO},\mathcal{D},C}(n) = \left|\Pr\Big[\mathrm{Exp}^{\mathrm{iO;Z}}_{\mathcal{D}}(n) = 1\Big] - \frac{1}{2}\right| \leq \mathsf{negl}(n)$$

for all sufficiently large $n \in \mathbb{N}$*, where the random variable* $\mathrm{Exp}^{\mathrm{iO;Z}}_{\mathcal{D}}$ *is defined via the following experiment:*

$\mathrm{Exp}^{\mathrm{iO;Z}}_{\mathcal{D}}(n)$

$b \leftarrow\!\!\$\ \{0,1\}$
$(C_0, C_1, state) \leftarrow \mathcal{D}_1^Z(1^n)$, // *where* $C_0, C_1 \in C_n$ *and* $C_0^Z \equiv C_1^Z$
$\tilde{C} \leftarrow \mathsf{iO}(1^n, C_b; r^*)$
$b' \leftarrow \mathcal{D}_2^Z(state, \tilde{C})$
return $b = b'$

We denote by $\mathrm{Exp}^{\mathrm{iO;Z}}_{\mathrm{D}}(n; b, r^*)$ *above experiment with fixed bit* b *and fixed randomness* r^* *used by* iO*.*

3 Composing Fully Black-Box Separations

In this section, we will give our first set of conditions that enables the composition of black-box separations. We refer to the introduction and technical overview for a high-level description. The main difference to our informal Theorem 1.1 is that, in the last condition, we require the ability to compile

out any constructive oracle C from (relativizing) adversaries $\mathcal{A}^{\mathsf{C},\mathsf{Break}^{\mathsf{C}}}$. Here, the breaker oracle Break is relativized to C, and the simulator further needs to compile out the oracle calls of Break to C.

Definition 3.1 (Oracle Composition)**.** *Let* $\mathsf{Const} = \{\mathsf{Const}_n\}_{n\in\mathbb{N}}$ *be a (family of) distributions over oracles, and* Break *be an oracle. Suppose that* $(\mathsf{Const}, \mathsf{Break})$ *implements a two-oracle separation from* $\mathcal{P}$ *to* $\mathcal{Q}$ *(Definition 2.3).*

We say that this two-oracle separation $\mathcal{P} \overset{BB}{\not\Rightarrow} \mathcal{Q}$ oracle-composes relative to Break *if the following conditions hold:*

Relativization of Break: *For all (families of) oracle distributions* $\{Z_n\}_{n\in\mathbb{N}}$*, there exists a polynomial-query adversary* $\mathcal{A}$ *such that for all oracle-aided candidate construction* $\mathsf{Q}^{(\cdot)}$ *for* $\mathcal{Q}$*:*

$$\Pr_{\mathsf{Z}\leftarrow\$Z_n}\left[\mathcal{A}^{\mathsf{Z},\mathsf{Break}^{\mathsf{Z}}} \text{ breaks } \mathsf{Q}^{\mathsf{Z}}\right] \notin \mathsf{negl}(n).$$

Composition: *For all polynomial-query oracle-aided adversaries* $\mathcal{A}^{(\cdot)}$*, there exists a polynomial* p *such that for all fixed ensemble* $\mathsf{C} \leftarrow \mathsf{Const}$*, there exists a* $p(n)$*-query oracle-aided adversary* $\mathcal{B}^{(\cdot)}$ *such that for all oracle* Z*:*[10]

$$\mathcal{A}^{\mathsf{C},\mathsf{Z},\mathsf{Break}^{\mathsf{C},\mathsf{Z}}} \equiv \mathcal{B}^{Z,\mathsf{Break}^{Z}}.$$

We state our main theorem leveraging the notion of oracle-composition.

Theorem 3.2 (Composition of Separations)**.** *Suppose there exists a two-oracle separation from* $\mathcal{P}_1$ *to* $\mathcal{Q}$*, with respect to oracles (ensembles)* $(\mathsf{Const}^1, \mathsf{Break}^1)$ *(resp. from* $\mathcal{P}_2$ *to* $\mathcal{Q}$*, with respect to oracles* $(\mathsf{Const}^2), \mathsf{Break}^2))$ *(Definition 2.3).*

Suppose further that $\mathsf{Break}^1 = \mathsf{Break}^2 = \mathsf{Break}$*, and moreover, that both separations oracle-compose relative to* Break *(Definition 3.1).*

Then, there exists a two-oracle separation from the joint primitive $(\mathcal{P}_1, \mathcal{P}_2)$ *to* $\mathcal{Q}$ *with respect to oracles* $((\mathsf{Const}^1, \mathsf{Const}^2), \mathsf{Break})$*, and, moreover, this two-oracle separation oracle-composes with respect to* Break*.*

Remark 3.3 (Quantitative Tightness of the Composition)**.** *The composition in Theorem 3.2 is relatively tight in terms of both quantitative advantage and*

[10] Given our conventions on polynomial-query adversaries, this requires that the algorithm $\mathcal{B}$ only makes oracle queries to Z_m, such that $m \leq \ell(n)$ for a fixed polynomial ℓ, and all oracle-aided circuits computed by $\mathcal{B}$ only feature oracle gates Z_m, such that $m \leq \ell(n)$. See Remark 3.4 for more details.

query complexity. More precisely, if base separations quantitatively separate $\mathcal{P}_b$ with q_b-query, ε_b-advantage-security, $b \in \{0,1\}$ to (q', ε')-security of $\mathcal{Q}$, then we show that black-box combinations of $(p(q_b), \varepsilon_b)$-secure $\mathcal{P}_b$ with the same quantitative security guarantees do not suffice to build (q', ε')-secure $\mathcal{Q}$, where p is the polynomial overhead of the composition condition. In our applications, p will be the identity polynomial.

Remark 3.4 (Handling Distinct Security Parameters)**.** *Our composition theorem handles constructions and reductions that call their available oracles on possibly many different security parameters. One restriction however is that, on security parameter n, these PPQ algorithms can only make oracle queries for security parameters at most $\ell(n)$ for some polynomial n, and that the break oracle will only be called to break constructions for security parameters at most $\ell(n)$. This is captured by our conventions for oracle-aided algorithms. This formally prevents adversaries from leveraging the power of oracles for potentially exponentially large security parameters. On the technical side, we need to preserve this invariant throughout compositions – for the sake of readability, we will only mention that this invariant is preserved in appropriate footnotes.*

We first give a brief intuition for the proof of Theorem 3.2. Thanks the oracles Break being identical, and the fact that they break any instantiation of $\mathcal{Q}$ in a relativized manner, they continue to do so even in the presence of additional constructive oracles. The more subtle part is to handle the existence of each $\mathcal{P}_b$ relative to $\mathsf{Const}^{\overline{b}}, \mathsf{Break}$. We argue this by fixing the oracle $\mathsf{Const}^{\overline{b}}$, and compiling out any adversary for $\mathcal{P}_b$ into one that does not use $\mathsf{Const}^{\overline{b}}$, leveraging the composition condition of Definition 3.1. More details follow.

Proof. We prove the required properties:

Relativization of Break: any instantiation of $\mathcal{Q}$ relative to $((\mathsf{Const}^1, \mathsf{Const}^2), \mathsf{Break})$ is broken, and moreover, that such a break relativizes. This directly follows by the relativization property of Break (Definition 3.1): to argue Break relativizes with respect to Z', it suffices to take $\mathsf{Z} = (\mathsf{Z}', (\mathsf{Const}^1, \mathsf{Const}^2))$.

Existence of $\mathcal{P}_1$ (resp. $\mathcal{P}_2$): By symmetry, we focus on proving that $\mathcal{P}_1$ exists; proving that $\mathcal{P}_2$ exists proceeds identically. Combining both statements yields the existence of the joint primitive $(\mathcal{P}_1, \mathcal{P}_2)$.

We show the following slight strengthening of this statement: there exists a PPT relativized construction $\mathsf{P}_1^{(\cdot)}$ of $\mathcal{P}_1$ such that, for all *fixed* (ensemble) $\mathsf{C}_2 \leftarrow \mathsf{Const}^2$, and polynomial-query oracle-aided $\mathcal{A}^{(\cdot)}$:

$$\Pr_{\mathsf{C}_1 \leftarrow \mathsf{Const}^1}\left[\mathcal{A}^{\mathsf{C}_1,\mathsf{C}_2,\mathsf{Break}^{\mathsf{C}_1,\mathsf{C}_2}} \text{ breaks } \mathsf{P}_1^{\mathsf{C}_1}\right] \in \mathsf{negl}(n).$$

This is a strengthening over the plain existence of $\mathcal{P}_1$ along two axes. For one, it states that $\mathcal{P}_1$ exists for *any* instantiation of the oracle (ensemble) C_2. Second, it states that there exists a secure construction of $\mathcal{P}_1$ which only makes oracle calls to C_1.

Suppose the above does not hold. Then, for all candidate construction $\mathsf{P}_1^{(\cdot)}$ of $\mathcal{P}_1$, there exist a fixed oracle (ensemble) $\mathsf{C}_2 \leftarrow \mathsf{Const}^2$ and a polynomial-query oracle-aided $\mathcal{A}^{(\cdot)}$ such that:

$$\Pr_{\mathsf{C}_1 \leftarrow \mathsf{Const}^1}\left[\mathcal{A}^{\mathsf{C}_1,\mathsf{C}_2,\mathsf{Break}^{\mathsf{C}_1,\mathsf{C}_2}} \text{ breaks } \mathsf{P}_1^{\mathsf{C}_1}\right] \notin \mathsf{negl}(n).$$

By the composition property of $\mathcal{P}_2 \overset{BB}{\not\Rightarrow} \mathcal{Q}$, we obtain an adversary $\mathcal{B}$ such that

$$\Pr_{\mathsf{C}_1 \leftarrow \mathsf{Const}^1}\left[\mathcal{B}^{\mathsf{C}_1,\mathsf{Break}^{\mathsf{C}_1}} \text{ breaks } \mathsf{P}_1^{\mathsf{C}_1}\right] \notin \mathsf{negl}(n).$$

Here we use the fact that $\mathcal{A}$ and $\mathcal{B}$ are functionally equivalent, and $\mathcal{B}$ only makes polynomially more queries than $\mathcal{A}$, so that $\mathcal{B}$ is indeed a polynomial-query algorithm that breaks $\mathsf{P}_1^{\mathsf{C}_1}$.[11]

But this contradicts the existence of $\mathcal{P}_1$ relative to $(\mathsf{Const}^1, \mathsf{Break})$.

Composition. Let $\mathcal{A}$ be an arbitrary polynomial-query algorithm, and fix ensembles $\mathsf{C}_1 \leftarrow \mathsf{Const}^1$, $\mathsf{C}_2 \leftarrow \mathsf{Const}^2$, and an auxiliary oracle (ensemble) Z. By the composition property of $\mathcal{P}_1 \overset{BB}{\not\Rightarrow} \mathcal{Q}$ with auxiliary oracle $\mathsf{Z}' = (\mathsf{C}_2, \mathsf{Z})$, there exists an adversary $\mathcal{B}$ making $p(n)$ queries for a polynomial p such that:

$$\mathcal{A}^{\mathsf{C}_1,\mathsf{C}_2,\mathsf{Z},\mathsf{Break}^{\mathsf{C}_1,\mathsf{C}_2,\mathsf{Z}}} \equiv \mathcal{B}^{\mathsf{C}_2,\mathsf{Z},\mathsf{Break}^{\mathsf{C}_2,\mathsf{Z}}}$$

By now applying the composition property of $\mathcal{P}_2 \overset{BB}{\not\Rightarrow} \mathcal{Q}$ with auxiliary oracle Z, we get an adversary $\mathcal{C}$ making $q(p(n))$ queries for a polynomial q such that:

$$\mathcal{B}^{\mathsf{C}_2,Z,\mathsf{Break}^{\mathsf{C}_2,Z}} \equiv \mathcal{C}^{Z,\mathsf{Break}^{Z}},$$

which concludes the proof.[12] □

[11] Furthermore, $\mathcal{B}$ additionally satisfies the requirement of only calling Const_1 on sufficiently small security parameters, by definition of the composition property.

[12] Again, the invariant that $\mathcal{C}$ only calls oracles with small security parameters is preserved.

4 Separating SZK Hardness from OWP and CRH

As a first example of our framework, we will show for two known separations that they oracle-compose under the same break oracle. The first result is a separation of the hardness of statistical zero-knowledge (SZK) from one-way permutations (OWP) by Bitansky, Degwekar, and Vaikuntanathan [BDV17], and the second a separation of the hardness of SZK from collision-resistant hash functions (CRH) by Bitansky and Degwekar [BD19]. Both are proven using a two-oracle technique, and use as a breaker oracle one that breaks the hardness of statistical distance, a complete problem for SZK. By showing that both results are oracle-composable with respect to that breaker oracle (Definition 3.1), we show that no fully black-box construction of hard problems in SZK from black-box combinations of OWP and CRH can exist.

Theorem 4.1. *There exists no fully black-box construction of (worst-case) hard problems in* SZK *from black-box combinations of one-way permutations and collision-resistant hash functions.*

First, we introduce the breaker oracle used in both aforementioned separations. We emphasize that our breaker oracle will take circuits of any lengths as input, which will become important in the proof.

Definition 4.2 (Oracle $\mathsf{SDO}^{\mathsf{Z}}$)**.** *Let* $t = \{t_n\}_{n\in\mathbb{N}}$, $t_n : \{0,1\}^{2n} \to \left(\frac{1}{3}, \frac{2}{3}\right)$ *be a family of uniformly random functions. The oracle* SDO *has* t *hard-coded. Given* n*-bit descriptions of oracle-aided circuits* $C_0[\mathsf{Z}], C_1[\mathsf{Z}] \in \{0,1\}^n$*, let* $t^* = t_n(C_0[\mathsf{Z}], C_1[\mathsf{Z}])$ *and let* $s = \mathsf{SD}(C_0[\mathsf{Z}], C_1[\mathsf{Z}])$*, return*

$$\mathsf{SDO}^{\mathsf{Z}}(C_0[\mathsf{Z}], C_1[\mathsf{Z}]; t) = \begin{cases} 0 & \text{if } s < t^* \\ 1 & \text{if } s \geq t^* \end{cases}.$$

The proof of Theorem 4.1 will follow from the next two sections, where we will show that OWP $\overset{BB}{\not\Rightarrow}$ [hard problem in SZK] and CRH $\overset{BB}{\not\Rightarrow}$ [hard problem in SZK] are oracle-composable with respect to SDO. Theorem 4.1 then follows by Theorem 3.2, which proves that $(OWP, CRH) \overset{BB}{\not\Rightarrow}$ [hard problems in SZK].

4.1 Oracle-Composability of OWP $\overset{BB}{\not\Rightarrow}$ [Hard Problems in SZK]

We show here the following:

Lemma 4.3. *The separation OWP $\overset{BB}{\not\Rightarrow}$ [hard problems in SZK] from [BDV17] is oracle-composable with respect to SDO (Definition 3.1).*

The proof of [BDV17] uses random permutations P_n over n-bit strings as the distribution of constructive oracles, and SDO as the breaker oracle (Definition 4.2).

Proof. We show the three properties of oracle-composition.

Relativization of SDO. We show that, for all oracle distributions $\mathsf{Z} = \{Z_n\}_{n\in\mathbb{N}}$, there exists a polynomial-query adversary $\mathcal{A}^{(\cdot)}$ that decides any problem in $\mathsf{SZK}^{\mathsf{Z}}$.[13] Namely for any relativizing candidate (promise) problem P^{Z} in $\mathsf{SZK}^{\mathsf{Z}}$:[14]

$$\Pr_{\mathsf{Z}}\left[\mathcal{A}^{\mathsf{Z},\mathsf{SDO}^{\mathsf{Z}}} \text{ decides } \mathsf{P}^{\mathsf{Z}}\right] \notin \mathsf{negl}(n).$$

This follows from the facts that $\mathsf{SDO}^{\mathsf{Z}}$ decides the statistical distance problem for any circuit with oracle-access to Z, and that the statistical distance problem is complete for SZK in a relativized manner. In other words, SDO breaks SZK in a relativizing manner.

Composition. We show that for all oracle Z, for all query-efficient adversaries $\mathcal{A}$, for all fixed ensemble of permutations $\pi = \{\pi_n\}_{n\in\mathbb{N}}$, there exists an adversary $\mathcal{B}$ that perfectly simulates $\mathcal{A}$:

$$\mathcal{A}^{\pi,\mathsf{Z},\mathsf{SDO}^{\pi,\mathsf{Z}}} \equiv \mathcal{B}^{\mathsf{Z},\mathsf{SDO}^{\mathsf{Z}}}.$$

First, by definition, $\mathcal{A}$ only calls π (resp. $\mathsf{SDO}^{\pi,\mathsf{Z}}$) on security parameters $m \leq \ell(n)$ for some polynomial ℓ (resp. on circuits making oracle calls to π on security parameter $m \leq \ell(n)$).

We define $\mathcal{B}$ as follows. It has all the truth tables of $\{\pi_m\}_{m\leq\ell(n)}$ along with $\mathcal{A}$ hard-coded. It executes $\mathcal{A}$ internally, and answers the oracle calls from $\mathcal{A}$ to Z by calling itself Z, and oracle calls to π_m by looking-up its

[13] Here we define $\mathsf{SZK}^{\mathsf{Z}}$ as the class of problems that reduce to the Statistical Difference problem for Z-oracle-aided circuits. Note that the proof of completeness of Statistical Difference relativizes [SV03], and so $\mathsf{SZK}^{\mathsf{Z}}$ also corresponds to promise languages for which there exists a statistical zero-knowledge proof, where both the prover and the verifier have oracle access to Z.

[14] Recall that $\mathsf{Z} = \{Z_n\}_{n\in\mathbb{N}}$ is an ensemble of oracles. Note that P^{Z} should have a short description. In particular, for an instance of size n, P^{Z} only makes call to $\{\mathsf{Z}_m\}_{m\leq\ell(n)}$ for some polynomial ℓ.

hard-coded truth table. For oracle calls $(C_0^{\pi,\mathsf{Z}}, C_1^{\pi,\mathsf{Z}})$ to the $\mathsf{SDO}^{\pi,\mathsf{Z}}$ oracle, it defines (exponentially large) circuits $\widetilde{C}_0^Z, \widetilde{C}_1^Z$ in which we also hard-code the truth table of all the $\{\pi_m\}_{m\leq\ell(n)}$ to answer π-queries made by $C_0^{\pi,\mathsf{Z}}, C_1^{\pi,\mathsf{Z}}$, respectively; recall this is possible by assumption on $\mathcal{A}$. It then calls $\mathsf{SDO}^{\mathsf{Z}}$ on $(\widetilde{C}_0^Z, \widetilde{C}_1^Z)$ and forwards the answer. Note that this crucially relies on the fact that (1) SDO supports input queries with unbounded length, (2) we only count the *query complexity* of $\mathcal{B}$, which is equal to the number of oracle calls of $\mathcal{A}$ to $\mathsf{Z}, \mathsf{SDO}^{\pi,\mathsf{Z}}$, and (3) the security parameters on which $\mathcal{A}$ calls π (either directly, or via calls to Break) is bounded (see Remark 3.4). This perfectly simulates $\mathcal{A}$. □

4.2 Oracle-Composability of CRH $\overset{BB}{\not\Rightarrow}$ [Hard Problems in SZK]

Next, we show that the separation from CRH also oracle-composes.

Lemma 4.4. *CRH $\overset{BB}{\not\Rightarrow}$ SZK is SDO-oracle-composable. The separation CRH $\overset{BB}{\not\Rightarrow}$ [hard problems in SZK] from [BD19] is oracle-composable with respect to SDO (Definition 3.1).*

Proof. The proof of [BDV17] uses as the distribution of constructive oracle $C_n = \mathcal{F}_n$ the uniform distribution over functions $F_n : \{0,1\}^n \to \{0,1\}^{m(n)}$ where $m(n) < n$ is a polynomial. The breaker oracle is again defined as SDO (Definition 4.2).

The relativization of SDO has already proven in Lemma 4.3. So it suffices to prove the composition property.

Composition. We show that for all oracle Z, for all adversaries $\mathcal{A}$, for all fixed function ensemble $f = \{f_n\}_n$ where $f_n \in F_n$, there exists an adversary $\mathcal{B}$ that perfectly simulates $\mathcal{A}$:

$$\mathcal{A}^{f,Z,\mathsf{SDO}^{f,Z}} \equiv \mathcal{B}^{Z,\mathsf{SDO}^{Z}}.$$

The proof and definition of $\mathcal{B}$ is very similar to the one in Lemma 4.3. We repeat it almost verbatim for the sake of completeness.

First, by definition, $\mathcal{A}$ only calls f (resp. $\mathsf{SDO}^{f,\mathsf{Z}}$) on security parameters $m \leq \ell(n)$ for some polynomial ℓ (resp. on circuits making oracle calls to f on security parameter $m \leq \ell(n)$).

We define $\mathcal{B}$ as follows. It has all the truth tables of $\{f_m\}_{m\leq\ell(n)}$ along with $\mathcal{A}$ hard-coded. It executes $\mathcal{A}$ internally, and answers the oracle calls from $\mathcal{A}$ to Z by calling itself Z, and oracle calls to f_m by looking-up its

hard-coded truth table. For oracle calls $(C_0^{f,\mathsf{Z}}, C_1^{f,\mathsf{Z}})$ to the $\mathsf{SDO}^{f,\mathsf{Z}}$ oracle, it defines (exponentially large) circuits $\widetilde{C}_0^Z, \widetilde{C}_1^Z$ in which we also hard-code the truth table of $\{f_m\}_{m\leq\ell(n)}$ to answer f_m-queries made by $C_0^{f,\mathsf{Z}}, C_1^{f,\mathsf{Z}}$, respectively; recall this is possible by assumption on $\mathcal{A}$. It then calls $\mathsf{SDO}^{\mathsf{Z}}$ on $(\widetilde{C}_0^Z, \widetilde{C}_1^Z)$ and forwards the answer. This again crucially relies on the fact that (1) SDO supports input queries with unbounded length, (2) we only count the *query complexity* of $\mathcal{B}$, which is equal to the number of oracle calls of $\mathcal{A}$ to $\mathsf{Z}, \mathsf{SDO}^{\pi,\mathsf{Z}}$, and (3) the security parameters on which $\mathcal{A}$ calls f (either directly, or via calls to Break) is bounded (see Remark 3.4). This perfectly simulates $\mathcal{A}$. □

5 Asharov-Segev Style Composition

In Sections 3 and 4, we studied the composition of separations to rule out *fully black-box* compositions of different primitives. In this section, we develop techniques to rule out a larger set of partially black-box compositions.

Let us give a roadmap for this section, as we introduce quite a few definitions and notations. In Definitions 5.1,5.2, we set up the definitions and languages to talk about oracle separations in the Asharov-Segev sense, where one primitive can have (limited) non-black-box access to another. In Definition 5.3, we introduce our main conditions, which define when an Asharov-Segev-style composition can be composed with a standard two-oracle-style one. In Theorem 5.4, we formalize the composition of Asharov-Segev-style separations.

Our primary motivating example is to study composition where one of the base primitives is iO: its strengths emerge only given the ability to obfuscate other cryptographic primitives.

Therefore, in the style of Asharov and Segev [AS15], we introduce a partially non-black-box version of Definition 3.1. The goal is to allow a hierarchical, non-black-box access *across* the base primitives. Namely, given two primitives A and B, we allow B black-box access to A, but not the other way around. For concreteness, a useful running example would be to think of B as iO, and A as any additional cryptographic assumption. Therefore, we focus on proving statements relative to the following oracles:

$$A, B^A, \mathsf{Break}^{A,B^A}$$

Note that we still build on separations relying on the two-oracle technique, and thus our reduction is still fully black-box. Due to the hierarchical def-

initions of oracle access, we now require two slightly distinct notions of composition to hold.

In the following, we restrict our access from B to A to be through *oracle-aided circuits*, following the ideas of Asharov and Segev [AS15]: we will limit the non-blackboxness to having B taking as input oracle-aided circuits with A-gates, and B having standard black-box access to A.

Looking ahead, we will need a notion of composability for B, which allows to take oracle-aided circuits with internal access to A as inputs — we will call this an "Asharov-Segev-Style composability", and a strengthened notion of composability for A, due to the ability of B to call A — we will call this an augmented composability. We refer to Definition 5.3 for more details.

We first set up some notation.

Definition 5.1 (Asharov-Segev-type Oracle)**.** *We say that an (oracle-aided ensemble of) oracle $\mathsf{C} = \{\mathsf{C}_n\}_n$ is an Asharov-Segev-type oracle if it takes as (potentially partial) input oracle-aided circuits. Moreover, instantiating the latter, inner oracle with some oracle $\mathsf{Z} = \{\mathsf{Z}_n\}_n$, we will denote the former, outer oracle by $\mathsf{C}[\mathsf{Z}]$.*

Moreover, we will assume that for all PPQ algorithm $\mathcal{A}$ with oracle access to $\mathsf{C}[\mathsf{Z}]$, there exists a polynomial ℓ such that, on security parameter n, (1) all queries from $\mathcal{A}$ to $\mathsf{C}_m[\mathsf{Z}]$ are over security parameter $m \leq \ell(n)$, and (2) all queries from $\mathcal{A}$ to $\mathsf{C}[\mathsf{Z}]$ only feature circuits with oracle-gates $\{\mathsf{Z}_m\}_{m \leq \ell(n)}$.

First, we strengthen the notion of two-oracle-style separation to accommodate non-black-box access of the primitives.

Definition 5.2 (Asharov-Segev-type Black-Box Separation)**.** *Let $\mathsf{Const} = \{\mathsf{Const}_n\}_{n\in\mathbb{N}}$ be an ensemble of distributions of oracles, and Break be a relativizing oracle. We say that $(\mathsf{Const}, \mathsf{Break})$ implements an Asharov-Segev-type separation from $\mathcal{P}$ to $\mathcal{Q}$ if oracles $\mathsf{C} \in \mathsf{Const}$ is an Asharov-Segev-type oracle (Definition 5.1), and, moreover, the following two properties hold:*

Insecurity of $\mathcal{Q}$: *For any candidate PPT relativized construction $\mathsf{Q}^{(\cdot)}$ of $\mathcal{Q}$, there exists a polynomial-query adversary $\mathcal{A} = \{\mathcal{A}\}_n$ such that for all $n \in \mathbb{N}$:*

$$\Pr_{\mathsf{C}\leftarrow\$\mathsf{Const}}\left[\mathcal{A}^{\mathsf{C},\mathsf{Break}^{\mathsf{C}}} \text{ breaks } \mathsf{Q}^{\mathsf{C}}\right] \notin \mathsf{negl}(n).$$

Relativized Existence of $\mathcal{P}$: *There exists a PPT relativized construction $\mathsf{P}^{(\cdot)}$ of $\mathcal{P}$ such that for any polynomial-query adversary $\mathcal{A}$ and all $n \in$*

$\mathbb{N}$, *and* any *fixed oracle* Z*:*

$$\Pr_{\mathsf{C}[\cdot]\leftarrow\$\mathsf{Const}}\left[\mathcal{A}^{\mathsf{C}[\mathsf{Z}],\mathsf{Break}^{\mathsf{C}[\mathsf{Z}]}} \text{ breaks } \mathsf{P}^{\mathsf{C}}\right] \in \mathsf{negl}(n),$$

where we recall that $\mathsf{C}[\mathsf{Z}]$ *denotes an oracle taking as input* Z*-oracle-aided circuits.*

The differences with Definition 2.3 are as follows: first, we syntactically require constructions C to take as (partial) input oracle-aided circuits (Definition 5.1). Second, we require that the existence of $\mathcal{P}$ does not depend on the oracle Z used to instantiate the aforementioned oracle-aided circuits. This is a fairly technical condition that we use to prove the existence of iO in relativized worlds. Fortunately for us, standard separations from iO explicitly prove that this condition holds for all fixed Z [BDV17, BD19].

Next, we state our augmented composition conditions.

Definition 5.3 (Asharov-Segev-type Oracle Composition)**.** *Let* $\mathsf{Const} = \{\mathsf{Const}_n\}_{n\in\mathbb{N}}$ *be a (family of) distributions over oracles, and* Break *be an oracle. Suppose that* $(\mathsf{Const}, \mathsf{Break})$ *implements a separation from* $\mathcal{P}$ *to* $\mathcal{Q}$*.*

Consider the following conditions:

Relativization of Break: *For all (families of) oracle distributions* $\mathsf{Z} = \{Z_n\}_{n\in\mathbb{N}}$*, there exists a polynomial-query adversary* $\mathcal{A}$ *such that for all oracle-aided candidate construction* $\mathsf{Q}^{(\cdot)}$ *for* $\mathcal{Q}$*:*

$$\Pr_{\mathsf{Z}}\left[\mathcal{A}^{\mathsf{Z},\mathsf{Break}^{\mathsf{Z}}} \text{ breaks } \mathsf{Q}^{\mathsf{Z}}\right] \notin \mathsf{negl}(n).$$

AS-type Composition (for Asharov-Segev-type Separations): *For all polynomial-query oracle-aided adversaries* $\mathcal{A}^{(\cdot)}$*, there exists a polynomial* p *such that for all fixed Asharov-Segev-style oracle* $\mathsf{C} \in \mathsf{Const}$ *(Definition 5.1), there exists a* $p(n)$*-query oracle-aided adversary* $\mathcal{B}^{(\cdot)}$ *such that for all oracles* Z*:*

$$\mathcal{A}^{\mathsf{Z},\mathsf{C}[\mathsf{Z}],\mathsf{Break}^{\mathsf{Z},\mathsf{C}[\mathsf{Z}]}} \equiv \mathcal{B}^{\mathsf{Z},\mathsf{Break}^{\mathsf{Z}}}.$$

Augmented Composition (for Two-Oracle Separations): *For all polynomial-query oracle-aided adversaries* $\mathcal{A}^{(\cdot)}$*, there exists a polynomial* p *such that for all fixed* $\mathsf{C} \in \mathsf{Const}$*, there exists a* $p(n)$*-query oracle-aided adversary* $\mathcal{B}^{(\cdot)}$ *such that for all Asharov-Segev-style oracle* Z*:*

$$\mathcal{A}^{\mathsf{C},\mathsf{Z}[\mathsf{C}],\mathsf{Break}^{\mathsf{C},\mathsf{Z}[\mathsf{C}]}} \equiv \mathcal{B}^{\mathsf{Z}[\mathsf{C}],\mathsf{Break}^{\mathsf{Z}[\mathsf{C}]}}.$$

We say that an Asharov-Segev-style separation $\mathcal{P} \overset{BB}{\not\Rightarrow} \mathcal{Q}$ AS-oracle-composes relative to Break *(we also write* $\mathcal{P} \overset{BB_{\mathsf{AS}}}{\not\Rightarrow} \mathcal{Q}$*) if it satisfies the relativization of* Break *and the AS-type composition properties.*

We say that a two-oracle separation $\mathcal{P} \overset{BB}{\not\Rightarrow} \mathcal{Q}$ augmented-oracle-composes relative to Break *(we also write* $\mathcal{P} \overset{BB_{\mathsf{Aug}}}{\not\Rightarrow} \mathcal{Q}$*) if it satisfies the relativization of* Break *and the augmented composition properties.*

Let us comment a bit on these new definitions. First, the AS-type composition property requires the ability to simulate any C oracle call using Z oracle calls, even though C takes as input circuits with Z-oracle gates. This creates an additional difficulty compared to the standard composition property of Definition 3.1.

Second, the augmented composition only requires to simulate out the explicit C oracle calls from $\mathcal{A}$ and Break, not those made internally by Z.

We are now ready to state the main theorem of this section:

Theorem 5.4. *Suppose there exists a two-oracle separation from* $\mathcal{P}_1$ *to* $\mathcal{Q}$*, with respect to oracles* $(\mathsf{Const}^1, \mathsf{Break}^1)$ *(Definition 5.2), and that there exists an Asharov-Segev-style separation from* $\mathcal{P}_2$ *to* $\mathcal{Q}$*, with respect to oracles* $(\mathsf{Const}^2, \mathsf{Break}^2)$ *(Definition 2.3).*

Suppose further that $\mathsf{Break}^1 = \mathsf{Break}^2 = \mathsf{Break}$*, and moreover, that* $\mathcal{P}_1 \overset{BB_{\mathsf{Aug}}}{\not\Rightarrow} \mathcal{Q}$ *relative to* Break*, and that* $\mathcal{P}_2 \overset{BB_{\mathsf{AS}}}{\not\Rightarrow} \mathcal{Q}$ *with respect to* Break *(Definition 5.3).*

Then, there exists an Asharov-Segev-style separation from the joint primitive $(\mathcal{P}_1, \mathcal{P}_2[\mathcal{P}_1])$ *to* $\mathcal{Q}$ *with respect to oracles* $((\mathsf{Const}^1, \mathsf{Const}^2), \mathsf{Break})$*, and, moreover, this separation AS-oracle-composes with respect to* Break*.*

Proof. First, note that the relativizing property Break, is independent of the constructive oracles and therefore, the analogous proof from Theorem 3.2 extends directly.

Because $\mathcal{P}_1$ and $\mathcal{P}_2$ do not have symmetric roles anymore, we prove their existence individually. The existence of $\mathcal{P}_1$ is proven very similarly to Theorem 3.2.

Existence of $\mathcal{P}_1$. We show the following slight strengthening of this statement: There exists a PPT relativized construction $\mathsf{P}_1^{(\cdot)}$ of $\mathcal{P}_1$ such that, for all *fixed* Asharov-Segev-style oracle $\mathsf{C}_2 \leftarrow \mathsf{Const}^2$, and oracle-aided $\mathcal{A}^{(\cdot)}$:

$$\Pr_{\mathsf{C}_1 \leftarrow \mathsf{Const}^1}\left[\mathcal{A}^{\mathsf{C}_1, \mathsf{C}_2[\mathsf{C}_1]}, \mathsf{Break}^{\mathsf{C}_1, \mathsf{C}_2[\mathsf{C}_1]} \text{ breaks } \mathsf{P}^{\mathsf{C}_1}\right] \in \mathsf{negl}(n),$$

where we recall that $\mathsf{C}_2[\mathsf{C}_1]$ denotes C_2 having the ability to call as input C_1-oracle-aided circuits.

Suppose the above does not hold. Then, for all candidate construction $\mathsf{P}_1^{(\cdot)}$ of $\mathcal{P}_1$, there exists a fixed oracle ensemble $\mathsf{C}_2[\mathsf{C}_1] \leftarrow \mathsf{Const}^2$ and an oracle-aided $\mathcal{A}^{(\cdot)}$ such that:

$$\Pr_{\mathsf{C}_1 \leftarrow \mathsf{Const}^1}\left[\mathcal{A}^{\mathsf{C}_1,\mathsf{C}_2[\mathsf{C}_1]}, \mathsf{Break}^{\mathsf{C}_1,\mathsf{C}_2[\mathsf{C}_1]} \text{ breaks } \mathsf{P}^{\mathsf{C}_1}\right] \notin \mathsf{negl}(n).$$

By the Asharov-Segev-style composition property of $\mathcal{P}_2 \overset{BB_{\mathsf{AS}}}{\not\Rightarrow} \mathcal{Q}$ (Definition 5.3), we obtain an adversary $\mathcal{B}$ such that

$$\Pr_{\mathsf{C}_1 \leftarrow \mathsf{Const}^1}\left[\mathcal{B}^{\mathsf{C}_1,\mathsf{Break}^{\mathsf{C}_1}} \text{ breaks } \mathsf{P}_1^{C_1}\right] \notin \mathsf{negl}(n).$$

Here we use the fact that $\mathcal{A}$ and $\mathcal{B}$ are functionally equivalent, and $\mathcal{B}$ only makes polynomially more queries than $\mathcal{A}$, then $\mathcal{B}$ still breaks $\mathsf{P}_1^{\mathsf{C}_1}$.[15]

But this contradicts the existence of $\mathcal{P}_1$ relative to $(\mathsf{Const}^1, \mathsf{Break})$.

Existence of $\mathcal{P}_2$. We show the following slight strengthening of this statement: There exists a PPT relativized construction $\mathsf{P}_2^{(\cdot)}$ of $\mathcal{P}_2$ such that, for all *fixed* $\mathsf{C}_1 \leftarrow \mathsf{Const}^1$, and oracle-aided $\mathcal{A}^{(\cdot)}$:

$$\Pr_{\mathsf{C}_2 \leftarrow \mathsf{Const}^2}\left[\mathcal{A}^{\mathsf{C}_1,\mathsf{C}_2[\mathsf{C}_1],\mathsf{Break}^{\mathsf{C}_1,\mathsf{C}_2[\mathsf{C}_1]}} \text{ breaks } \mathsf{P}^{\mathsf{C}_2}\right] \in \mathsf{negl}(n).$$

Note that here the oracle C_2 accessed by P_2 does not call any circuits with C_1 gates.

Suppose the above does not hold. Then, for all candidate construction $\mathsf{P}_2^{(\cdot)}$ of $\mathcal{P}_2$, there exists a fixed oracle ensemble $\mathsf{C}_1 \leftarrow \mathsf{Const}^2$ and an oracle-aided $\mathcal{A}^{(\cdot)}$ such that:

$$\Pr_{\mathsf{C}_2 \leftarrow \mathsf{Const}^2}\left[\mathcal{A}^{\mathsf{C}_1,\mathsf{C}_2[\mathsf{C}_1],\mathsf{Break}^{\mathsf{C}_1,\mathsf{C}_2[\mathsf{C}_1]}} \text{ breaks } \mathsf{P}^{\mathsf{C}_2}\right] \notin \mathsf{negl}(n).$$

By the augmented composition property of $\mathcal{P}_1 \overset{BB_{\mathsf{AS}}}{\not\Rightarrow} \mathcal{Q}$ (Definition 5.3), we obtain an adversary $\mathcal{B}$ such that

$$\Pr_{\mathsf{C}_2 \leftarrow \mathsf{Const}^2}\left[\mathcal{B}^{\mathsf{C}_2[\mathsf{C}_1],\mathsf{Break}^{\mathsf{C}_2[\mathsf{C}_1]}} \text{ breaks } \mathsf{P}^{\mathsf{C}_2}\right] \notin \mathsf{negl}(n).$$

[15] Furthermore, $\mathcal{B}$ additionally satisfies the requirement of only calling C_1 on sufficiently small security parameters, by definition of the composition property.

Here we use the fact that $\mathcal{A}$ and $\mathcal{B}$ are functionally equivalent, and $\mathcal{B}$ only makes polynomially more queries than $\mathcal{A}$, then $\mathcal{B}$ still breaks $\mathsf{P}_2^{\mathsf{C}_2}$.[16]

But this contradicts the relativized existence $\mathcal{P}_2$ relative to the (Const_2, Break) and the inner oracle $\mathsf{Z} = \mathsf{C}_1$ (Definition 5.2).

Composition. Let $\mathcal{A}$ be an arbitrary polynomial-query algorithm, and $\mathsf{C}_1 \in \mathsf{Const}^1$, $\mathsf{C}_2 \in \mathsf{Const}^2$, and an auxiliary oracle Z. By the augmented composition property of $\mathcal{P}_1 \overset{BB}{\not\Rightarrow} \mathcal{Q}$ with auxiliary oracle $\mathsf{Z}' = (\mathsf{C}_2[\mathsf{C}_1], \mathsf{Z})$, there exists an adversary $\mathcal{B}$ making $p(n)$ queries for a polynomial p such that:

$$\mathcal{A}^{\mathsf{C}_1,\mathsf{C}_2[\mathsf{C}_1],\mathsf{Z},\mathsf{Break}^{\mathsf{C}_1,\mathsf{C}_2[\mathsf{C}_1],\mathsf{Z}}} \equiv \mathcal{B}^{\mathsf{C}_2[\mathsf{C}_1],\mathsf{Z},\mathsf{Break}^{\mathsf{C}_2[\mathsf{C}_1],\mathsf{Z}}}$$

By now applying the Asharov-Segev-style composition property of $\mathcal{P}_2 \overset{BB}{\not\Rightarrow} \mathcal{Q}$ with auxiliary oracle Z, we get an adversary $\mathcal{C}$ making $q(p(n))$ queries for a polynomial q such that:

$$\mathcal{B}^{\mathsf{C}_2[\mathsf{C}_1],Z,\mathsf{Break}^{\mathsf{C}_2[\mathsf{C}_1],Z}} \equiv \mathcal{C}^{Z,\mathsf{Break}^{Z}},$$

which concludes the proof.[17]

□

6 Separating SZK Hardness from iO & CRH

In this section, we apply our framework from the last section to show that there exists no black-box construction of hard problems in SZK from black-box combinations of collision-resistant hash functions (CRH) and indistinguishability obfuscation (iO), where the iO is allowed to obfuscate circuits with CRH gates. Previously, it was known that combining one-way permutations with iO is neither sufficient to build collision-resistant hash functions [AS15], nor hard problems in SZK [BDV17] in a black-box way. To obtain our results, we compose separations due to Bitansky, Degwekar and Vaikuntanathan [BDV17] and Bitansky and Degwekar [BD19].

[16] Furthermore, $\mathcal{B}$ additionally satisfies the requirement of only calling C_2 on sufficiently small security parameters, and only calls C_2 on C_1-oracle-aided circuits with sufficiently small security parameters, by definition of the composition property and our conventions on PPQ algorithms.

[17] Again, both $\mathcal{B}$ and $\mathcal{C}$ additionally satisfy the requirement of only calling their oracles on sufficiently small security parameters, and only call C_2 on C_1-oracle-aided circuits on sufficiently small security parameters, by definition of the composition property and our conventions on PPQ algorithms.

Theorem 6.1. *There exists no fully black-box construction of (worst-case) hard problems in* SZK *from black-box combinations of collision-resistant hash functions and indistinguishability obfuscation. More precisely:*

$$(\mathsf{CRH}, \mathsf{iO}[\mathsf{CRH}]) \overset{BB_{\mathsf{AS}}}{\not\Rightarrow} \textit{ [hard problems in } \mathsf{SZK}\textit{]},$$

where $\mathsf{iO}[\mathsf{CRH}]$ *denotes an indistinguishability obfuscator that can obfuscate circuits with CRH-gates, and where* $\overset{BB_{\mathsf{AS}}}{\not\Rightarrow}$ *is defined in Definition 5.2.*

Looking ahead, in Lemma 4.3 and Lemma 6.3, we prove that the separations from [BDV17] and [BD19] satisfy our composition conditions of Definition 5.3. Combined with Theorem 5.4, we obtain Theorem 6.1.

First, we show that the separation from CRH to hard problems in SZK in [BD19] oracle-composes. The separation uses the same breaker oracle SDO as in Section 4; see Definition 4.2.

Lemma 6.2. *The separation* $\mathsf{CRH} \overset{BB_{\mathsf{Aug}}}{\not\Rightarrow}$ *[hard problems in* SZK*] from [BD19] is augmented-oracle-composable with respect to* SDO *(Definition 5.3).*

Proof. We have already showed in Lemma 4.4 that the breaker SDO relativizes and that the separation satisfies plain composition. It only remains to show that it satisfies the augmented composition property.

The simulator for the augmented composition property is defined nearly identically to the proof of Lemma 4.4, except that it relays queries to $\mathsf{Z}[\mathsf{C}]$ (as opposed to Z), and calls the oracle $\mathsf{SDO}^{\mathsf{Z}[\mathsf{C}]}$ (as opposed to $\mathsf{SDO}^{\mathsf{Z}}$). The rest of the proof is identical.[18]

□

Next, we will show that the separation from iO (and one-way permutations) to hard problems in SZK, due to [BDV17] is AS-oracle-composable with respect to SDO (Definition 5.3). In their oracle separation, the constructive oracle implementing iO is defined as follows. Let $\mathcal{O} = \{\mathcal{O}_n : \{0,1\}^{2n} \to \{0,1\}^{5n}\}$ be the family of injective functions mapping from bit-strings of length $2n$ to length $5n$. To obfuscate an oracle-aided circuit $C[\cdot]$, where $[\cdot]$ its internal oracle, the obfuscator samples a random $r \in \{0,1\}^n$ and compute $\tilde{C} = \mathcal{O}(C[\cdot], r)$. To evaluate an obfuscated circuit on an input x, they further define an oracle $\mathsf{Eval}^{\mathcal{O},[\cdot]}$ as follows: on input $(\tilde{C}, x)$, invert

[18] In particular the invariants that the simulator only calls its oracles on small security parameters, and that its oracle-aided-circuits used as inputs to oracle calls also only use small security parameters, are preserved.

$(C[\cdot], r) = \mathcal{O}^{-1}(\tilde{C})$. Output $\perp$ if this fails. Otherwise, interpret $C[\cdot]$ as an oracle-aided circuit, and output $C[\cdot](x)$.

We now show our composability properties.

Lemma 6.3. *The separation $iO \overset{BB_{\mathsf{AS}}}{\not\Rightarrow}$ [hard problems in SZK] AS-oracle-composes with respect to SDO (Definition 5.3).*

Proof. We define the constructive oracle $\mathsf{Const}_n = (\mathcal{O}_n, \mathsf{Eval}^{\mathcal{O},[\cdot]})$. We have already shown that SDO relativizes in Lemma 4.4. Therefore, we are left with showing the relativized existence of iO (Definition 5.2) and composition properties (Definition 5.3).

Relativized Existence of iO. Fortunately for us, the proof of existence of iO relative to $\mathsf{Const}, \mathsf{SDO}$ in [BDV17] establishes this stronger fact. This is because their proof of existence for iO using oracle-aided input circuits holds for *any fixed* oracle f as their inner oracle, and does not depend on f itself.

AS-type Composition. Our goal is to show that for all polynomial-query oracle-aided adversaries $\mathcal{A}^{(\cdot)}$, there exists a polynomial p such that for all n and all fixed oracle $(\mathsf{O}, \mathsf{Eval}^{\mathsf{O},[\cdot]}) \in \mathsf{Const}_n$ (Definition 5.1), there exists a $p(n)$-query oracle-aided adversary $\mathcal{B}^{(\cdot)}$ such that for all oracles Z:

$$\mathcal{A}^{\mathsf{Z},\mathcal{O}[\mathsf{Z}],\mathsf{Eval}^{\mathcal{O},[\mathsf{Z}]},\mathsf{SDO}^{\mathsf{Z},\mathcal{O}[\mathsf{Z}],\mathsf{Eval}^{\mathcal{O},[\mathsf{Z}]}}} \equiv \mathcal{B}^{\mathsf{Z},\mathsf{Break}^{\mathsf{Z}}}.$$

Let us parse this requirement. We recall that the notation $[\mathsf{Z}]$ means that the oracle expects Z-oracle-aided circuits as (partial) inputs. So, our goal is to compile out, for any fixed "inner" oracle Z, all the explicit calls of both $\mathcal{A}$ and SDO to the obfuscation oracle.

Recall that $\ell(n)$ denotes a bound on the security parameters used by $\mathcal{A}$: this includes $\mathsf{Z}, \mathcal{O}, \mathsf{Eval}$ oracles, as well as oracle-aided-circuits that $\mathcal{A}$ feeds as input to $\mathcal{O}, \mathsf{Eval}$, and SDO.

We define $\mathcal{B}$ as follows. It has the truth table of $\{\mathsf{O}_m\}_{m \le \ell(n)}$ hard-coded. It executes $\mathcal{A}$ internally, and answers to oracle calls from $\mathcal{A}$ as follows:

- for Z-calls, $\mathcal{B}$ answers using its Z oracle;
- for O-calls, it uses the hard-coded truth tables of $\{\mathsf{O}_m\}_{m \le \ell(n)}$;
- for Eval-calls, it uses the truth table of O_m to (attempt to) invert the input, and computes the result $C[\mathsf{Z}](x)$ using its Z oracle. Crucially,

$C[\mathsf{Z}]$ has *small size* as an oracle-aided circuit, of size at most $\ell(n)$. So computing $C[\mathsf{Z}](x)$ can be done using at most $\ell(n)$ calls to Z.[19] Note that this is the same query complexity with respect to Z as the original Eval algorithm, and so this part of the simulation preserves the overall query complexity with respect to Z.

- for Break-calls, it uses the truth table of $\{\mathsf{O}_m\}_{m\leq\ell(n)}$ to create circuits equivalent to the input circuits, using only Z-gates (note that these circuits might be exponentially larger), and runs $\mathsf{SDO}^{\mathsf{Z}}$.

Again, we crucially rely on the fact that SDO supports input queries with unbounded length, and that we only count the *query complexity* of $\mathcal{B}$. This perfectly simulates $\mathcal{A}$.[20] □

Theorem 6.1 follows from combining Lemma 4.3, Lemma 6.3, and Theorem 5.4.

We can readily extend our result to further rule out the usefulness of one-way permutations:

Corollary 6.4. *There exists no fully black-box construction of (worst-case) hard problems in* SZK *from black-box combinations of collision-resistant hash functions, one-way permutations, and indistinguishability obfuscation. More precisely:*

$$(\mathsf{CRH}, \mathsf{OWP}, \mathsf{iO}[\mathsf{CRH}, \mathsf{OWP}]) \overset{BB_{\mathsf{AS}}}{\not\Rightarrow} \textit{[hard problems in } \mathsf{SZK}\textit{]},$$

where $\mathsf{iO}[\mathsf{CRH}, \mathsf{OWP}]$ *denotes an indistinguishability obfuscator that can obfuscate circuits with both CRH and OWP gates.*

Proof. Given Theorem 5.4, and Theorem 6.1, it suffices to argue that the separation $\mathsf{OWP} \overset{BB_{\mathsf{Aug}}}{\not\Rightarrow}$[hard problems in SZK] augmented-oracle-composes (Definition 5.3). Given that SDO relativizes, it suffices to show that the separation satisfies the augmented-oracle-composition property. This follows almost identically to the proof of Lemma 4.3, relaying queries to $\mathsf{Z}[\mathsf{C}]$ (instead of Z), and calling the oracle $\mathsf{SDO}^{\mathsf{Z}[\mathsf{C}]}$ (instead of $\mathsf{SDO}^{\mathsf{Z}}$). □

[19] Recall that O_n and Eval_n process circuits with description size n. Therefore the number of Z-gates involved in one execution is also upper bounded by n. Because of our conventions, $\mathcal{A}$ only calls O and Eval on security parameter at most $m = \ell(n)$, and the conclusion follows.

[20] Again, the invariants that $\mathcal{B}$ only calls its oracles on small security parameters $m \leq \ell(n)$, and that its oracle-aided-circuits used as inputs to oracle calls also only use small security parameters $m \leq \ell(n)$, are preserved.

Acknowledgments

We thank the anonymous reviewers for valuable comments.

References

[AS15] 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.

[BBF13] 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 Berlin Heidelberg, Germany.

[BD19] Nir Bitansky and Akshay Degwekar. On the complexity of collision resistant hash functions: New and old black-box separations. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part I*, volume 11891 of *Lecture Notes in Computer Science*, pages 422–450, Nuremberg, Germany, December 1–5, 2019. Springer, Cham, Switzerland.

[BDV17] Nir Bitansky, Akshay Degwekar, and Vinod Vaikuntanathan. Structure vs. hardness through the obfuscation lens. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 696–723, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland.

[BKSY11] Zvika Brakerski, Jonathan Katz, Gil Segev, and Arkady Yerukhimovich. Limits on the power of zero-knowledge proofs in cryptographic constructions. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 559–578, Providence, RI, USA, March 28–30, 2011. Springer Berlin Heidelberg, Germany.

[BMM24] Amos Beimel, Tal Malkin, and Noam Mazor. Structural lower bounds on black-box constructions of pseudorandom functions. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part V*, volume 14924 of *Lecture Notes in Computer Science*, pages 459–488, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[BOV03] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 299–315, Santa Barbara, CA, USA, August 17–21, 2003. Springer Berlin Heidelberg, Germany.

[BPR+08] Dan Boneh, Periklis A. Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *49th Annual Symposium on Foundations of Computer Science*, pages 283–292, Philadelphia, PA, USA, October 25–28, 2008. IEEE Computer Society Press.

[BPW16] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos - trapdoor permutations from 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 474–502, Tel Aviv, Israel, January 10–13, 2016. Springer Berlin Heidelberg, Germany.

[BT24] Jan Buzek and Stefano Tessaro. Collision resistance from multi-collision resistance for all constant parameters. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part V*, volume 14924 of *Lecture Notes in Computer Science*, pages 429–458, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[CFM21] Geoffroy Couteau, Pooya Farshim, and Mohammad Mahmoody. Black-box uselessness: Composing separations in cryptography. In James R. Lee, editor, *ITCS 2021: 12th Innovations in Theoretical Computer Science Conference*, volume 185, pages 47:1–47:20, Virtual Conference, January 6–8, 2021. Leibniz International Proceedings in Informatics (LIPIcs).

[CKP15] 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 Berlin Heidelberg, Germany.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. 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 468–497, Warsaw, Poland, March 23–25, 2015. Springer Berlin Heidelberg, Germany.

[DG17] Nico Döttling and Sanjam Garg. Identity-based encryption from the Diffie-Hellman assumption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 537–569, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland.

[DI06] Bella Dubrov and Yuval Ishai. On the randomness complexity of efficient sampling. In Jon M. Kleinberg, editor, *38th Annual ACM Symposium on Theory of Computing*, pages 711–720, Seattle, WA, USA, May 21–23, 2006. ACM Press.

[GGKT05] 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.

[GHMM18] 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, Cham, Switzerland.

[GMM07] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and CCA security for public key encryption. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography*

Conference, volume 4392 of *Lecture Notes in Computer Science*, pages 434–455, Amsterdam, The Netherlands, February 21–24, 2007. Springer Berlin Heidelberg, Germany.

[GMM17] 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, Cham, Switzerland.

[GT00] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science*, pages 305–313, Redondo Beach, CA, USA, November 12–14, 2000. IEEE Computer Society Press.

[HHRS07] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *48th Annual Symposium on Foundations of Computer Science*, pages 669–679, Providence, RI, USA, October 20–23, 2007. IEEE Computer Society Press.

[HL18] 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.

[HR04] Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 92–105, Santa Barbara, CA, USA, August 15–19, 2004. Springer Berlin Heidelberg, Germany.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity*

Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995, pages 134–147. IEEE Computer Society, 1995.

[IR89] 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.

[KNY14] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for NP. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 254–273, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer Berlin Heidelberg, Germany.

[MP12] Mohammad Mahmoody and Rafael Pass. The curious case of non-interactive commitments - on the power of black-box vs. non-black-box use of primitives. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 701–718, Santa Barbara, CA, USA, August 19–23, 2012. Springer Berlin Heidelberg, Germany.

[MZ25] Xinyu Mao and Jiapeng Zhang. Black-box separation between multi-collision resistance and collision resistance. Cryptology ePrint Archive, Report 2025/2049, 2025.

[PRV12] Periklis A. Papakonstantinou, Charles W. Rackoff, and Yevgeniy Vahlis. How powerful are the DDH hard groups? Cryptology ePrint Archive, Report 2012/653, 2012.

[RTV04] 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 Berlin Heidelberg, Germany.

[Sim98] 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 Berlin Heidelberg, Germany.

[SV03] Amit Sahai and Salil Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, March 2003.

[SZ25] Omri Shmueli and Mark Zhandry. On one-shot signatures, quantum vs. classical binding, and obfuscating permutations. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025, Part II*, volume 16001 of *Lecture Notes in Computer Science*, pages 350–383, Santa Barbara, CA, USA, August 17–21, 2025. Springer, Cham, Switzerland.