On the Complexity of Collision Resistant Hash Functions: New and Old Black-Box Separations

Nir Bitansky[1] and Akshay Degwekar[2]

[1] Tel Aviv University, Tel Aviv, Israel.
nirbitan@tau.ac.il,
[2] MIT, Cambridge, MA, US, akshayd@alum.mit.edu

Abstract. The complexity of collision-resistant hash functions has been long studied in the theory of cryptography. While we often think about them as a Minicrypt primitive, black-box separations demonstrate that constructions from one-way functions are unlikely. Indeed, theoretical constructions of collision-resistant hash functions are based on rather structured assumptions.

We make two contributions to this study:

1. *A New Separation:* We show that collision-resistant hashing does not imply hard problems in the class Statistical Zero Knowledge in a black-box way.
2. *New Proofs:* We show new proofs for the results of Simon, ruling out black-box reductions of collision-resistant hashing to one-way permutations, and of Asharov and Segev, ruling out black-box reductions to indistinguishability obfuscation. The new proofs are quite different from the previous ones and are based on simple *coupling arguments*.

No Title Given

No Author Given

No Institute Given

Table of Contents

1 Introduction

Collision-resistant hash functions (CRHFs) are perhaps one of the most studied and widely used cryptographic primitives. Their applications range from basic ones like "hash-and-sign" [**?**, **?**] and statistically hiding commitments [**?**, **?**] to more advanced ones like verifiable delegation of data and computation [**?**, **?**] and hardness results in complexity theory [**?**, **?**].

Constructions. Collision resistance is trivially satisfied by random oracles and in common practice, to achieve it, we heuristically rely on unstructured hash functions like SHA. Accordingly, we often think of CRHFs as a creature of *Minicrypt*, the realm of symmetric key cryptography [**?**]. However, when considering theoretical constructions with formal reductions, collision resistance is only known based on problems with some algebraic structure, like Factoring, Discrete Log, and different short vector and bounded distance decoding problems (in lattices or in binary codes) [**?**, **?**, **?**, **?**, **?**, **?**, **?**]. Generic constructions are known from claw-free permutations [**?**, **?**], homomorphic primitives [**?**, **?**], and private information retrieval [**?**], which likewise are only known from similar structured assumptions. An exception is a recent work by Holmgren and Lombardi [**?**] which constructs CRHFs from a new assumption called *one-way product functions*. These are functions where efficient adversaries succeed in inverting two random images with probability at most $2^{-n-\omega(\log n)}$. Indeed, this assumption does not explicitly require any sort algebraic structure.

Understanding the Complexity of CRHFs. In light of the above, it is natural to study what are the minimal assumptions under which CRHFs can be constructed, and whether they require any sort of special structure. Here Simon [**?**] provided an explanation for our failure to base CRHFs on basic Minicrypt primitives like one-way functions or one-way permutations. He showed that there are no black-box reductions of CRHFs to these primitives. In fact, Asharov and Segev [**?**] demonstrated that the difficulty in constructing CRHFs from general assumptions runs far deeper. They showed that CRHFs cannot be black-box reduced even to *indistinguishability obfuscation* (and one-way permutations), and accordingly not to anyone of the many primitives it implies, like public key encryption, oblivious transfer, or functional encryption.

CRHFs and SZK. An aspect common to many CRHF constructions is that they rely on assumptions that imply hardness in the class SZK. Introduced by Goldwasser, Micali and Rackoff [**?**], SZK is the class of *promise problems* with statistical zero-knowledge proofs. Indeed, SZK hardness is known to follow from various algebraic problems that lead to CRHFs, such as Discrete Logarithms [**?**], Quadratic Residuosity [**?**], and Lattice Problems [**?**, **?**], as well as from generic primitives that lead to CRHFs such as homomorphic encryption [**?**], lossy functions [**?**], and computational private information retrieval [**?**].

The formal relation between SZK and CRHFs is still not well understood. As possible evidence that SZK hardness may be sufficient to obtain collision resistance, Komargodski and Yogev [**?**] show that average-case hardness in SZK implies a relaxations of CRHFs known as *distributional* CRHFs. Applebaum

and Raykov [?] show that CRHFs are implied by average-case hardness in a subclass of SZK of problems that have a *perfect randomized encoding*. Berman et al. [?] showed that average-case hardness of a variant of entropy approximation, a complete problem for the class of Non-Interactive SZK (NISZK), suffices to construct yet a different relaxation known as *multi-collision resistance*.

Is hardness in SZK necessary for CRHFs? Our perception of CRHFs as a Minicrypt primitive, as well as the result by Holmgren and Lombardi mentioned above, suggest that this should not be the case. However, we do not know how to prove this. Meaningfully formalizing a statement of the form "CRHFs do not require SZK hardness" requires care — it is commonly believed that SZK *does* contain hard problems, and if this is the case then formally, CRHFs (or any other assumption for that matter) imply hardness in SZK. To capture this statement we again resort to the methodology of black-box separations; that is, we aim to prove that hard problems in SZK cannot be obtained from CRHFs in a black-box way.

Recent work by Bitansky, Degwekar, and Vaikuntanathan [?] showed that a host of primitives, essentially, all primitives known to follow from IO, do not lead to hard problems in SZK through black-box reductions. Their separation, however, does not imply a separation from CRHFs; indeed, CRHFs are not known to follow from IO, and in fact according to Asharov and Segev [?], cannot in a black-box way.

1.1 This Work

In this work, we close the above gap, proving that CRHFs do not imply hardness in SZK through black-box reductions.

Theorem 1.1. *There are no fully black-box reductions of any (even worst-case) hard problem in* SZK *to CRHFs.*

Here by *fully* black box we mean reductions where both the construction and the security proof are black box in the CRHF and the attacker, respectively. This is the common type of reductions used in cryptography. We refer the reader to the technical overview in Section 2 for more details.

New proofs of Simon and Asharov and Segev. Our second contribution is new proofs for the results of Simon [?], ruling out fully black-box reductions of CRHFs to OWPs,[1] and of Asharov and Segev [?], ruling out black-box reductions of CRHFs to OWPs and IO. The new proofs draw from ideas used in [?]. They are based mostly on simple *coupling arguments* and are quite different from the original proofs.

1.2 More Related Work on Black-Box Separations

Following the seminal work of Impagliazzo and Rudich [?], black-box separations in cryptography have been thoroughly studied (see, e.g., [?, ?, ?, ?, ?, ?, ?, ?,

[1] Simon also ruled out a stronger type of reductions known as semi-black-box reductions [?]. We only rule out the notion of fully black-box reductions described above.

?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?]). Most of this study has been devoted to establishing separations between different cryptographic primitives and some of it to putting limitations on basing cryptographic primitives on NP-hardness [?, ?, ?, ?, ?, ?].

Perhaps most relevant to our works are the works of Simon [?], Asharov and Segev [?] and [?] mentioned above, as well as the work by Haitner et al. [?] who gave an alternative proof for the Simon result (and extended it to the case of statistically-hiding commitments of low round complexity).

We also note that [?] claim to show that distributional CRHFs cannot be reduced to multi-collision resistant hash functions in a black box way, which given the black-box construction of distributional CRHFs from SZK hardness [?], would imply that SZK hardness cannot be obtained from multi-collision resistance in a black box way. However, for the time being there seems to be a gap in the proof of this claim [?].

2 Techniques

We now give an overview of the techniques behind our results.

Ruling Out Black-Box Reductions. Most constructions in cryptography are fully black-box [?], in the sense that both the construction and (security) reduction are black box. In a bit more detail, a fully black-box construction of a primitive $\mathcal{P}'$ from another primitive $\mathcal{P}$ consists of two algorithms: a construction C and a reduction R. The construction $\mathsf{C}^{\mathcal{P}}$ implements $\mathcal{P}'$ for any valid oracle $\mathcal{P}$. The reduction $\mathsf{R}^{\mathsf{A},\mathcal{P}}$, given oracle-access to any adversary A that breaks $\mathsf{C}^{\mathcal{P}}$, breaks the underlying $\mathcal{P}$. Hence, breaking the instantiation $\mathsf{C}^{\mathcal{P}}$ of $\mathcal{P}'$ is at least as hard as breaking $\mathcal{P}$ itself.

A common methodology to rule out fully black black-box constructions of a primitive $\mathcal{P}'$ from primitive $\mathcal{P}$ (see e.g., [?, ?, ?]), is to demonstrate oracles (Γ, A) such that:

- relative to Γ, there exists a construction C^{Γ} realizing $\mathcal{P}$ that is secure in the presence of A,
- but *any* construction C'^{Γ} realizing $\mathcal{P}'$ can be broken in the presence of A.

Indeed, if such oracles (Γ, A) exist, then no efficient reduction will be able to use (as a black-box) the attacker A against $\mathcal{P}'$ to break $\mathcal{P}$ (as the construction of $\mathcal{P}$ is secure in the presence of A).

We now move on to explain how each of our results is shown in this framework.

2.1 Collision Resistance When SZK is Easy

Our starting point is the work by [?] who showed oracles relative to which Indistinguishability Obfuscation (IO) and One-Way Permutations (OWPs) exist and yet SZK is easy. We next recall their approach and explain why it falls short

of separating CRHFs from SZK. We then explain the approach that we take in order to bridge this gap.

Black-box Constructions of SZK Problems. The [**?**] modeling of problems in SZK follows the characterization of SZK by Sahai and Vadhan [**?**] through its complete Statistical Difference Problem (SDP). SDP is a promise problem, where given circuit samplers (C_0, C_1), the task is to determine if the statistical distance between their respective output distributions is large ($> 2/3$) or small ($< 1/3$). Accordingly, we can model a black-box construction of a statistical distance problem SDP^{Ψ}, relative to an oracle Ψ, defined by

$$\mathrm{SDP}^{\Psi}_{Y} = \left\{ (C_0, C_1) : \mathrm{SD}(C_0^{\Psi}, C_1^{\Psi}) \geq \frac{2}{3} \right\} ,$$
$$\mathrm{SDP}^{\Psi}_{N} = \left\{ (C_0, C_1) : \mathrm{SD}(C_0^{\Psi}, C_1^{\Psi}) \leq \frac{1}{3} \right\} .$$

Jumping ahead, our eventual goal will be construct an oracle $\Gamma = (\Psi, \mathsf{A})$ such that SDP^{Ψ} is easy in the presence of A, and yet Ψ can be used to securely realize a CRHF, in the presence of A. Here we naturally choose Ψ to be a random shrinking function f, and for the SZK breaker A adopt the oracle SDO^f from [**?**]. SDO^f is a randomized oracle that takes as input a pair of oracle-aided circuits $(C_0^{(\cdot)}, C_1^{(\cdot)})$, computes the statistical distance $s = \mathrm{SD}(C_0^f, C_1^f)$, samples a random value $t \leftarrow (1/3, 2/3)$, and outputs:

$$\mathsf{SDO}^f(C_0, C_1; t) := \begin{cases} N & \text{If } s < t \\ Y & \text{If } s \geq t \end{cases} .$$

This oracle is clearly sufficient to break (or rather, decide) SDP^f. The challenge is in showing that CRHFs exist in the presence of the oracle SDO^f, which may make exponentially many queries to f when computing the statistical distance.

One-Way Permutations in the Presence of SDO. Toward proving the existence of CRHFs in the presence of SDO, we first recall the argument from [**?**] as to why one-way permutations exist relative to SDO, and then explain why it falls short of establishing the existence of CRHFs.

Consider the oracle $\Gamma = (f, \mathsf{SDO}^f)$, where f is a random permutation. Showing that $f(x)$ is hard to invert for an adversary $\mathsf{A}^{f,\mathsf{SDO}^f}(f(x))$ with access to f and SDO^f relies on two key observations:

1. Inverting f requires detecting random *local changes*. Indeed, imagine an alternative experiment where we replace f with a slightly perturbed function $f_{x' \to f(x)}$, which diverts a random x' to $f(x)$. In this experiment, the attacker would not be able to distinguish x from x' and would output them with the exact same probability. Note, however, that if the attacker can invert f in the real experiment (namely, output x) with noticeable probability, then this means that the probabilities of outputting x and x' in the original experiment must noticeably differ. Indeed, in the original experiment x' is independent

of the attacker's view. It is not hard to show that without access to the oracle SDO^f, such perturbations cannot be detected (this can be shown for example via a coupling argument, as we explain in more detail in Section 2.2).

2. The SDO^f oracle itself, and thus $\mathsf{A}^{f,\mathsf{SDO}^f}$, can be made oblivious to random, local changes. Hence, even given access to the SDO^f oracle, the adversary cannot invert with non-trivial probability. This is shown based on the idea of "smoothening": any two circuits (C_0^f, C_1^f) can be transformed into new circuits that do not make any specific query x with high probability. This allows arguing that even if we perturb f at a given point, their statistical distance s does not change by much. In particular, if s is moderately far from the random threshold t, chosen by SDO, s' the statistical distance of the perturbed circuits remains on the same side of t, which means that SDO's answer will remain invariant. Indeed, such "farness" holds with overwhelming probability over SDO's choice of t.

What About Collision Resistance? The above approach is not sufficient to argue that collisions are hard to find (when f is replaced with a shrinking function). The reason is that collisions are "non-local" — they are abundant, and it is impossible to eliminate all of them in a shrinking function. In fact, as we shall show later on, a similar argument to the one above can be made to work relative to an oracle that trivially breaks CRHFs (this leads to our new proofs of the separations of CRHFs from OWPs and IO [**?**, **?**]). Accordingly, a different approach is required.

Our Approach: Understanding What Statistical Difference Oracles Reveal. At high level, to show that collisions in f are hard to find, we would like to argue that queries to SDO^f leak no information about any $f(x)$, except for inputs x, which the adversary had already explicitly revealed by querying f itself. This would essentially reduce the argument to the standard argument showing that random oracles are collision resistant — each new query collides with any previous query with probability at most 2^{-m}, where m is f's output length. Overall, an attacker making q queries cannot find a collision except with negligible probability $q^2 2^{-m}$.

However, showing that SDO^f reveals nothing is too good to be true. Rather, we show that this is the case with overwhelming probability. That is, with overwhelming probability on any partial execution, the value $f(x)$ of any x not explicitly queried within the execution is uniformly random. Roughly speaking, the property that such partial executions should satisfy is that all queries to SDO^f satisfy smoothness and farness conditions similar to those discussed above. The essential observation is that when such conditions hold the answer of SDO^f remains invariant not only to a random local change, but to *any* local change. In particular, a partial execution transcript satisfying these conditions would remain invariant if we change the value $f(x)$ for any x not explicitly queried to any particular $y \neq f(x)$.

A Note on Leakage from Random Oracles. Our approach is in part inspired by the works of Unruh [**?**] and Coretti et al. [**?**] on *random oracles with auxiliary information*. They show that revealing short auxiliary information about f (so called leakage), essentially has the effect of fixing f on a small set of values, while the rest of f remains hidden. This does not suffice for us, because it does not restrict in any way which values are fixed. We need to ensure that *all* values not explicitly queried remain hidden even under the leakage from the oracle SDO. (Our argument is restricted though to the specific oracle SDO and does not say anything about arbitrary leakage.)

2.2 Proving Simon & Asharov-Segev : A Coupling-Based Approach

Next, we sketch the main ideas underlying the new proofs of Simon's result that OWPs do not imply CRHFs through fully black-box constructions, and the extended result by Asharov and Segev, which consider not only OWPs, but also IO. In this overview, we focus on the simpler result by Simon. We refer the reader to the full version of this paper for the extension to IO.

Simon's Collision Finding Oracle. The oracle $\Gamma = (f, \mathsf{Coll}^f)$ introduced by Simon consists of a random permutation f and a collision finding oracle Coll^f. The oracle Coll^f given a circuit C^f returns a random w along with a random element that collides with w; namely a random w' in the preimage of $y = C^f(w)$. In particular, if the circuit C is compressing, then the oracle will output a collision $w \neq w'$ with high probability, meaning that CRHFs cannot exist in its presence.

Our Proof. To prove that Coll does not help inverting f, Simon used careful conditional probability arguments, whereas Haitner et al. [**?**], and then Asharov and Segev [**?**] adding also IO to the picture, relied on a *compression and reconstruction argument*, originally due to Gennaro and Trevisan [**?**].Our proof is inspired by the [**?**] proof that the statistical distance oracle SDO does not help inverting permutations (discussed above). At high level, we would like to argue that the collision-finding oracle Coll, like the oracle SDO, is oblivious to random local changes. Following the intuition outlined for SDO, an attacker that fails to detect random local changes will also fail in inverting random permutations.

Punctured Collision Finders. To fulfil this plan, we consider a *punctured* version PColl of the oracle Coll, where the function f can be erased at a given set of values S. Roughly speaking, PColl will allow us to argue that Coll is not particularly sensitive to the value $f(x)$ of almost any x. To define PColl, we first give a more concrete description of Coll and then explain how we change it.

The oracle Coll, for any circuit $C : \{0,1\}^k \to \{0,1\}^*$, assigns a random input $w \in \{0,1\}^k$ and a random permutation π of $\{0,1\}^k \simeq [2^k]$. It then returns (w, w'), where w' is the first among $\pi(1), \pi(2), \ldots$ such that $C^f(w) = C^f(w')$. The oracle PColl^f_S is parameterized by a set of punctured inputs $S \subseteq \{0,1\}^n$. Like Coll, for any C, it samples a random input w and a permutation π. Differently from Coll, if $C^f(w)$ queries any $x \in S$, the oracle returns $\perp$. Else, it iterates

over the inputs $\{0,1\}^k$ according to π and finds the first value w' such that (1) $C^f(w')$ makes no queries to any $x \in S$, and (2) $C^f(w) = C^f(w')$. The oracle outputs the collision (w, w').

The PColl oracle satisfies the following essential property. Let τ be a transcript generated by the attacker $\mathsf{A}^{f,\mathsf{Coll}^f}$ and assume that for all Coll answers (w, w') in τ, neither $C^f(w)$ nor $C^f(w')$ query any $x \in S$. Then $\mathsf{A}^{f,\mathsf{PColl}^f_S}$ generates the exact same transcript τ. Indeed, this follows directly from the definition of the punctured oracle PColl.

Proving Hardness of Inversion by Smoothening and Coupling. Equipped with the punctured oracle, we now explain how it can be used argue the hardness of inversion. We first consider a smoothening process analogous to the one considered in the statistical distance separation discussed above. That is, we make sure that (with overwhelming probability) all queries C made to Coll are smooth in the sense that $C^f(w)$ does not query any specific input with high probability when w is chosen at random. We then make a few small perturbations to our oracles, and argue that they are undetectable by a coupling argument. Finally, we deduce univertability.

Step 1: Let x be the preimage that $\mathsf{A}^{f,\mathsf{Coll}^f}(f(x))$ aims to find. We first consider, instead of Coll, the punctured oracle $\mathsf{PColl}^f_{\{x\}}$. Due to smoothness, almost every transcript produced by $\mathsf{A}^{f,\mathsf{Coll}^f}(f(x))$ is such that x is not queried by $C^f(w), C^f(w')$ for any query C and answer (w, w') returned by Coll . Any transcript satisfying the latter can be coupled with an identical transcript generated by $\mathsf{A}^{f,\mathsf{PColl}^f_{\{x\}}}(f(x))$, and deduce that the probability of inversion (outputting x) in this new experiment E_1 is close to the probability in the original experiment E_0.

Step 2: We perturb the oracle again. We sample a random $x' \leftarrow \{0,1\}^n$ and make the following two changes: (1) we change the oracle f to $f_{x' \to f(x)}$, which diverts x' to $f(x)$, and (2) we puncture at x', namely, we consider $\mathsf{PColl}^f_{\{x,x'\}}$.

We next observe that in this new experiment E_2, x and x' are symmetric. Accordingly, x and x' are output with the same probability in the experiment E_2. To complete the proof, we apply a coupling argument to show that x and x' are output with *almost* the same probability also in the previous experiment E_1. This is enough as in E_1 the view of the attacker is independent of x', which will allows us to deduce that the probability of inversion is negligible overall.

Let us describe the coupling argument more explicitly. Both experiments E_1 and E_2 are determined by the choice of f, x, x' and randomness $R = \{w, \pi\}$ for Coll. We can look at the events $X_1 = X_1(f, x, x', R)$ and $X_2 = X_2(f, x, x', R)$, where X_1 occurs when the attacker outputs x in the experiment E_1 and X_2 occurs when it outputs x in E_2. Similarly, we can look at X'_1 and X'_2, which describe the events that x' is output in each of the experiments. Then by coupling, we know that

$$\left|\Pr[X_1] - \Pr[X_2]\right| \leq \Pr_{f,x,x',R}[I_{X_1} \neq I_{X_2}] \ ,$$

where I_{X_1}, I_{X_2} are the corresponding indicators. The same holds for X_1', X_2'. Thus, we can bound:

$$\begin{aligned}\left|\Pr[X_1] - \Pr[X_1']\right| &\leq \left|\Pr[X_1] - \Pr[X_2]\right| + \left|\Pr[X_2] - \Pr[X_2']\right| + \left|\Pr[X_1'] - \Pr[X_2']\right| \\ &\leq \Pr_{f,x,x',R}[I_{X_1} \neq I_{X_2}] + \quad 0 \quad + \Pr_{f,x,x',R}\left[I_{X_1'} \neq I_{X_2'}\right] \ .\end{aligned}$$

It is left to see that when fixing f, x, R the outputs in the two experiments E_1, E_2 (and thus also X_1, X_2 and X_1', X_2') are identical as long as x' does not coincide with any of the queries to f, nor with any of the queries induced by any $\mathsf{PColl}_{\{x\}}$ answer (w, w'). Since the number of such queries is bounded and x' is chosen independently at random, this will almost surely be the case.

Organization

In Section **??**, we provide relevant preliminaries. In Section **??**, we prove that there are no fully black-box reductions of SZK hardness to CRHFs. In Section **??**, we reprove Simon's result that there are no fully black-box reductions of CRHFs to OWPs. The extension of this result to IO can be found in the full version of this paper.

3 Preliminaries

In this section, we introduce the basic definitions and notation used throughout the paper.

3.1 Conventions

For a distribution D, we denote the process of sampling from D by $x \leftarrow D$. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}^+$ is negligible if for every constant c, there exists a constant n_c such that for all $n > n_c$ $\mathsf{negl}(n) < n^{-c}$.

Randomized Algorithms. As usual, for a random algorithm A, we denote by $\mathsf{A}(x)$ the corresponding output distribution. When we want to be explicit about the algorithm using randomness r, we shall denote the corresponding output by $\mathsf{A}(x; r)$. We refer to uniform probabilistic polynomial-time algorithms as PPT algorithms.

Oracles. We consider *oracle-aided algorithms (or circuits)* that make repeated calls to an oracle Γ. Throughout, we will consider deterministic oracles Γ that are a-priori sampled from a distribution Γ on oracles. More generally, we consider infinite oracle ensembles $\Gamma = \{\Gamma_n\}_{n\in\mathbb{N}}$, one distribution Γ_n for each security parameter $n \in \mathbb{N}$ (each defined over a finite support). For example, we may consider an ensemble $f = \{f_n\}$ where each $f_n : \{0,1\}^n \to \{0,1\}^n$ is a random function. For such an ensemble Γ and an oracle aided algorithm (or circuit) A with finite running time, we will often abuse notation and denote by $\mathsf{A}^{\Gamma}(x)$

and execution of A on input x where each of (finite number of) oracle calls that A makes is associated with a security parameter n and is answered by the corresponding oracle Γ_n. When we write $\mathsf{A}_1^{\Gamma}, \ldots, \mathsf{A}_k^{\Gamma}$ for k algorithms, we mean that they all access the same realization of Γ.

3.2 Coupling and Statistical Distance.

Definition 3.1 (Coupling). *Given two random variables X, Y over $\mathcal{X}, \mathcal{Y}$, a* coupling *of X, Y is defined to be any distribution $P_{X'Y'}$ on $\mathcal{X} \times \mathcal{Y}$ such that, the marginals of $P_{X'Y'}$ on $\mathcal{X}$ and $\mathcal{Y}$ are the distributions X, Y respectively.*

Denote by $\mathcal{P}_{XY}$ the set of all couplings of X, Y.

Lemma 3.2. *Given any two distributions X, Y supported on $\mathcal{X}$,*

$$\mathrm{SD}(X,Y) = \inf_{P_{X'Y'} \in \mathcal{P}_{XY}} \Pr_{(x,y) \leftarrow P_{X'Y'}}[x \neq y].$$

Furthermore, for distributions over a discrete domain $\mathcal{X}$ the infimum is attained: that is, there exists a coupling P_{XY} such that $\mathrm{SD}(X,Y) = \Pr_{(x,y)\leftarrow P_{XY}}[x \neq y]$.

The lemma allows us to bound the statistical distance between two random variables (hybrid experiments in our case) by setting up a coupling between two experiments and bounding the probability of them giving a different outcome. Looking ahead, in **??**, we describe an explicit coupling for the Simon's collision finder oracle, of the form above that allows us to bound the statistical distance between hybrids.

4 Separating SZK and CRHFs

4.1 Fully Black-Box Constructions of SZK Problems

The class of problems with Statistical Zero Knowledge Proofs (SZK) [**?**, **?**] can be characterized by complete promise problems [**?**], particularly statistical difference, and the transformation is black-box. In order to consider black-box constructions of hard problems in SZK, we start by defining statistical difference problem relative to oracles. This modelling follows [**?**].

In the following definition, for an oracle-aided (sampler) circuit $C^{(\cdot)}$ with n-bit input and an oracle Ψ, we denote by $\mathbf{C}^{\Psi}$ the output distribution $C^{\Psi}(r)$ where $r \leftarrow \{0,1\}^n$. We denote statistical distance by SD: for two distributions X and Y $\mathrm{SD}(X,Y) = \frac{1}{2}\sum_x |\Pr[X=x] - \Pr[Y=x]|$.

Definition 4.1 (Statistical Difference Problem relative to oracles). *For an oracle Ψ, the statistical difference promise problem relative to Ψ, denoted as* $\mathrm{SDP}^{\Psi} = (\mathrm{SDP}^{\Psi}_Y, \mathrm{SDP}^{\Psi}_N)$*, is given by*

$$\mathrm{SDP}^{\Psi}_Y = \left\{ (C_0, C_1) : \mathrm{SD}(\mathbf{C}_0^{\Psi}, \mathbf{C}_1^{\Psi}) \geq \frac{2}{3} \right\} ,$$

$$\mathrm{SDP}^{\Psi}_N = \left\{ (C_0, C_1) : \mathrm{SD}(\mathbf{C}_0^{\Psi}, \mathbf{C}_1^{\Psi}) \leq \frac{1}{3} \right\} .$$

Next, we define formally define fully black-box reductions from CRHFs to SZK.

Definition 4.2 (Black-Box Construction of SZK-hard Problems). *A* fully black-box construction of a hard statistical distance problems (SDP) from CRHFs *consists of*

- ***Black-box construction:*** *A collection of oracle-aided circuit pairs* $\Pi^{(\cdot)} = \left\{\Pi_n^{(\cdot)}\right\}_{n\in\mathbb{N}}$ *where* $\Pi_n = \left\{(C_0^{(\cdot)}, C_1^{(\cdot)}) \in \{0,1\}^{n\times 2}\right\}$ *such that each* (C_0, C_1) *defines an SDP instance.*
- ***Black-box security proof:*** *A probabilistic oracle-aided reduction* R *with functions* $q_{\mathsf{R}}(\cdot), \varepsilon_{\mathsf{R}}(\cdot)$ *such that the following holds: Let* f *be any distribution on functions. For any probabilistic oracle-aided* A *that* decides Π in the worst-case*, namely, for all* $n \in \mathbb{N}$*,*

$$\Pr\left[\mathsf{A}^f(C_0, C_1) = B \qquad \textit{for all} \qquad \begin{array}{l}(C_0, C_1) \in \Pi_n, B \in \{Y, N\}\\ \textit{such that } (C_0, C_1) \in \mathrm{SDP}_B^f\end{array}\right] = 1$$

the reduction breaks collision resistance of f*, namely, for infinitely many* $n \in \mathbb{N}$*,*

$$\Pr_f\left[f_n(x) = f_n(x') \textit{ where } (x, x') \leftarrow \mathsf{R}^{f,\mathsf{A}}\right] \geq \varepsilon_{\mathsf{R}}(n) \ ,$$

where R *makes at most* $q_{\mathsf{R}}(n)$ *queries to any of its oracles* (A, f) *where each query to* A *consists of circuits* C_0, C_1 *each of which makes at most* $q_{\mathsf{R}}(n)$ *queries to* f*.*

Next, we state the main result of this section: that any fully black-box construction of SDP problems from CRHFs has to either run in time exponential in the security parameter or suffer exponential security loss.

Theorem 4.3. *For any fully black-box construction* $(\Pi, \mathsf{R}, q_{\mathsf{R}}, \varepsilon_{\mathsf{R}})$ *of SDPs from CRHFs, the following holds:*

1. *(The reduction runs in exponential time.)* $q_{\mathsf{R}}(n) \geq 2^{n/10}$*. Or,*
2. *(Reduction succeeds with exponentially small probability.)* $\varepsilon_{\mathsf{R}}(n) \leq 2^{-n/10}$*.*

We prove the theorem by describing an oracle $\Gamma = (f, \mathsf{A})$ such that, A solves SDP^f but f is a CRHF relative to Γ. The rest of the section is devoted to describing this oracle and proving the theorem. We start by describing the adversary that breaks SDP: the statistical distance oracle.

4.2 The Statistical Distance Oracle

Next we describe the statistical distance oracle SDO from [**?**] that solves SZK instances.

Definition 4.4 (Oracle SDO^{Ψ}). *The oracle consists of $t = \{t_n\}_{n\in\mathbb{N}}$ where $t_n : \{0,1\}^{2n} \to (\frac{1}{3}, \frac{2}{3})$ is a uniformly random function. Given n-bit descriptions of oracle-aided circuits $C_0, C_1 \in \{0,1\}^n$, let $t^* = t_n(C_0, C_1)$, and let $s = \mathrm{SD}(\mathbf{C}_0^{\Psi}, \mathbf{C}_1^{\Psi})$, return*

$$\mathsf{SDO}^{\Psi}(C_0, C_1; t) := \begin{cases} 0 & \text{If } s < t^* \\ 1 & \text{If } s \geq t^* \end{cases}$$

It is immediate to see that SDO^{Ψ} decides SDP^{Ψ} in the worst-case.

Claim 4.4.1. For any oracle Ψ,

$$\mathrm{SDP}^{\Psi} \in \mathsf{P}^{\Psi, \mathsf{SDO}^{\Psi}} \ .$$

Remark 4.5 (On the Oracle Used). Our separation is sensitive to the oracle used. Subsequent to [**?**], [**?**] observed that the Simon's collision finding oracle Coll can be used to decide SZK. Clearly, no separation between CRHFs and SZK holds relative to the Simon's oracle. It turns out that Simon's oracle can be used to estimate a different measure of distance between distributions, the Triangular Discrimination,[2] which like statistical distance also gives an SZK-complete promise problem [**?**]. Our separation does hold with a variant of Coll and SDO that measures triangular discrimination, but does not output a collision.

4.3 Insensitivity to Local Changes

Next, we recall the notions of smoothness and farness from [**?**] that are used to argue that the SDO^{Ψ} oracle is insensitive to local changes. Roughly speaking *farness* says that the random threshold t used for a query (C_0, C_1) to SDO^{Ψ} is "far" from the actual statistical distance. [**?**] show that with high probability over the choice of random threshold t, farness holds for all queries (C_0, C_1) made to SDO^{Ψ} by any (relatively) efficient adversary. This intuitively means that changing the distributions $(\mathbf{C}_0^{\Psi}, \mathbf{C}_1^{\Psi})$, on sets of small density, will not change the oracle's answer. The proofs are included in **??** for completeness.

Definition 4.6 ((Ψ, t, ε)-Farness). *Two oracle-aided circuits $(C_0, C_1) \in \{0,1\}^n$ satisfy (Ψ, t, ε)-$\mathsf{farness}$ if the statistical difference $s = \mathrm{SD}(\mathbf{C}_0^{\Psi}, \mathbf{C}_1^{\Psi})$ and threshold t are ε-far:*

$$|s - t| \geq \varepsilon \ .$$

For an adversary A, we denote by $\mathsf{farness}(\mathsf{A}, \Psi, \varepsilon)$ the event that every SDO query (C_0, C_1) made by $\mathsf{A}^{\Psi, \mathsf{SDO}^{\Psi}}$ satisfies (Ψ, t, ε)-farness, where $t = t_n(C_0, C_1)$ is the threshold sampled by SDO.

---

[2] The triangular discrimination is defined as $\mathrm{TD}(X, Y) = \frac{1}{2}\sum_x \frac{(\Pr[X=x] - \Pr[Y=x])^2}{(\Pr[X=x] + \Pr[Y=x])}$. This measure also lies in the interval $[0, 1]$ and is a metric.

Lemma 4.7 ([?](Claim 3.7)). *Fix any Ψ and any oracle-aided adversary A such that $\mathsf{A}^{\Psi,\mathsf{SDO}^{\Psi}}$ makes at most q queries to SDO^{Ψ}. Then*

$$\Pr_{t}[\mathsf{farness}(\mathsf{A},\Psi,\varepsilon)] \geq 1 - 6q\varepsilon \ ,$$

where the probability is over the choice t of random thresholds by SDO.

We now turn to define the notion of *smoothness*. Roughly speaking we will say that an oracle-aided circuit C is smooth with respect to some oracle Ψ if any specific oracle query is only made with small probability. In particular, for a pair of smooth circuits (C_0, C_1), local changes to the oracle Ψ should not change significantly the statistical distance $s = \mathrm{SD}(\mathbf{C}_0^{\Psi}, \mathbf{C}_1^{\Psi})$.

Definition 4.8 ((Ψ,ε)-Smoothness). *A circuit $C^{(\cdot)}$ is (Ψ,ε)-smooth, if every location $x \in \{0,1\}^*$ is queried with probability at most ε. That is,*

$$\max_{x} \Pr_{w}\left[C^{\Psi}(w) \text{ queries } \Psi \text{ at } x\right] < \varepsilon \ .$$

For an adversary A, we denote by $\mathsf{smooth}(\mathsf{A},\Psi,\varepsilon)$ the event that in every SDO query (C_0, C_1) made by $\mathsf{A}^{\Psi,\mathsf{SDO}^{\Psi}}$ both circuits are (Ψ,ε)-smooth.

Lemma 4.9 ([?](Claim 3.9)). *Let Ψ, Ψ' be oracles that differ on at most c values in the domain. Let C_0 and C_1 be (Ψ,ε)-smooth. Let $s = \mathrm{SD}(C_0^{\Psi}, C_1^{\Psi})$ and $s' = \mathrm{SD}(C_0^{\Psi'}, C_1^{\Psi'})$ then $|s - s'| \leq 2c\varepsilon$.*

The above roughly means that (under the likely event that farness holds) making smooth queries should not help the adversary detect local changes in the oracle Ψ. [?] show that we can always "smoothen" the adversary's circuit at the expense of making (a few) more queries to Ψ, which intuitively deems the statistical difference oracle SDO^{Ψ} useless altogether for detecting local changes in Ψ.

In what follows, a (q', q)-query algorithm A makes at most q' queries to the oracle Ψ and q queries to SDO^{Ψ} such that for each query (C_0, C_1) to SDO, the circuits C_0, C_1 themselves make at most q queries to Ψ on any input.

Lemma 4.10 (Smoothing Lemma for SDO [?](Lemma 3.10)). *For any (q, q)-query algorithm A and $\beta \in \mathbb{N}$, there exists a $(q + 2\beta q^2, q)$-query algorithm S such that for any input $z \in \{0,1\}^*$ and oracles $\Psi, \mathsf{SDO}^{\Psi}$:*

1. *$\mathsf{S}^{\Psi,\mathsf{SDO}^{\Psi}}(z)$ perfectly simulates the output of $\mathsf{A}^{\Psi,\mathsf{SDO}^{\Psi}}(z)$,*
2. *$\mathsf{S}^{\Psi,\mathsf{SDO}^{\Psi}}(z)$ only makes queries (C_0, C_1) where both C_0, C_1 are (Ψ,ε)-smooth queries to SDO^{Ψ} with probability:*

$$\Pr_{\mathsf{S}}[\mathsf{smooth}(\mathsf{S},\Psi,\varepsilon)] \geq 1 - 2^{-\varepsilon\beta + \log(2q^2/\varepsilon)} \ ,$$

over its own random coin tosses.

4.4 Collision Resistance in the Presence of SDO Oracle.

In this section, we prove the oracle separation between collision resistant hash functions and SZK.

Let $\mathcal{F}_n$ be the set of all functions from $\{0,1\}^n$ to $\{0,1\}^{m(n)}$ where $m(n) < n$ is a shrinking function. Let $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ denote the family of these sets of functions. Let $\mathcal{T} = \{\mathcal{T}_n\}_{n\in\mathbb{N}}$ where $\mathcal{T}_n$ denotes the set of threshold functions $t : \{0,1\}^n \to (1/3, 2/3)$. [3]

Definition 4.11 (The Oracle f). *The oracle $f = \{f_n\}_{n\in\mathbb{N}}$ on input $x \in \{0,1\}^n$ returns $f_n(x)$ where $f_n : \{0,1\}^n \to \{0,1\}^m$ is a random function from $\mathcal{F}_n$.*

The oracle we consider is $\Gamma = (f, \mathsf{SDO}^f)$. It is easy to see that all $\mathrm{SDP}^f \in \mathsf{P}^{f,\mathsf{SDO}^f}$. What remains to show is that f is still collision resistant in the presence of the SDO^f oracle. We do so next.

Theorem 4.12. *Let A be a (q,q)query adversary for $q = O(2^{m/10})$. Then,*

$$\Pr\Big[f_n(x) = f_n(x') \text{ where } (x,x') \leftarrow \mathsf{A}^{f,\mathsf{SDO}^f}(1^n)\Big] \le 2^{-m/10} .$$

Proof. Fix oracle $f_{-n} = \{f_k\}_{k\neq n}$ arbitrarily. Consider the $(q + 2\beta q^2, q)$query smooth version S, of A given by Lemma **??** for $\beta = 2^{m/5} \cdot m$ and $\varepsilon = 2^{-m/5}$. We assume w.l.o.g that S makes no repeated oracle queries and that whenever S outputs a collision (x, x'), x is its last oracle query and x' is a previous query (both to the f oracle).

The first assumption is w.l.o.g because S may store a table of previously made queries and answers. The second is w.l.o.g because S may halt once its f-queries include a collision and output that collision; also, if one, or both, outputs x, x' have not been queried, S can query it at the end (and if needed change the order of the output so that x is queries last). The latter costs at most two additional queries, and does not affect the smoothness of S.

Next, we define some notation about transcripts generated in the process.

Transcripts. A transcript π consists of all queries asked and answers received by S to the oracle (f, SDO^f). Let x_i denote the i-th query to the f-oracle. We say that $x \notin \pi$ if the location x is not among the queries explicitly made in π.

The Underlying Joint Distribution. The proof infers properties of the joint distribution (f, t, π) consisting of the oracle f, the SDO oracle's random thresholds t and the transcript generated by S. The distribution is generated as follows: $f \leftarrow \mathcal{F}$ and $t \leftarrow \mathcal{T}$ and $\pi \leftarrow \mathsf{S}^{f,\mathsf{SDO}^{f;t}}$ where $\mathsf{SDO}^{f;t}$ denotes running the SDO oracle with random thresholds t. Denote this distribution by $P_{FT\Pi}$.

[3] While we describe the threshold function as a real valued function, it can be safely discretized because statistical distance for any pair of circuits $C_0, C_1 : \{0,1\}^m \to \{0,1\}^*$, takes values that are multiples of $2^{-(m+1)}$. We omit the details here.

Note that given f, t, the transcript π is generated in a deterministic manner as S is deterministic and the oracle's behavior is completely specified. Furthermore, we also consider partial transcripts obtained by running S and stopping after i queries. This transcript is denoted by $\pi_{<i}, x_i$: that is the $\pi_{<i}$ consists of queries and responses received and x_i is the next query to the oracle f. Note that x_i is a deterministic function of $\pi_{<i}$. Given the distribution $P_{FT\Pi}$, the conditional distributions $P_{FT|\Pi=\pi}$ or $P_{FT|\Pi=\pi_{<i}}$ are well defined: these consist of uniform distribution on pairs (f, t) that when run using S result in the transcript being π (or $\pi_{<i}$).

The Good Event. We define the concept of Good transcripts. Roughly speaking, these are transcripts π that satisfy sufficient smoothness and farness so to guarantee that that the value $f(x)$ at any $x \notin \pi$ is completely hidden.

Definition 4.13 (Good). *A tuple $(f, t, \pi, x, \varepsilon)$ is good, denoted by $\mathsf{good}(f, t, \pi, x, \varepsilon)$ if the following hold:*

1. $\pi = \mathsf{S}^{f_{x\to\perp}, \mathsf{SDO}^{f_{x\to\perp}; t}}(1^n)$*, where $f_{x\to\perp}$ is the function equal to f everywhere except at x where it takes the value $\perp$.*
2. *(x is not explicitly queried:) $x \notin \pi$.*
3. *(Transcript is smooth:) Every SDO-query made by $\mathsf{S}^{f_{x\to\perp}, \mathsf{SDO}^{f_{x\to\perp}; t}}(1^n)$ is $(f_{x\to\perp}, 2\varepsilon)$-smooth. Denote this event by $\mathsf{smooth}(f_{x\to\perp}, t, \pi, 2\varepsilon)$.*
4. *(Transcript is far:) Every SDO-query (C_0, C_1) made by $\mathsf{S}^{f_{x\to\perp}, \mathsf{SDO}^{f_{x\to\perp}; t}}(1^n)$, satisfies $(f_{x\to\perp}, t, 12\varepsilon)$-farness where $t = t(C_0, C_1)$. Denote this by $\mathsf{far}(f, t, \pi, 12\varepsilon)$.*

The key reason for using $f_{x\to\perp}$ instead of f in the definition is that when an execution of $\mathsf{S}^{f_{x\to\perp}, \mathsf{SDO}^{f_{x\to\perp}; t}}$ generates a transcript π while making only smooth and far queries, all executions of $\mathsf{S}^{f_{x\to z}, \mathsf{SDO}^{f_{x\to z}; t}}$ for all z, *also generate* π while not necessarily being smooth or far themselves.

A tuple (f, t, π, ε) is good if for all $x \notin \pi$, $\mathsf{good}(f, t, \pi, x, \varepsilon)$ holds.

Lemma 4.14. *Let $P_{FT\Pi}$ as defined above. Then,*

$$\Pr_{(f,t,\pi)\leftarrow P_{FT\Pi}}[\mathsf{good}(f, t, \pi, \varepsilon)] \geq 1 - 16q\varepsilon - 2^{-\beta\varepsilon + \log(2q^2/\varepsilon)}$$

The same holds for i-length partial transcripts generated as well, for all i.

Lemma 4.15. *For any transcript π and query $x \notin \pi$ such that*

$$\Pr_{(f,t,\pi)\leftarrow P_{FT\Pi}}[\mathsf{good}(f, t, \pi, x, \varepsilon)] > 0 \ ,$$

it holds that,

$$\left\{f(x) : (f, t) \leftarrow P_{FT|\Pi=\pi, \mathsf{good}(f,t,\pi,x,\varepsilon)}\right\} \equiv U_m \ .$$

Next, we prove **??** assuming **????**. Then, we prove the two lemmas.

Let $\mathsf{hit}(\pi)$ denote the event that π contains two queries x, x' such that $f_n(x) = f_n(x')$. Then,

$$\begin{aligned}\Pr_{f,t}\Big[f_n(x) = f_n(x') \wedge (x,x') = \mathsf{S}^{f,\mathsf{SDO}^{f,t}}(1^n)\Big] &= \Pr_{f,t,\pi}[\mathsf{hit}(\pi)] \\ &\leq \Pr_{f,t,\pi}[\mathsf{hit}(\pi) \wedge \mathsf{good}(f,t,\pi,\varepsilon)] \\ &+ \Pr_{f,t,\pi}\Big[\overline{\mathsf{good}(f,t,\pi,\varepsilon)}\Big] \ .\end{aligned}$$

We will bound the two terms separately. The first term will involve using **??** while the second term is bound using **????**. We begin by bounding the first term. This is done by decomposing the probability of hitting a collision by the first query that hits a collision:

$$\begin{aligned}&\Pr_{f,t}[\mathsf{hit}(\pi) \wedge \mathsf{good}(f,t,\pi,\varepsilon)] \\ &\leq \sum_i \Pr_{f,t}\Big[\mathsf{hit}(\pi_{\leq i}) \wedge \overline{\mathsf{hit}(\pi_{<i})} \wedge \mathsf{good}(f,t,\pi_{<i},\varepsilon)\Big] \\ &= \sum_i \Pr_{f,t}\Big[f(x_i) \in \mathsf{hitSet}(\pi_{<i}) \wedge \overline{\mathsf{hit}(\pi_{<i})} \wedge \mathsf{good}(f,t,\pi_{<i},\varepsilon)\Big] \ ,\end{aligned}$$

where $x_i \notin \pi$ denotes the i-th f query made by S and $\mathsf{hitSet}(\pi_{<i})$ denotes the answers to f-queries in $\pi_{<i}$,

$$\begin{aligned}&= \sum_i \sum_{\pi_{<i},x_i} \Pr_{f,t}\Big[(\pi_{<i},x_i) = \mathsf{S}^{f,\mathsf{SDO}^{f,t}}(1^n) \wedge \mathsf{good}(f,t,\pi_{<i},x_i,\varepsilon)\Big] \\ &\cdot \Pr_{f,t \leftarrow P_{FT|\Pi=\pi_{<i},\mathsf{good}}}[f(x_i) \in \mathsf{hitSet}(\pi_{<i})]\end{aligned}$$

The last equality follows from the definition of conditional probability. At this point, we can use **??** to argue that

$$\Pr_{f,t \leftarrow P_{FT|\Pi=\pi_{<i},\mathsf{good}(f,t,\pi_{<i},x_i,\varepsilon)}}[f(x_i) \in \mathsf{hitSet}(\pi_{<i})] \leq \frac{i}{2^m}$$

because $f(x_i)$ is uniformly random and $|\mathsf{hitSet}(\pi_{<i})| \leq i$. Hence, we get that,

$$\begin{aligned}&\leq \sum_i \frac{i}{2^m} \cdot \sum_{\pi_{<i},x_i} \Pr_{f,t}\Big[(\pi_{<i},x_i) = \mathsf{S}^{f,\mathsf{SDO}^{f,t}}(1^n) \wedge \mathsf{good}(f,t,\pi_{<i},x_i,\varepsilon)\Big] \\ &\leq \sum_{i=1}^{q'} \frac{i}{2^m} \leq \frac{q'^2}{2^m} \ ,\end{aligned}$$

where $q' = q + 2\beta q^2 + 2$, the number queries that S makes to f.