Cryptographic Hashing From Strong One-Way Functions*

Or: One-Way Product Functions and their Applications

Justin Holmgren† Alex Lombardi‡

Abstract

Constructing collision-resistant hash families (CRHFs) from one-way functions is a long-standing open problem and source of frustration in theoretical cryptography. In fact, there are strong negative results: black-box separations from one-way functions that are $2^{-(1-o(1))n}$-*secure* against polynomial time adversaries (Simon, EUROCRYPT '98) and even from indistinguishability obfuscation (Asharov and Segev, FOCS '15).

In this work, we formulate a mild strengthening of exponentially secure one-way functions, and we construct CRHFs from such functions. Specifically, our security notion requires that every polynomial time algorithm has at most $2^{-n} \cdot \mathrm{negl}(n)$ probability of inverting *two independent challenges*.

More generally, we consider the problem of simultaneously inverting k functions $f_1, \ldots, f_k$, which we say constitute a "one-way product function" (OWPF). We show that sufficiently hard OWPFs yield hash families that are multi-input correlation intractable (Canetti, Goldreich, and Halevi, STOC '98) with respect to all sparse (bounded arity) output relations. Additionally assuming indistinguishability obfuscation, we construct hash families that achieve a broader notion of correlation intractability, extending the recent work of Kalai, Rothblum, and Rothblum (CRYPTO '17). In particular, these families are sufficient to instantiate the Fiat-Shamir heuristic in the plain model for a natural class of interactive proofs.

An interesting consequence of our results is a potential new avenue for bypassing black-box separations. In particular, proving (with necessarily non-black-box techniques) that parallel repetition amplifies the hardness of specific one-way functions – for example, all one-way permutations – suffices to directly bypass Simon's impossibility result.

*This work was done in part while the authors were visiting the Weizmann Institute of Science in January 2018, supported by the Binational Science Foundation (Grants No. 2016726, 2014276) and European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482) and Project PROMETHEUS (Grant 780701).

†Massachusetts Institute of Technology. Email: `holmgren@mit.edu`. Research supported in part by NSF Grant CNS-1413920

‡Massachusetts Institute of Technology. Email: `alexjl@mit.edu`. Research supported in part by NSF Grants CNS-1350619 and CNS-1414119, and by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236.

Contents

1 Introduction

Cryptographically secure hash functions are a fundamental building block in cryptography. Some of their most ubiquitous applications include the construction of digital signature schemes [NY89], efficient CCA-secure encryption [BR93], succinct delegation of computation [Kil94], and removing interaction from protocols [FS86]. In their most general form, hash functions can be modeled as "random oracles" [BR93], in which case it is heuristically assumed that an explicitly described hash function H (possibly sampled at random from a family) behaves like a random function, as far as a computationally bounded adversary can tell.

One of the most basic properties one might desire from a hash function is *collision resistance*, which requires that a computationally bounded adversary, given an explicit (shrinking) function H, cannot find a pair of distinct inputs (x, y) such that $H(x) = H(y)$. Since their introduction [Dam87], collision-resistant hash functions have proved extremely useful in designing cryptographic primitives and protocols. As such, the following problem has received much attention in theoretical cryptography.

Question 1.1. *What are the assumptions from which collision-resistant hash functions can be built? In particular, can they be built from an arbitrary one-way function?*

The question of building CRHFs from arbitrary one-way functions is particularly intriguing because OWFs are sufficient to construct a wide class of cryptographic primitives, including: pseudorandom generators [HILL99], pseudorandom functions [GGM86] and secret-key encryption, universal one-way hash functions [Rom90] and digital signatures, commitment schemes [Nao91], zero-knowledge proofs [GMW91], and garbled circuits [Yao86, LP09].

Unfortunately, all known constructions of CRHFs have required assumptions beyond general one-way functions, such as *structured* generic assumptions (e.g. the existence of claw-free pairs of permutations) or the hardness of specific problems (e.g. computing discrete logarithms or finding approximately short vectors on lattices). Even worse, there are strong negative results on the prospect of constructing CRHFs from arbitrary OWFs in the form of *black-box impossibility results*. The first such result is due to Simon [Sim98].

Theorem 1.2 ([Sim98], informal)**.** *There is an oracle relative to which no collision-resistant hash functions exist, but* exponentially secure *one-way permutations exist.*

In fact, CRHFs have proved to be an extremely frustrating primitive in theoretical cryptography, as they have evaded attempts to describe a hierarchy of cryptographic primitives (with "weaker" objects implied by the existence of "stronger" objects). In a stark demonstration of this problem, Asharov and Segev [AS15] proved that CRHFs are not even implied (in a black box[1] way) by one-way functions and the extremely powerful notion of indistinguishability obfuscation [BGI+01, GGH+13].

Theorem 1.3 ([AS15], informal)**.** *There is an oracle relative to which no collision-resistant hash functions exist, but* exponentially secure *one-way permutations and indistinguishability obfuscation exist.*

These negative results indicate substantial barriers to building CRHFs from OWFs (or OWPs, or indeed from any of the vast array of primitives implied by IO and OWPs). Collision resistance

[1] "Black box" usage of IO and one-way functions is formalized through the notion of obfuscation for *oracle-aided* circuits. We refer the reader to [AS15] for details.

is also just *one* desirable property of random oracles, and our question above is a special case of the following more ambitious question.

Question 1.4. *Which random oracle properties can be guaranteed under standard cryptographic assumptions, and how weak can these assumptions be made?*

It is known that some random oracle properties are *not realizable* in the standard model [CGH04, GK03]. However, there has been a recent line of work [CCR16, KRR16, CCRR18] showing that under strong assumptions, many random oracle properties (specifically in the context of "single input correlation intractability") *can* be realized, and Question 1.4 in its full generality remains wide open.

1.1 Our Contributions

In this work, we make progress on all of the above questions by defining a natural strengthening of exponentially secure OWFs[2] that suffices for building CRHFs and more. An "uber" version of our assumption – which we state for the purpose of intuition but is quantitatively and qualitatively much stronger than what we actually require – states that for every $k = \text{poly}(n)$, there exists an injective (polynomial-time computable) function $f : \{0,1\}^* \to \{0,1\}^*$ with the following "batch one-wayness" property: For every polynomial-size adversary $\mathcal{A}$, the probability that $\mathcal{A}(f(X_1), \ldots, f(X_k)) = (X_1, \ldots, X_k)$ for $X_1, \ldots, X_k \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^n$ is bounded by $2^{-kn} \cdot \text{poly}(n)$.

Based on various significant *weakenings* of this uber-assumption, we construct:

- Collision-resistant hash families whose security against polynomial-time adversaries matches that of a random oracle.
- More generally, for every k, we construct hash families $\mathcal{H}$ that are "k-ary output intractable" (inspired by a related definition of Zhandry [Zha16]). Loosely speaking, given $H \leftarrow \mathcal{H}$, it is computationally hard to find distinct inputs $X_1, \ldots, X_k$ such that $(H(X_1), \ldots, H(X_k))$ satisfy any fixed sparse relation R. The quantitative hardness that we achieve again matches that of a random oracle.

We are able to construct even stronger hash families if we additionally assume sub-exponentially secure indistinguishability obfuscation. This construction allows for applications including an instantiation of the Fiat-Shamir heuristic [FS86] for a natural class of interactive proofs.

Our main results and contributions are, in more detail, as follows.

1.1.1 Defining OWPFs.

We introduce the notion of a family of one-way k-product functions (k-OWPFs), which is a family of k-tuples of functions $(f_1, \ldots, f_k)$ that are jointly "extremely one-way". Such a family is most interesting when the hardness of inversion exceeds that of any individual f_i. For simplicity, suppose that each f_i is injective. In this case, we consider the assumption that no polynomial-time algorithm can recover $X_1, \ldots, X_k \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^n$ given $(f_1(X_1), \ldots, f_k(X_k))$ with probability better than δ. Ideally, this could be true for δ as large as $2^{-(k-o(k))n}$. We call this a δ-hardness assumption of *batch inversion* for $(f_1, \ldots, f_k)$.

The existence of such a family would follow from the following two conditions:

[2] Actually, OWFs where any *polynomial-time* algorithm can invert with only exponentially small probability

- A $\delta^{1/k}$-secure injective one-way function f, and
- An optimal *parallel repetition theorem* for the hardness of f, i.e. one which states that if a function f is (s, δ)-hard to invert, then its k-wise repetition f^k is (s, δ^k)-hard to invert.

While such a dream parallel repetition property likely does not hold for *general* f [DJMW12], the counterexample presented therein does not preclude a similar result for a broad class of functions f.

In fact, the parallel repetition framework described above yields a special kind of OWPF family: one in which all k functions $f_1, \ldots, f_k$ are equal. We say that such OWPF families are *symmetric*. Another special case of interest, which we call a *one-way power family*, is a OWPF family of the form $\mathcal{F}^k$, meaning that the k functions $f_1, \ldots, f_k$ are sampled independently at random from a fixed family $\mathcal{F}$.

Our constructions (that do not require obfuscation) are based directly on symmetric injective OWPFs as a building block rather than general OWPFs. We agument these constructions by providing generic transformations between different notions of OWPFs, including constructions of (weaker) symmetric OWPFs from (stronger) general OWPFs, and constructions of *injective* k-OWPFs from arbitrary k-OWPFs (with some security loss).

One of our main contributions in this work is initiating the study of OWPFs and establishing their basic properties. We expect that OWPFs will prove useful in future work.

On Extreme Hardness Amplification

For all of our constructions without obfuscation, we actually rely on *symmetric* OWPF families. That is, we want a family $\mathcal{F} = \{\mathcal{F}_n\}$ such that if we sample $f \leftarrow \mathcal{F}_n$ and $x_1, \ldots, x_k \leftarrow \{0,1\}^n$, it is δ^k-hard to simultaneously invert $f(x_1), \ldots, f(x_k)$. Clearly a necessary condition for this is that $\mathcal{F}$ is a δ-secure one-way function family. But is this sufficient? The answer in general is no, as we discuss next.

First of all, this type of attempted hardness amplification fails for any family whose functions have short trapdoors that enable polynomial-time inversion. Given $f, f(x_1), \ldots, f(x_k)$, an adversary can simply guess the trapdoor for f, succeed with some small probability *that does not depend on* k, and conditioned on guessing correctly can efficiently invert $f(x_1), \ldots, f(x_k)$.

It is natural to next consider *functions* (or ensembles of functions $\{f_n : \{0,1\}^n \to \{0,1\}^*\}_n$ indexed only by input length) that are secure against non-uniform adversaries, and in particular do not have any trapdoors. However, [DJMW12] present an example of a single one-way function f for which it is as easy to invert $f(x_1), \ldots, f(x_k)$ as it is to invert a single $f(x)$. Although their counterexample heavily relies on the fact that there are multiple permissible solutions to each instance x, there is also evidence that parallel repetition sometimes fails to increase the security of *injective* one-way functions [Wic18].

Despite the above negative results, we emphasize that symmetric OWPFs only require direct products to amplify hardness for *specific* functions, rather than broad classes of functions. Moreover, one-way product functions may exist even if parallel repetition does not amplify the hardness of *any* function f beyond negligible. In particular, $f_1, \ldots, f_k$ may not all be the same function, and may be sampled from a joint distribution on k-tuples of functions. These observations leave us with at least two promising avenues towards constructing OWPF candidates:

1. Given the contrived nature of known counterexamples to one-way function parallel repetition, any "natural" δ-secure injective OWF family also serves as a candidate one-way power family with security roughly δ^k.

2. It may be possible to "fortify" any one-way function family $\mathcal{F}$ into a related family $\mathcal{F}'$ whose security *does* amplify to an extreme degree, yielding symmetric OWPFs.

Finally, we mention a concrete candidate symmetric OWPF family based on the *multiple discrete logarithm problem*. That is, in some group $\mathbb{G}_n$ of order $|\mathbb{G}_n| \approx 2^n$, the problem is to simultaneously compute k discrete logarithms $X_1, \ldots, X_k \overset{\text{i.i.d.}}{\leftarrow} [2^n]$ given input $(g, g^{X_1}, \ldots, g^{X_k})$, where g is a generator for $\mathbb{G}_n$. In [CK18], evidence for the hardness of computing multiple discrete logarithms is given in the form of lower bounds in the generic group model [Sho97]. In particular, [CK18] show that (in our language) k-batch inversion is nearly 2^{-kn}-hard for polynomial-time generic-group algorithms.

1.1.2 Constructions from OWPFs

Our first application of OWPFs is a construction of a collision-resistant hash family from suitably secure symmetric 2-OWPFs. Informally, we prove

Theorem 1.5. *Suppose that there exist symmetric injective 2-OWPFs with security $2^{-n-\omega(\log n)}$. Then, there exists a collision-resistant hash family.*

This type of OWPF does not follow in a black-box way from even exponentially-hard one-way permutations; this is how we avoid the [Sim98, AS15] impossibility results.

Through one of our generic transformations of OWPFs, we also obtain a construction that does not assume injectivity:

Theorem 1.6. *Suppose that there exist symmetric 2-OWPFs with security $2^{-(1.6+\epsilon)n}$. Then, there exists a collision-resistant hash family.*

Optimality and Implications of Theorem 1.5.

While we have explained how our result is not captured by the [Sim98, AS15] framework, one could question the necessity of this new OWPF assumption. For example, [AS15] only rules out black-box constructions of CRHFs from $2^{-\epsilon n}$-secure IO and one-way permutations (for $\epsilon = \frac{1}{50}$ in particular), and [Sim98] proves a quantitatively similar impossibility. What about assuming only $2^{-n/2}$-secure OWPs, which are weaker and more standard than our symmetric OWPFs? As a complementary result, we show that these are insufficient – we strengthen the Asharov-Segev analysis to rule out black box constructions from IO and even 2^{-n}-secure one-way permutations.

Theorem 1.7 (Extension of [AS15] Theorem 1.1, informal)**.** *There is no black-box construction of CRHFs from sub-exponentially secure IO, sub-exponentially secure OWPs, and OWPs that ppt algorithms $\mathcal{A}$ can invert with probability at most $\text{size}(\mathcal{A})^c \cdot 2^{-n}$ for some absolute constant c.*

Theorem 1.7 indicates a sharp limit on directly improving Theorem 1.5; in the latter, we show that injective 2-OWPFs that are $2^{-n} \cdot \text{negl}(n)$-hard to invert suffice for constructing CRHFs from IO, while the former result says that improving the $2^{-n} \cdot \text{negl}(n)$ to $\frac{2^{-n}}{\text{negl}(n)}$ is impossible for black-box

constructions. In particular, for black-box constructions, exponentially secure one-way permutations (in the usual sense) are insufficient.

As far as we know, Theorem 1.5 is the first OWF-based construction in which security under parallel repetition is used "for cryptographic good"; that is, used to construct more expressive primitives than can be achieved by (exponentially secure) OWFs alone, at least in a black-box manner.

Extension to Output Intractability

Theorem 1.5 can be substantially generalized beyond collision-resistance. In particular, given a $2k$-ary relation, we consider the problem of finding $X_1, \ldots, X_k$ such that $(X_1, \ldots, X_k, H(X_1), \ldots, H(X_k)) \in R$ for $H \leftarrow \mathcal{H}_n$. If this problem is hard, then $\mathcal{H}$ is said to be multi-input correlation intractable for R, a notion due to [CGH04]. Collision-resistance is the special case when $k = 2$ and

$$R = \{(x_1, x_2, y_1, y_2) : (x_1 \neq x_2) \wedge (y_1 = y_2)\}.$$

Random oracles are correlation intractable for any *sparse* relation R – that is, as long as for every $\mathbf{x} = (x_1, \ldots, x_k)$, $\Pr_{\mathbf{Y} \leftarrow (\{0,1\}^{n-1})^k}[(\mathbf{x}, \mathbf{Y}) \in R] \leq \text{negl}(n)$. In many applications, this correlation-intractability is the crucial property of a random oracle, and a fundamental theoretical question is whether it can be achieved by *concrete* hash families.

Despite the initial negative result of [CGH04], which ruled out correlation intractability for arbitrary (e.g., unbounded-arity) relations, there has been substantial work on constructing hash families that are correlation intractable for "bounded" single-input/output relations [CCR16, KRR16, CCRR18] as well as hash families that are "output intractable" [Zha16], that is, correlation intractable with respect to relations of the form "$(x_i \neq x_j$ for all $i \neq j) \wedge R(y_1, \ldots, y_k) = 1$."[3]

Using suitably secure k-OWPFs, we construct hash families that are output intractable for *all* sparse output relations (with known bounded arity). The quantitative intractability that we prove depends on the sparsity of the relation, similarly to the situation for a true random oracle. Equivalently, we rely on weaker assumptions to show correlation-intractability of sparser relations.

A simplified version of our result is as follows.

Theorem 1.8 (informal)**.** *Suppose that there exists a family of symmetric injective k-OWPFs with security $(s + \text{poly}(n), \delta)$, let $m = m(n)$ denote any output length, and let $p = p(n)$ denote any sparsity. Then, there exists a hash family $\mathcal{H} = \{\mathcal{H}_{n,m(n)}\}$ that is output intractable, with security $(s, \delta \cdot p \cdot 2^{kn})$, with respect to all k-ary relations of sparsity p.*

In particular, if the k-OWPF family has optimal (2^{-kn}) security, then the hash family constructed in Theorem 1.8 has output intractability matching that of a random oracle.

As an interesting special case, we note that Theorem 1.8 gives a construction of k-multi-collision resistant hash functions [BDRV18, BKP18, KNY18] from symmetric injective k-OWPFs with security $2^{-n-k\log(k)} \cdot \text{negl}(n)$, an assumption which (up to a lower order term in the exponent) becomes weaker as k increases from 2 to any $o(\frac{n}{\log(n)})$. As any multi-collision-resistant hash family implies the existence of constant round statistically hiding commitments [BDRV18, KNY18], this yields constant round statistically hiding commitments from $2^{-n} \cdot \text{negl}(n)$-secure (injective and symmetric) k-OWPFs for any $k = o(\frac{n}{\log(n)})$. Unlike the assumptions required for collision resistance, this assumption would follow from optimal parallel repetition for *any polynomially secure (injective) one-way function.*

[3] [Zha16] considers a slightly different notion of output intractability. We elaborate on this later.

1.1.3 Combining OWPFs with Indistinguishability Obfuscation

Our results above, Theorem 1.5 and Theorem 1.8, are constructions of cryptographic hash families from (symmetric) OWPFs alone, and hence (partially) address the question of what hash families can be constructed from assumptions in the realm of one-wayness.

We additionally consider which hash families can be constructed in the plain model under stronger assumptions. Namely, we combine OWPFs with the powerful notion of indistinguishability obfuscation [BGI+01, GGH+13]. This line of reasoning yields another construction of CRHFs, and more generally a construction of multi-input correlation intractable hash functions for a broader class of relations than achieved by Theorem 1.8. In our IO-based construction, we are able to handle relations R which depend on both the input variables $\mathbf{x}$ and the output variables $\mathbf{y}$, as long as the relation R is efficiently *locally* samplable. Informally, we need to be able to efficiently sample a random output $\mathbf{Y}$ such that $(\mathbf{x}, \mathbf{Y}) \in R$ such that each output Y_i is sampled only knowing the corresponding input x_i (with arbitrary preprocessed shared randomness "between the variables").

Moreover, our construction is extremely simple and confirms typical intuition about obfuscation: our hash family is an obfuscated (puncturable) PRF $\mathcal{O}(F_s(\cdot))$. We only require the existence of suitably secure OWPFs in the security proof; they are not needed in the construction. This result extends the framework of [CCR16, KRR16] on constructing strong hash functions from obfuscation (and additional assumptions).

Our main result utilizing obfuscation (Theorem 6.3) is stated and proved in Section 6.3. The result is proved by viewing OWPFs themselves as a (weak) form of obfuscation: an injective k-OWPF $(f_1, \cdots, f_k)$ allows us to obfuscate *multi-point functions*, i.e., programs of the form

$$P_{x_1,\ldots,x_k}(x) = \begin{cases} i & x = x_i \text{ for some } i \\ 0 & \text{otherwise.} \end{cases}$$

Since this construction is oblivious to whether or not the OWPF family $\mathcal{F}$ is symmetric, this yields a construction of correlation intractable hash families (and in particular, of CRHFs) relying on weaker OWPF assumptions, at the cost of additionally assuming IO. That is, the assumptions on asymmetric OWPFs required here are quantitatively (and even qualitatively) weaker than those required without obfuscation, as we avoid the cost of converting asymmetric OWPFs into symmetric OWPFs.

As an interesting special case, the notion of correlation intractability that we achieve in Theorem 6.3 is powerful enough to capture nontrivial cases of the Fiat-Shamir paradigm for converting (constant round, public-coin) interactive proof systems into non-interactive argument systems. One such formal result is stated in Theorem 6.6, but the main intuition is that we can instantiate the Fiat-Shamir transform for proof systems with the property that a malicious prover can efficiently determine which verifier messages he can cheat on. This intuition captures protocols that follow the "commit-challenge-response" framework using a generic commitment scheme (which is the case that Theorem 6.6 handles). This approach yields a construction of NIZK argument schemes (in the common reference string model) through the Fiat-Shamir transform whose security relies on IO and the existence of exponentially secure one-way functions – no OWPF assumptions are needed in this case.

1.2 Related Work

Extremely Lossy Functions. [Zha16] introduces the notion of an extremely lossy function (ELF). In [Zha16], ELFs are used as a central building block to construct several hash families with strong security properties. In particular, they can be used to construct hash functions satisfying a notion of output intractability that is incomparable to we achieve in Section 5. Informally, [Zha16] considers the more general setting of $k+1$-ary relations $R(y_1, \ldots, y_k, w)$ with the property that for random $(y_1, \ldots, y_k)$, it is computationally hard to find a witness w for which $R(y_1, \ldots, y_k, w) = 1$ (where our notion would correspond to the case that for random $(y_1, \ldots, y_k)$, *no such witness exists*), and constructs hash functions that are correlation intractable for such relations R that are efficiently decidable.

The only current construction of ELFs relies on an exponentially strong DDH assumption. An interesting open question is whether OWPFs imply the existence of ELFs, or even ordinary (i.e. moderately) lossy one-way functions.

CRHFs from Extremely Strong LPN. Two recent works [YZW+17, BLVW18] give constructions of CRHFs from the Learning Parity with Noise (LPN) problem in parameter settings that resemble an exponential hardness assumption. We note that one of the same works [BLVW18] proves that these particular LPN assumptions imply hardness in the complexity class $\mathrm{BPP}^{\mathrm{SZK}}$, placing this construction on similar complexity-theoretic ground as prior constructions from discrete logarithm and SIS. The LPN-based CRHFs are also provably broken in quasi-polynomial time, while our CRHF is plausibly as collision-resistant as a random oracle.

Single-Input Correlation Intractability. Correlation intractability [CGH04] is a clean but powerful property of random oracles that has drawn considerable interest, particularly for its relevance to the Fiat-Shamir transform [FS86, BR93]. Circumventing the negative results of [CGH04, GK03, BDSG+13], there has been a recent line of work [CCR16, KRR16, CCRR18] on constructing (single input) correlation intractable hash functions and instantiating the Fiat-Shamir heuristic in the standard model, under strong assumptions. We build on this line of work, particularly the work of [KRR16], to achieve results for special cases of *multi-input* correlation intractability under weaker or incomparable assumptions than are required in these previous works.

CRHFs from IO and SZK-hardness. [BDV17] constructs CRHFs from indistinguishability obfuscation and any average-case hard problem in the complexity class $\mathsf{SZK}^{0,1}$. We consider SZK-hardness to be a "structured assumption" which makes it different from (even very strong) assumptions on injective one-way functions; indeed, the same work proves an Asharov-Segev-like impossibility result for constructing (even worst-case) hard SZK instances from IO and OWPs. A fascinating open question is whether OWPFs (with or without IO) imply SZK-hardness of any form.

1.3 Technical Overview

We now outline some of our constructions in more detail. In order to clearly demonstrate the power of OWPFs and our techniques, we focus on the following two special cases: constructing CRHFs from symmetric 2-OWPFs, and constructing CRHFs from IO and (asymmetric) injective 2-OWPFs.

1.3.1 Construction of CHRFs

For simplicity, we first assume that we have an ensemble of one-way permutations $\{f_n : \{0,1\}^n \to \{0,1\}^n\}$, where for every constant $c > 0$, double inversion is $2^{-n} \cdot n^{-c}$ hard for size-n^c adversaries. In this case, we construct a particularly simple CRHF: to sample a collision-resistant $H : \{0,1\}^n \to \{0,1\}^{n-1}$, first sample $P : \{0,1\}^n \to \{0,1\}^{n-1}$ from a programmable pairwise independent hash family $\mathcal{P}$ (see Definition 2.6), and define $H = P \circ f_n$. This and similar constructions have proved very useful in prior works [NY89, PW11, Zha16].

We now sketch the proof of security. Assume for contradiction that some poly-size algorithm $\mathcal{A}$ finds collisions in H with probability $\epsilon = \epsilon(n)$. We show how to use $\mathcal{A}$ to simultaneously find $X_1^* = f_n^{-1}(Y_1^*)$ and $X_2^* = f_n^{-1}(Y_2^*)$ with probability roughly $\epsilon \cdot 2^{-n}$, given uniformly random $Y_1^*, Y_2^* \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^n$. Specifically, we will invoke $\mathcal{A}$ not on a uniformly sampled $H = P \circ f_n$, but on a differently defined $H = P_{\text{plant}} \circ f_n$, where P_{plant} is sampled from $\mathcal{P}$ *conditioned on* $P_{\text{plant}}(Y_1^*) = P_{\text{plant}}(Y_2^*)$.

Intuitively, we now argue (by a purely statistical argument) that (X_1^*, X_2^*) looks sufficiently like a *uniformly random* collision of H that $\mathcal{A}$ must output that exact collision with probability roughly $\epsilon \cdot 2^{-n}$. To make this intuition rigorous, suppose first that we ignore Y_1^* and Y_2^*, and simply invoke $\mathcal{A}$ on a randomly sampled $H = P \circ f_n$. Then with probability ϵ, $\mathcal{A}$ will find a collision (X_1, X_2) in H. Conditioned on this event, (X_1, X_2) will be *equal* to (X_1^*, X_2^*) with probability 2^{-2n}, for a total probability of $\epsilon \cdot 2^{-2n}$ that both events occur. But (X_1^*, X_2^*) is a collision in H with probability only $2^{-(n-1)}$. Thus, conditioning on this event (i.e., sampling $H = P_{\text{plant}} \circ f_n$ instead of $H = P \circ f_n$) boosts the probability that $\mathcal{A}$ outputs (X_1^*, X_2^*) to $\epsilon \cdot 2^{-2n} \cdot 2^{n-1} = \epsilon \cdot 2^{-n-1}$.

Therefore, the CRHF we constructed satisfies the standard notion of security: every polynomial-size adversary finds collisions with probability that is negligible in n. From stronger hardness assumptions on $\{f_n\}$, i.e. that double-inversion is $\delta(n)$-hard for size-$s(n)$ adversaries, one obtains a correspondingly more secure CRHF.

Beyond Permutations and Injective One-Way Functions The above argument actually does not rely in any way on f_n being a permutation. It is, however, important that f_n is injective, so that all collisions in $P \circ f_n$ are due to P, and thus in some sense are randomly distributed.

We also show that the injectivity requirement can be traded off against a stronger hardness assumption. In fact, if $\{f_n\}$ is extremely secure to begin with, we can construct a family of functions which is statistically injective, and still nearly as secure.

For simplicity, we illustrate this transformation for *one-way functions*. Suppose that $\{f_n\}$ is $\delta(n)$-hard to invert for polynomial-time adversaries (think of $\delta(n) = 2^{-(1-o(1))n}$, although such extreme parameters are not necessary). We first observe that $\{f_n\}$ cannot be "extremely" non-injective; if one independently samples $X_1 \leftarrow \{0,1\}^n$ and $X_2 \leftarrow \{0,1\}^n$, then the probability that $f_n(X_1) = f_n(X_2)$ must be at most δ (otherwise one could break the security of f_n by random guessing). This can be leveraged to obtain a fully injective function (with some small error probability), as follows.

Set n to be any function of n' (think of $n(n') = 3n'$). Then define the ensemble of function families $\mathcal{F} = \{\mathcal{F}_{n'}\}$ as follows. To sample a function $f \leftarrow \mathcal{F}_{n'}$, sample $P : \{0,1\}^{n'} \to \{0,1\}^n$ from a pairwise independent hash family, and define $\tilde{f}_{n'} = f_n \circ P$. A simple pairwise independence argument shows that $\mathcal{F}$ is statistically injective, with failure probability at most $2^{2n'} \cdot \delta(n)$ (with the suggested parameters in mind, this is $2^{-(1-o(1))n'}$).

Security of $\mathcal{F}$ follows from observing that if an adversary cannot invert $f_n(X)$ with probability better than δ when sampling $X \leftarrow \{0,1\}^n$, then for any subset $\mathcal{X} \subseteq \{0,1\}^n$, the adversary cannot invert $f_n(X')$ with probability better than $\delta \cdot \frac{2^n}{|\mathcal{X}|}$ when sampling $X' \leftarrow \mathcal{X}$. With good probability $(1 - 2^{2n'-n}$, or with our suggested parameters $1 - 2^{-n'})$, it holds that $P : \{0,1\}^{n'} \to \{0,1\}^n$ is actually injective, so that inverting $f_n \circ P$ corresponds to inverting f_n when inputs are drawn from the uniform distribution on $\mathrm{Img}(P)$. The above discussion shows that this is $\delta \cdot 2^{n-n'}$-hard (or with our suggested parameters $2^{-(1-o(1))n'}$-hard) even for adversaries that are given arbitrary advice about P.

While the above description refers to the case of one-way functions (i.e. 1-OWPFs), similar arguments can be made for arbitrary OWPFs (with different quantitative tradeoffs), as discussed in Section 3.3.

1.3.2 Constructions Using Obfuscation

We now outline our general proof strategy – which we informally refer to as the *planting technique* – for all of our constructions based on IO, using collision resistance as an example. The planting technique is inspired by the recent work of Kalai, Rothblum, and Rothblum [KRR16] on instantiating the Fiat-Shamir heuristic using obfuscation.

For simplicity, we focus on hash functions that shrink by a single bit. Our construction is then simply an obfuscation $H \stackrel{\mathsf{def}}{=} \mathcal{O}(F_S)$ of a puncturable pseudorandom function $F_S : \{0,1\}^n \to \{0,1\}^{n-1}$, where $\mathcal{O}$ is an indistinguishability obfuscator. Recall that we also assume the existence of an injective but *not necessarily symmetric* 2-OWPF that cannot be inverted in polynomial time with probability better than $2^{-n-\omega(\log n)}$.

The proof of security then proceeds as follows. Assume for contradiction that some ppt algorithm $\mathcal{A}$ finds a collision (X_1, X_2) of H with non-negligible[4] probability ϵ. We then consider the behavior of $\mathcal{A}$ on an obfuscation of a *different* program H_{plant} which overrides the functionality of F_S with a hard-coded planted collision $H_{\mathrm{plant}}(X_1^*) = H_{\mathrm{plant}}(X_2^*) = Y^*$, for independent and uniformly random X_1^*, X_2^*, and Y^*. That is, the functionality of H_{plant} is

$$H_{\mathrm{plant}}(x) \stackrel{\mathsf{def}}{=} \begin{cases} Y^* & \text{if } x = X_1^* \text{ or } x = X_2^* \\ F_S(x) & \text{otherwise.} \end{cases}$$

We then prove two contradictory claims.

Claim 1 (informal): The probability that $\mathcal{A}$ outputs (X_1^*, X_2^*) is approximately $\epsilon \cdot 2^{-n-1}$, i.e. $2^{-n-O(\log n)}$.

This claim is argued as follows.

(a) If $\mathcal{A}$ is given an obfuscation of a program H_{punc} that (in contrast to H_{plant}) overrides F_S with hard-coded mappings $X_1^* \mapsto Y_1^*$ and $X_2^* \mapsto Y_2^*$ for *independent* uniform $Y_1^*, Y_2^* \leftarrow \{0,1\}^{n-1}$, then the probability that $\mathcal{A}$ successfully produces a collision *and that collision is* (X_1^*, X_2^*) is very nearly $\epsilon \cdot 2^{-2n}$ by the security of $\mathcal{O}$ and F_S.

[4] In fact, our approach readily generalizes to obtain exponentially-secure CRHFs, at the cost of quantitatively stronger computational assumptions.

(b) (X_1^*, X_2^*) is only a valid collision of H_{punc} when $Y_1^* = Y_2^*$, so the probability that $\mathcal{A}$ outputs (X_1^*, X_2^*) conditioned on $Y_1^* = Y_2^*$ is approximately $\epsilon \cdot 2^{-2n} \cdot 2^{n-1} = \epsilon \cdot 2^{-n-1}$. But the distribution of H_{punc} conditioned on $Y_1^* = Y_2^*$ is exactly the distribution of H_{plant}.

Claim 2 (informal): The probability that $\mathcal{A}$ outputs (X_1^*, X_2^*) is $2^{-n-\omega(\log n)}$.

Since IO is the "best-possible" obfuscation [GR07], it suffices for there to exist *some* obfuscation of H_{plant} that hides (X_1^*, X_2^*). This would follow from a "special-purpose" obfuscator $\mathcal{O}'$ for membership testing in two-element sets (in our case $\{X_1^*, X_2^*\}$). The security property we need is that every ppt algorithm recovers (X_1^*, X_2^*) from $\mathcal{O}'(\{X_1^*, X_2^*\})$ with probability bounded by $2^{-n-\omega(\log n)}$.

This is a variant of "point function obfuscation", a notion which was studied by [Can97, CMR98, Wee05]. Our variant (with uniformly random X_1^*, X_2^*) admits a particularly easy construction from injective 2-OWPFs – the obfuscation is $(W_1^* = f_1(X_1^*), W_2^* = f_2(X_2^*))$, and is evaluated on an input x as

$$\begin{cases} 1 & \text{if } f_1(x) = W_1^* \text{ or } f_2(x) = W_2^* \\ 0 & \text{otherwise.} \end{cases}$$

There are conceivably other ways to obtain this point function obfuscation, but for this particular construction, security is equivalent to the hardness of batch inverting (f_1, f_2).

1.4 Conclusions and Questions

In this work, we have introduced a new family of computational assumptions – namely, the existence of various flavors of one-way product functions (OWPFs). We find these assumptions to be clean, plausible, and useful.

In terms of power, OWPFs allow the construction of hash families that achieve several elusive random oracle-like properties. In particular, our black-box construction of CRHFs shows that OWPFs are more powerful than *black box usage* of exponentially-secure one-way functions.

OWPFs are also extremely plausible. Depending on s, δ, and k, we view (s, δ)-secure k-OWPFs as somewhere between standard and exponentially-secure one-way functions. The plausibility is supported by a concrete candidate instantiation – the discrete log problem, which is provably a nearly optimal OWPF in the generic group model.

Indeed, this particular combination of plausibility and usefulness gives us some hope that CRHFs can be constructed solely based on exponentially strong one-way functions. More generally, our results suggest a possible blueprint for circumventing black-box impossibility results from OWFs:

1. Build OWPFs from OWFs (using necessarily non-black-box techniques).
2. Build primitives in a black-box way from OWPFs.

One bonus of this approach is that it could result in constructions that are non-black-box only *in the security proof*, and thus has the potential for practical efficiency.

Independently, OWPFs satisfy several desirable properties for a cryptographic assumption. For example, for any family $\mathcal{F}$, the assumption "$\mathcal{F}$ is a k-OWPF" is a *search complexity assumption* [GK16]: for some efficiently sampleable distribution $\mathcal{D}$ and efficiently checkable relation $\mathcal{R}$, the assumption is equivalent to requiring that on input $x \sim \mathcal{D}$, every bounded-time algorithm has bounded probability of finding y such that $(x, y) \in \mathcal{R}$.

1.4.1 Questions

There remain many intriguing questions about the precise power of OWPFs. In particular:

- What are the complexity-theoretic implications of OWPFs? For example, do they imply hardness in SZK? We emphasize that all prior constructions of CRHFs have been from assumptions that imply (average-case) SZK hardness, but CRHFs themselves are not known to imply any sort of SZK hardness.
- What implies OWPFs? Is it possible to construct non-trivial k-OWPFs from previously studied cryptographic assumptions? Above we outlined an approach to *generically* constructing OWPFs, but it is also possible that OWPFs can be based on concrete, structured assumptions.

1.5 Organization

The rest of the paper is organized as follows. In Section 3, we define OWPFs and discuss the associated hardness assumptions, including a concrete candidate: the multiple discrete logarithm problem. We also prove generic reductions between OWPF notions. In Section 4, we present our construction of collision-resistant hash functions from (suitably secure) symmetric 2-OWPFs. In Section 5, we generalize the construction from Section 4 to obtain output intractable hash functions from symmetric OWPFs. In Section 6, we show that any (IO-)obfuscated puncturable PRF satisfies a broader notion of correlation intractability assuming that suitable OWPFs exist. This includes collision-resistant hash functions and output intractable hash functions from weaker OWPF assumptions as well as an instantiation of the Fiat-Shamir transform for "commit-challenge-response" proof systems. Finally, in Appendix A, we formally state and prove Theorem 1.7, our complementary result showing that Theorem 1.5 is optimal.

2 Preliminaries

We write ppt to denote probabilistic polynomial-time. We say that two distribution ensembles $\{X_n\}$ and $\{Y_n\}$ are δ-indistinguishable if for all polynomial-sized circuit ensembles $\{\mathcal{A}_n\}$,

$$\left| \Pr\left[\mathcal{A}_n(X_n)\right] = 1 - \Pr\left[\mathcal{A}_n(Y_n) = 1\right] \right| \leq O(\delta(n)).$$

For a relation R, we say that $R(x) = 1$ if $x \in R$ and $R(x) = 0$ otherwise.

For any primitive $\mathcal{P}$ whose security is parametrized by a pair $(s(\lambda), \delta(\lambda))$ (denoting time and advantage), we say that $\mathcal{P}$ is *polynomially secure* if $\mathcal{P}$ is $(\lambda^c, 1/\lambda^c)$-secure for all $c > 0$. We say that $\mathcal{P}$ is *sub-exponentially secure* if there exists some $\epsilon > 0$ such that $\mathcal{P}$ is $(2^{\lambda^\epsilon}, 2^{-\lambda^\epsilon})$-secure. We say that $\mathcal{P}$ is *δ-secure* if $\mathcal{P}$ is (λ^c, δ)-secure for all $c > 0$, and we say that $\mathcal{P}$ is *sub-exponential advantage-secure* if there exists some $\epsilon > 0$ such that $\mathcal{P}$ is 2^{-n^ϵ}-secure.

2.1 One-Way Functions

Definition 2.1 (One-Way Functions)**.** *A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is a (s, δ)-secure one-way function (OWF) if for every $\lambda \in \mathbb{N}$ and every circuit ensemble $\{\mathcal{A}_\lambda\}$ of size $|\mathcal{A}_\lambda| \leq s(\lambda)$, it holds that*

$$\Pr_{\substack{x \leftarrow \{0,1\}^\lambda \\ x' \leftarrow \mathcal{A}_\lambda(f(x))}} [f(x') = f(x)] \leq O(\delta(\lambda)).$$

Definition 2.2 (Families of One-Way Functions)**.** *$\mathcal{F} = \{f_I : \mathcal{D}_I \to \mathcal{R}_I\}_{I \in \mathcal{I}}$ is a (s, δ)-secure family of one-way functions if there are ppt algorithms $(\mathsf{Gen}, \mathsf{Samp})$ and a deterministic polynomial-time algorithm Eval with the following syntax:*

- Gen *takes as input a security parameter 1^λ and outputs an index $I \in \mathcal{I}$.*
- Samp *takes as input an index $I \in \mathcal{I}$, and outputs $x \in \mathcal{D}_I$.*
- Eval *takes as input an index $I \in \mathcal{I}$ and $x \in \mathcal{D}_I$, and outputs $y = f_I(x)$.*

Additionally, there is a security requirement that for every circuit $\mathcal{A}$ of size $s(\lambda)$,

$$\Pr_{\substack{I \leftarrow \mathsf{Gen}(1^\lambda) \\ x \leftarrow \mathsf{Samp}(I) \\ x' \leftarrow \mathcal{A}(I, f_I(x))}} [f_I(x') = f_I(x)] \leq O(\delta(\lambda)).$$

For simplicity, we will only consider function families over the domain $\{0,1\}^\lambda$.

2.2 Cryptographic Hash Functions

The following definitions are adopted (with modification) from [Gol04].

Definition 2.3 (Cryptographic Hash Function)**.** *Fix a function $m : \mathbb{N} \to \mathbb{N}$ such that $1^{m(n)}$ is computable from 1^n in polynomial time. A family of functions*

$$\mathcal{H} = \{h_I : \{0,1\}^{n(I)} \to \{0,1\}^{m(n(I))}\}_{I \in \mathcal{I}}$$

is a (cryptographic) hash family *if there is a ppt algorithm Gen and a deterministic polynomial-time Eval such that:*

- *(Efficient Sampling) On input 1^n, Gen outputs an index $I \in \mathcal{I}$ such that $n(I) = n$.*
- *(Admissible Indexing – technical[5]) There is a polynomial-time algorithm that when given $I \leftarrow \mathsf{Gen}(1^n)$ as input, outputs 1^n.*
- *(Efficient Evaluation) For all $I \in \mathcal{I}$ and all $x \in \{0,1\}^{n(|I|)}$, $\mathsf{Eval}(I, x) = h_I(x)$.*

The above definition details the *functionality* of a hash function; there are several security notions that one could require. We first focus on the notion of *k-collision-resistance*, recovering the usual definition of a collision-resistant hash family when $k = 2$.

[5]Roughly, we would like the notion of polynomial-time in the *description length* of a hash function to coincide with the notion of polynomial-time in the security parameter

Definition 2.4 (k-collision-resistance)**.** *A family of cryptographic hash functions*

$$\mathcal{H} = \{h_I : \{0,1\}^{n(I)} \to \{0,1\}^{m(n(I))}\}_{I \in \mathcal{I}}$$

is a (length-restricted) k-collision-resistant hash family (k-CRHF) with security $\delta = \delta(m(\cdot))$ if the following two conditions hold.

- *(Shrinking) $m(n) \leq n - \log(k)$.*
- *(k-Collision-Resistance) For all polynomial-size circuits $\mathcal{A}$,*

$$\Pr_{\substack{I \leftarrow \mathsf{Gen}(1^n) \\ (X_1,\ldots,X_k) \leftarrow \mathcal{A}_n(I)}} [h_I(X_1) = \ldots = h_I(X_k) \text{ but } X_1, \ldots, X_k \text{ are all distinct}] \leq O(\delta(m(n))).$$

We say that $\mathcal{H}$ is polynomially secure *if $\mathcal{H}$ is $1/m(n)^c$-secure for all $c > 0$.*

Definition 2.5 (Universal One-Way Hash Families)**.** *A universal one-way hash family (UOWHF) is a family of cryptographic hash functions*

$$\mathcal{H} = \{h_I : \{0,1\}^{n(|I|)} \to \{0,1\}^{m(n(|I|))}\}_{I \in \mathcal{I}}$$

as in Definition 2.3 which are shrinking as in Definition 2.4, but (2-)collision-resistance is weakened to require only that for all polynomial-size circuits $\mathcal{A}_0, \mathcal{A}_1$, there is a negligible function $\nu(\cdot)$ such that

$$\Pr_{\substack{(X,\mathsf{st}) \leftarrow \mathcal{A}_0(1^n) \\ I \leftarrow \mathsf{Gen}(1^n) \\ X' \leftarrow \mathcal{A}_1(I,\mathsf{st})}} [h_I(X) = h_I(X') \wedge X \neq X'] \leq \nu(m(n)).$$

Finally, we define k-wise independent hash functions, which exist unconditionally.

Definition 2.6 ((Programmable) k-wise Independent Hash Functions)**.** *A family of k-wise independent hash functions is a family of hash functions*

$$\mathcal{H} = \{h_I : \{0,1\}^{n(|I|)} \to \{0,1\}^{m(n(|I|))}\}_{I \in \mathcal{I}}$$

as in Definition 2.3 with the property that for every collection $x_1, \ldots, x_k \in \{0,1\}^n$ of distinct inputs, and every collection $y_1, \ldots, y_k \in \{0,1\}^m$ of (not necessarily distinct) outputs, we have

$$\Pr_{I \leftarrow \mathsf{Gen}(1^n)} [h_I(x_i) = y_i \text{ for all } i] = \frac{1}{2^{km}}.$$

Moreover, we say that $\mathcal{H}$ is programmable *if there is an efficient sampling algorithm* $\mathrm{CondGen}(\mathbf{x}, \mathbf{y})$ *with the property that for every $\mathbf{x} = (x_1, \ldots, x_k)$ and $\mathbf{y} = (y_1, \ldots, y_k)$ as above,* $\mathrm{CondGen}(\mathbf{x}, \mathbf{y})$ *samples from the distribution of $I \leftarrow \mathsf{Gen}(1^n)$ subject to the condition that $h_I(x_i) = y_i$ for all i.*

3 One-Way Product Functions: Definitions and Reductions

In this section, we define one-way product functions and their associated batch inversion problems, we discuss the discrete log problem as a concrete candidate, and we establish reductions between different notions of OWPFs.

Definition 3.1 (k-Batch Inversion, k-OWPFs)**.** *Let $\mathcal{F}$ be a family of k-tuples of functions, i.e.,*

$$\mathcal{F} = \{(f_{1,I}, f_{2,I}, \ldots, f_{k,I})\}_{I \in \mathcal{I}},$$

where each $f_{i,I} : D_{i,I} \to R_{i,I}$. We say that k-batch inversion is $(s(\lambda), \delta(\lambda))$-hard for $\mathcal{F}$ (equivalently $\mathcal{F}$ is a (s, δ)-secure k-OWPF family) if for every size-$s(\lambda)$ circuit $\mathcal{A}$, we have

$$\Pr\left[\forall i \in [k], f_{i,I}(X'_i) = f_{i,I}(X_i)\right] \leq O(\delta(\lambda))$$

in the probability space defined by sampling

1. $I \leftarrow \mathsf{Gen}(1^\lambda)$.
2. *For* $i = 1, \ldots, k$, $X_i \leftarrow \mathsf{Samp}(I_i)$.
3. $(X'_1, \ldots, X'_k) \leftarrow \mathcal{A}(I, f_{1,I}(X_1), \ldots, f_{k,I}(X_k))$.

In the special case $k = 2$, we refer to 2-batch inversion as "double inversion".

For the rest of this paper, we will work only over a fixed domain $\mathcal{D} = \{0,1\}^\lambda$ for simplicity.

Remark 3.1. *For any family $\mathcal{F}$ as above, if any of the families $\mathcal{F}_i := \{f_{i,I}\}_{I \in \mathcal{I}}$ is a family of (s, δ)-secure one-way functions, then k-batch inversion is (s, δ)-hard for $\mathcal{F}$. That is, (s, δ)-secure k-OWPFs follow from (s, δ)-secure OWFs.*

Given Remark 3.1 above, we note that batch inversion assumptions are most naturally suited to the setting where $\delta \leq 2^{c\lambda}$ for some c, i.e., δ is *exponentially small*. Moreover, the batch inversion problem is quite plausibly $(\mathrm{poly}(\lambda), \delta)$-hard for $\delta < 2^{-\lambda}$, i.e. where δ is so small that any one-way function can trivially be inverted with probability δ (by outputting a uniformly random guess).

For any family of k-tuples of functions $\mathcal{F}$, we now state the strongest quantitative assumption that is plausible regarding batch inversion for $\mathcal{F}$ (and in particular, such families exist in the random oracle model).

Definition 3.2 (Optimal Batch Inversion Assumption for $\mathcal{F}$)**.** *There exists a universal constant c such that for every function $s = s(\lambda)$, the k-batch inversion problem for $\mathcal{F}$ is $(s(\lambda), s(\lambda)^{ck} 2^{-k\lambda})$-hard.*

This assumption, while not technically falsifiable in the framework of [Nao03, GW11], is still "morally" falsifiable, and in particular is a complexity assumption in the framework of [GK16].

We now consider two important special cases of k-OWPFs.

Definition 3.3 (Symmetric k-OWPFs)**.** *We say that a family $\mathcal{F}'$ of k-OWPFs is symmetric if for all indices $I \in \mathcal{I}$, we have $f_{1,I} = f_{2,I} = \ldots = f_{k,I}$. In other words, $\mathcal{F}'$ is a family of symmetric k-OWPFs if there is a family $\mathcal{F} = \{f_I\}_{I \in \mathcal{I}}$ such that (1) $\mathcal{F}' = \{(f_I, f_I, \ldots, f_I)\}_{I \in \mathcal{I}}$ and (2) $\mathcal{F}'$ is a family of k-OWPFs.*

As described in the introduction, the existence of a family of δ-secure symmetric k-OWPFs would follow from the following two conditions:

- A $\delta^{1/k}$-secure family $\mathcal{F}$ of injective one-way functions, and

- An optimal *parallel repetition theorem* for the hardness of $\mathcal{F}$, i.e. one which states that if a function $f \leftarrow \mathcal{F}$ is (s, δ)-hard to invert, then its k-wise repetition f^k is (s, δ^k)-hard to invert.

However, such a "dream parallel repetition theorem" (even for a specific family $\mathcal{F}$) is not required for δ-secure k-OWPFs to exist. As an example, for any $k \ll \frac{n}{\log(n)}$, consider the question of obtaining 2^{-n}-secure symmetric k-OWPFs; this is a parameter setting of interest for the application of k-multi-collision resistant hash functions. The existence of such a family would also follow from a 2^{-cn}-secure injective OWF family $\mathcal{F}$, along with a much weaker parallel repetition theorem for the hardness of $\mathcal{F}$; hardness would only have to amplify by a factor of $\frac{1}{c}$ in the exponent after k repetitions.

Definition 3.4 (One-Way Power Families)**.** *We say that a function family $\mathcal{F}'$ is a one-way power family if there is a family $\mathcal{F} = \{f_I\}_{I \in \mathcal{I}}$ such that (1) $\mathcal{F}' = \mathcal{F}^k = \{(f_{I_1}, f_{I_2}, \ldots, f_{I_k})\}_{(I_1,\ldots,I_k) \in \mathcal{I}^k}$ and (2) $\mathcal{F}'$ is a family of k-OWPFs.*

In constrast to symmetric OWPFs, (s, δ)-secure one-way power families follow from the following two conditions.

- A $\delta^{\frac{1}{k}}$-secure family $\mathcal{F}$ of injective one-way functions, and
- A *different* form of (optimal) parallel repetition for $\mathcal{F}$, i.e. one which states that if a function $f \leftarrow \mathcal{F}$ is (s, δ)-hard to invert, then k independently sampled functions $f_1, \ldots, f_k \leftarrow \mathcal{F}$ are (s, δ^k) hard to simultaneously invert.

This alternative form of parallel repetition avoids the issue of breaking f^k by brute-forcing a short trapdoor for f; in the case of one-way power families, each of the k functions would have a different trapdoor.

We again emphasize that these optimal parallel repetition results are far stronger than what is required to obtain many of our applications of OWPFs.

3.1 Concrete Candidate: Discrete Logarithm

The optimal batch inversion assumption above, even in the setting of symmetric k-OWPFs, is supported by the work of [CK18], who consider the *multiple discrete logarithm problem*:

Definition 3.5 (Multiple Discrete Logarithm Problem, informal)**.** *Given a sequence of groups $\mathcal{G} = \{G_\lambda, \lambda \in \mathbb{N}\}$ (with efficiently computable operations and sampling algorithms), the multiple discrete logarithm problem is, given as input $(g, y_1, \ldots, y_k) = (g, g^{x_1}, \ldots, g^{x_k})$ (for uniformly random $x_1, \ldots, x_k$), to return all k discrete logarithms $(x_1, \ldots, x_k)$.*

In [CK18], evidence for the hardness of computing multiple discrete logarithms is given in the form of lower bounds in the generic group model [Sho97]. Specifically, they show

Theorem 3.1 ([CK18] Theorem 8, interpreted)**.** *Any generic group algorithm for the multiple discrete logarithm problem running in time T in a group of order $\Theta(2^\lambda)$ has success probability at most $T^{2k} 2^{-\lambda k} \text{poly}(\log(T), \lambda, k)^k$.*

In other words, the optimal batch inversion assumption holds for generic group discrete logarithms. Moreover, the best known algorithms for multiple discrete logarithm over elliptic curve groups are these generic algorithms, and hence the optimal batch inversion assumption over elliptic curve groups is plausible. This yields a candidate family of symmetric k-OWPFs satisfying optimal batch inversion hardness.

The multiple discrete logarithm problem (as defined above) provides a candidate *symmetric* OWPF family. We could alternatively consider the problem of computing k discrete logarithms, *each over an entirely different group*; this would constitue a candidate (asymmetric) OWPF family. In the special case where the k groups are sampled independently at random from some family, this would constitute a candidate one-way power family.

3.2 OWPFs that are Sufficient for CRHFs

In order to build collision-resistant hash functions, we do not need the optimal double inversion assumption, but the following weaker assumption (albeit for injective functions).

Conjecture 1. *There is a $2^{-\lambda-\omega(\log\lambda)}$-secure injective 2-OWPF family.*

That is, we require that double inversion is $2^{-\lambda} \cdot \mathrm{negl}(\lambda)$-hard (rather than $2^{-2\lambda}$-hard) for polynomial time algorithms. Our correlation intractability results are also achieved under assumptions significantly weaker than the optimal assumption (we state the necessary assumptions in Section 5 and Section 6.3).

In the rest of this section, we describe how to obtain OWPFs of a special form – either symmetric, injective, or both – from more general OWPFs through a few different transformations. We consider these transformations with the goal of obtaining important applications of (symmetric injective) OWPFs, such as multi-collision-resistant hash functions, in mind.

3.3 From OWPFs to Injective OWPFs

Our symmetric OWPF-based constructions most naturally work with (statistically) injective symmetric OWPFs, but an arbitrary OWPF family may be far from injective. To handle this issue, we present a modular transformation which converts, with some security loss, any symmetric OWPF family into a (statistically) injective symmetric OWPF family. In the rest of the paper, we will often assume that our symmetric OWPF families are statistically injective, which can be guaranteed using this transformation.

In addition, we provide a second transformation which converts arbitrary OWPF families into (statistially) injective OWPF families with the property that one-way power families (Definition 3.4) are mapped to one-way power families under this transformation. The security loss in the "one-way power family" case matches the security loss in the symmetric case, while the security loss for general OWPFs is quantitatively worse (for reasons that will become clear). This transformation allows for additional constructions from general OWPFs (and one-way power families), both with and without obfuscation.

We begin with the symmetric case. Let $\mathcal{F} = \{\{(f_I : \{0,1\}^\lambda \to \{0,1\}^*)^k\}_{I \in \mathcal{I}_\lambda}\}_{\lambda \in \mathbb{N}}$ be a family of symmetric OWPFs. We consider the following family $\mathcal{F}'$ of OWPFs with input domain $\{0,1\}^n$. We show that with an appropriate choice of n, it is a *statistically injective* k-OWPF.

Construction 3.2. *Given a family of OWPFs $\mathcal{F}$ and a function $\lambda = \mathrm{poly}(n)$, define the OWPF family $\mathcal{F}'$ as follows. Let $\mathcal{H}_n : \{0,1\}^n \to \{0,1\}^\lambda$ be a pairwise independent hash family.*

$\mathcal{F}'.\mathsf{Gen}$: *On input* 1^n *sample* $H \leftarrow \mathcal{H}_n$*, sample* $I \leftarrow \mathcal{I}$*, and output* (H, I)*.*

$\mathcal{F}'.\mathsf{Samp}$: *On input* (H, I)*, output a uniformly random* $W \leftarrow \{0,1\}^n$*.*

$\mathcal{F}'.\mathsf{Eval}$: *On input* $\big((H, I), W\big)$*, output* $f_I(H(W))$*.*

We use the notation $f'_{I,H}$ as shorthand for a member of the family $\mathcal{F}'$. We first describe the parameter settings in which $\mathcal{F}'$ is statistically injective. Let Inj denote the event (over the randomness of $\mathcal{F}'.\mathsf{Gen}$) that the function $f'_{I,H}$ is injective.

Claim 3.2.1. *Suppose that* $\mathcal{F}$ *is a family of* δ*-secure* k*-OWPFs. Then, the probability of* $\neg$Inj *is at most* $2^{2n} \cdot \delta(\lambda)^{\frac{1}{k}}$*.*

Proof. Let N denote the random variable equal to the number of distinct pairs (w_1, w_2) for which $f_I(H(w_1)) = f_I(H(w_2))$. Then we have $\Pr[\neg\text{Inj}] = \Pr[N \geq 1]$, which by Markov's inequality is at most $\mathbb{E}[N]$.

Let $C(w_1, w_2)$ denote the event that $f_I(H(w_1)) = f_I(H(w_2))$, and let $1_{C(w_1,w_2)}$ denote the corresponding indicator random variable, so that $N = \sum_{w_1 \neq w_2} 1_{C(w_1,w_2)}$. For every $w_1 \neq w_2$ and every i, the pairwise independence of $\mathcal{H}_n$ implies that

$$\mathbb{E}[1_{C(w_1,w_2)} | I = i] = \Pr_{x_1, x_2 \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^\lambda} [f_i(x_1) = f_i(x_2)] \,.$$

We call the latter probability the collision probability of i, and denote it by $\mathsf{CP}(i)$. By the above, $\mathbb{E}[N | I = i] = \binom{2^n}{2} \cdot \mathsf{CP}(i)$.

In order for the trivial attack (guess $x_1, x_2, \ldots, x_k$ uniformly at random) to not violate the δ-security of $\mathcal{F}$ as a k-OWPF, it must be that

$$\mathbb{E}\left[\mathsf{CP}(I)^k\right] \leq \delta(\lambda). \tag{1}$$

Thus, we have

$$\begin{aligned}
\Pr[\neg\text{Inj}] &\leq \mathbb{E}[N] \\
&= \mathbb{E}\big[\mathbb{E}[N|I]\big] \\
&= \mathbb{E}\left[\binom{2^n}{2} \cdot \mathsf{CP}(I)\right] \\
&= \binom{2^n}{2} \cdot \mathbb{E}[\mathsf{CP}(I)] \\
&\leq 2^{2n} \cdot \delta(\lambda)^{\frac{1}{k}},
\end{aligned}$$

where the last inequality follows from Jensen's inequality together with Eq. (1). □

Having analyzed the injectivity of $\mathcal{F}'$, we now argue about its security.

Proposition 3.3. *If* $\mathcal{F}$ *is a family of* $\big(s(\lambda) + \text{poly}(\lambda), \delta(\lambda)\big)$*-secure* k*-OWPFs, then for any non-constant* $\lambda = \text{poly}(n)$*, it holds that* $\mathcal{F}'$ *is an* $\big(s(\lambda), \delta'(n)\big)$*-secure family of* k*-OWPFs, where* $\delta'(n)$ *is the maximum of:*

- $\delta(\lambda) \cdot \left(\frac{2^{-n}}{2^{-\lambda}}\right)^k$ *and*
- $2^{-\lambda} \cdot 2^{2n}$.

Given the bounds proved in Proposition 3.3 and Claim 3.2.1, we now consider the special case $\delta(\lambda) = 2^{-\theta k n}$ for intuition. In one reasonable setting of parameters, we can choose

$$\lambda(n) = \frac{k+2}{(1-\theta)k+\theta} n,$$

which yields a OWPF family with security and non-injectivity probability both bounded by

$$\delta'(n) = 2^{-\frac{\theta(k+2)}{(1-\theta)k+\theta} n}.$$

As an example, this yields a $2^{-n} \cdot \mathrm{negl}(n)$-secure injective symmetric k-OWPF (which is sufficient for k-multi collision-resistant hash functions when $k = o(\frac{n}{\log(n)})$ for any $\theta > \frac{3k}{4k-1}$. This implies a construction of collision-resistant hash functions from $2^{\frac{2n}{7}-2n}$-secure symmetric 2-OWPFs.[6]

Proof of Proposition 3.3. Let $P^{(n)}$ denote the distribution of $(H, \mathbf{X})$ in the experiment defined by independently sampling $H \leftarrow \mathcal{H}_n$ and $\mathbf{W} \leftarrow (\{0,1\}^n)^k$, and then defining $X_1 = H(W_1), \ldots, X_k = H(W_k)$. Specifically, we have

$$P^{(n)}(h, \mathbf{x}) = \Pr_{H \leftarrow \mathcal{H}_n}[H = h] \cdot \frac{\prod_{i=1}^k \left|\{w : h(w) = x_i\}\right|}{2^{kn}}. \tag{2}$$

Let $Q^{(n)}$ denote the distribution of $(H, \mathbf{X})$ in the experiment defined by independently sampling $H \leftarrow \mathcal{H}_n$ and $\mathbf{X} \leftarrow (\{0,1\}^\lambda)^k$. Specifically, we have

$$Q^{(n)}(h, \mathbf{x}) = \Pr_{H \leftarrow \mathcal{H}_n}[H = h] \cdot \Pr_{\mathbf{X} \leftarrow (\{0,1\}^\lambda)^k}[\mathbf{X} = \mathbf{x}]. \tag{3}$$

We first note that if h is an injective function, then $P^{(n)}(h, \mathbf{x}) \leq 2^{k\lambda} 2^{-kn} Q^{(n)}(h, \mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^\lambda$.

Now, to prove Proposition 3.3, consider the event $\textsc{Win}_n$ that consists of the outcomes $(I, h, \mathbf{x})$ for which $\mathcal{A}_n(I, h, f_I(x_1), \ldots, f_I(x_k))$ outputs $(w_1, \ldots, w_k)$ such that for each $i \in [k]$, $f_I(h(w_i)) = f_I(x_i)$. Now suppose that $\mathcal{A}$ wins the k-inversion game for $\mathcal{F}'$ with probability greater than $2\delta'$; this exactly means that $P(\textsc{Win}_n) \geq 2\delta'$. Then, consider the algorithm $\mathcal{B}_n$ that on input $Y_1, \ldots, Y_k$ samples $H \leftarrow \mathcal{H}_n$, computes $(W_1, \ldots, W_k) \leftarrow \mathcal{A}_n(H, Y_1, \ldots, Y_k)$, and outputs $(H(W_1), \ldots, H(W_k))$. The probability of $\mathcal{B}_n$ winning the k-inversion game for f (on security parameter $\lambda(n)$) is just $Q^{(n)}(\textsc{Win}_n)$. However, we now note that

$$\begin{aligned}
Q^{(n)}(\textsc{Win}_n) &\geq Q^{(n)}(\textsc{Win}_n \wedge h \text{ injective}) \\
&\geq 2^{k(n-\lambda)} P^{(n)}(\textsc{Win}_n \wedge h \text{ injective}) \\
&\geq 2^{k(n-\lambda)} \left(P^{(n)}(\textsc{Win}_n) - \Pr_{H \leftarrow \mathcal{H}_n}[H \text{ not injective}] \right) \\
&\geq 2^{k(n-\lambda)} (2\delta' - 2^{-\lambda+2n}) \\
&\geq 2^{k(n-\lambda)} \delta' \geq \delta.
\end{aligned}$$

[6] For the specific application of polynomially secure (M)CRHFs, one can tweak parameters differently and obtain a construction from $2^{\frac{-2k}{3k-1} kn}$-secure symmetric k-OWPFs. This is because it suffices to have non-injectivity probability $\mathrm{negl}(n)$ for the later construction to work.

This contradicts the security of $\mathcal{F}$, so we have proved Proposition 3.3. □

Having handled the symmetric case, we now turn to our second transformation.

Construction 3.4. *Given a family of OWPFs $\mathcal{F}$ and a function $\lambda = \mathrm{poly}(n)$, define the OWPF family $\mathcal{F}'$ as follows. Let $\mathcal{H}_n : \{0,1\}^n \to \{0,1\}^\lambda$ be a pairwise independent hash family.*

$\mathcal{F}'.\mathsf{Gen}$**:** *On input 1^n sample $H_1, \ldots H_k \leftarrow \mathcal{H}_n$ independently at random, sample $I \leftarrow \mathcal{I}$, and output $(H_1, \ldots, H_k, I)$.*

$\mathcal{F}'.\mathsf{Samp}$**:** *On input $(j, (H_1, \ldots, H_k, I))$, output a uniformly random $W \leftarrow \{0,1\}^n$.*

$\mathcal{F}'.\mathsf{Eval}$**:** *On input $(j, (H_1, \ldots, H_k, I), W)$, output $f_{j,I}(H_j(W))$.*

Note that if $\mathcal{F}$ is a one-way power family, then so is $\mathcal{F}'$. We now argue about the security of $\mathcal{F}'$, with an argument that works for any OWPF family.

Proposition 3.5. *If $\mathcal{F}$ is a family of $(s(\lambda) + \mathrm{poly}(\lambda), \delta(\lambda))$-secure k-OWPFs, then for any non-constant $\lambda = \mathrm{poly}(n)$, it holds that $\mathcal{F}'$ is an $(s(\lambda), \delta'(n))$-secure family of k-OWPFs, where $\delta'(n)$ is the maximum of:*

- $\delta(\lambda) \cdot \left(\frac{2^{-n}}{2^{-\lambda}}\right)^k$ *and*
- $k \cdot 2^{-\lambda} \cdot 2^{2n}$.

Proof. This follows by an argument almost identical to that of Proposition 3.3.

Let $P^{(n)}$ denote the distribution of $(H_1, \ldots, H_k, \mathbf{X})$ in the experiment defined by independently sampling $H_1, \ldots, H_k \leftarrow \mathcal{H}_n$ and $\mathbf{W} \leftarrow (\{0,1\}^n)^k$, and then defining $X_1 = H_1(W_1), \ldots, X_k = H_k(W_k)$. Specifically, we have

$$P^{(n)}(h_1, \ldots, h_k, \mathbf{x}) = \prod_{i=1}^{k} \Pr_{H_i \leftarrow \mathcal{H}_n}[H_i = h_i] \cdot \frac{\prod_{i=1}^{k} \left|\{w : h(w) = x_i\}\right|}{2^{kn}}. \tag{4}$$

Let $Q^{(n)}$ denote the distribution of $(H_1, \ldots, H_k, \mathbf{X})$ in the experiment defined by independently sampling $H_1, \ldots, H_k \leftarrow \mathcal{H}_n$ and $\mathbf{X} \leftarrow (\{0,1\}^\lambda)^k$. Specifically, we have

$$Q^{(n)}(h_1, \ldots, h_k, \mathbf{x}) = \prod_{i=1}^{k} \Pr_{H_i \leftarrow \mathcal{H}_n}[H_i = h_i] \cdot \Pr_{\mathbf{X} \leftarrow (\{0,1\}^\lambda)^k}[\mathbf{X} = \mathbf{x}]. \tag{5}$$

We first note that if $h_1, \ldots, h_k$ are all injective functions, then $P^{(n)}(h_1, \ldots, h_k, \mathbf{x}) \leq 2^{k\lambda} 2^{-kn} \cdot Q^{(n)}(h_1, \ldots, h_k, \mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^\lambda$. We also note that by a union bound, the k hash functions $H_1, \ldots, H_k$ are all injective with probability at least $1 - k \cdot 2^{2n} 2^{-\lambda}$. Thus, the security of $\mathcal{F}'$ follows from the security of $\mathcal{F}$ by an identical reduction as in Proposition 3.3. □

Moreover, when $\mathcal{F} = \mathcal{G}^k$ is a one-way power family, then $\mathcal{F}'$ is also statistically injective with (essentially) the same parameters as in Claim 3.2.1. For each j, let $\textsc{Inj}_j$ denote the event that f'_{I_j, H_j} is injective, and let $\textsc{Inj} = \bigcup_j \textsc{Inj}_j$.

Claim 3.5.1. *Suppose that $\mathcal{F} = \mathcal{G}^k$ is a δ-secure one-way power family. Then,*

$$\Pr[\neg \textsc{Inj}] \leq k \cdot 2^{2n} \cdot \delta(\lambda)^{\frac{1}{k}}.$$

Proof. By symmetry, $\Pr[\text{INJ}_j]$ is independent of j, and moreover the INJ_j are independent events. Therefore, we have

$$\Pr[\text{INJ}] \leq \sum_j \Pr[\text{INJ}_j] = k \prod_j \Pr[\text{INJ}_j]^{\frac{1}{k}} = k \Pr[\text{INJ}_1 \wedge \ldots \wedge \text{INJ}_k]^{\frac{1}{k}}.$$

But $\Pr[\text{INJ}_1 \wedge \ldots \wedge \text{INJ}_k]$ is at most $2^{2kn} \cdot \delta$ by the same reasoning as in Claim 3.2.1. Namely, $\Pr[\text{INJ}_1 \wedge \ldots \wedge \text{INJ}_k]$ is at most $2^{2kn} \cdot \mathsf{CP}_{[1:k]}$, where $\mathsf{CP}_{[1:k]}$ denotes the probability that a uniformly random k-tuple of pairs $((x_{1,1}, x_{1,2}), \ldots, (x_{k,1}, x_{k,2})) \in (\{0,1\}^\lambda)^{2k}$ satisfies $f_{I_j}(x_{j,1}) = f_{I_j}(x_{j,2})$ for every j; this follows from the pairwise independence of $\mathcal{H}_n$ and the fact that $H_1, \ldots, H_k$ are sampled independently.

Moreover, $\mathsf{CP}_{[1:k]}$ is at most $\delta(\lambda)$, as an adversary that on input $(I_1, \ldots, I_k, y_1, \ldots, y_k)$ guesses $x_1, \ldots, x_k$ uniformly at random succeeds in batch inverting $\mathcal{F}$ with probability at most δ, but succeeds with probability at least $\mathsf{CP}_{[1:k]}$. This completes the proof of Claim 3.5.1. □

Thus, we have a transformation from one-way power families to statistically injective one-way power families with essentially the same security loss as in the case of symmetric k-OWPFs.

On the other hand, for general OWPFs, we can only prove the following weaker claim about injectivity.

Claim 3.5.2. *Suppose that $\mathcal{F}$ is a δ-secure k-OWPF family. Then,* $\Pr[\neg\text{INJ}] \leq k \cdot 2^{2n} \cdot \delta \cdot 2^{(k-1)\lambda}$.

The parameters in this claim are tight for the following reason: suppose that $\mathcal{G}$ is a perfectly secure $(k-1)$-OWPF family and $\mathcal{F}$ is defined so that a member of $\mathcal{F}$ is a member of $\mathcal{G}$ combined with a constant function (as the kth function f_k). Then, no k-tuple of functions in $\mathcal{F}'$ consists of k injective functions.

Proof of Claim 3.5.2. We claim that for any fixed j, $\Pr[\neg\text{INJ}_j] \leq 2^{2n} \cdot \delta \cdot 2^{(k-1)\lambda}$; the desired result then follows from a union bound.

The fact that $\Pr[\neg\text{INJ}_j] \leq 2^{2n} \cdot \delta \cdot 2^{(k-1)n}$ follows by a similar argument to that of Claim 3.2.1. Namely, $\Pr[\neg\text{INJ}_j]$ is at most $2^{2n} \cdot \mathsf{CP}_j$, where CP_j denotes the probability that a random pair $(x_1, x_2) \in (\{0,1\}^\lambda)^2$ satisfies $f'_{j,I}(x_1) = f'_{j,I}(x_2)$; this follows from the pairwise independence of $\mathcal{H}_n$. But this probability in turn is at most $\delta \cdot 2^{(k-1)\lambda}$, as an adversary that on input $(I, y_1, \ldots, y_k)$ guesses $x_1, \ldots, x_k$ uniformly at random succeeds in batch inverting $\mathcal{F}$ with probability at most δ, but succeeds with probability at least $2^{-(k-1)\lambda} \cdot \mathsf{CP}_j$.

This completes the proof of Claim 3.5.2. □

3.4 From OWPFs to Symmetric OWPFs

In this section, we will construct families of symmetric OWPFs in two different ways: one construction is from general OWPF families, while the other is from one-way power families (Definition 3.4). The two reductions will have different security losses.

As usual, we assume that all functions in a fixed OWPF family have input domain $\{0,1\}^n$.

Theorem 3.6. *Let $\mathcal{F} = \{(f_{1,I}, \ldots, f_{k,I})\}_{I \in \mathcal{I}}$ be a $(s + \text{poly}(n), \delta)$-secure family of k-OWPFs with domain $\{0,1\}^n$. Then, for any L, the function family $\mathcal{F}' = \{(f'_I, f'_I, \ldots, f'_I)\}_{I \in \mathcal{I}}$ is a (s, δ')-secure family of symmetric L-OWPFs, where $f'_I(x||j) = j||f_{j,I}(x)$ and*

$$\delta' = \delta + k(1 - \frac{1}{k})^L \min(\delta \cdot 2^{(k-1)n}, 1).$$

Remark 3.2. *Note that if all $f_{j,I}$ are injective with probability $1-\eta$, then a random element of the family $\mathcal{F}'$ is injective with probability at least $1-\eta$.*

Proof. Suppose that some size s adversary $\mathcal{A}(y'_1,\ldots,y'_L)$ wins the OWPF security game for $\mathcal{F}'$ with probability ϵ, where $y'_i = j_i||f_{j_i,I}(x_i)$ for each i. Let WIN denote the event that $\mathcal{A}$ produces L valid inverses (i.e. it wins the security game), and let DISTINCT be the event that $\{j_1,\ldots,j_L\}$ contains at least k distinct elements. We prove two claims about the behavior of $\mathcal{A}$.

Claim 3.6.1. $\Pr[\text{WIN} \wedge \text{DISTINCT}] \leq \delta$.

Proof. This follows from the $(s+\text{poly}(n),\delta)$-security of $\mathcal{F}$. Namely, a k-OWPF adversary $\mathcal{A}'$ given $(\mathcal{I}, y_1,\ldots,y_k)$ can select $j_1,\ldots,j_L \xleftarrow{\$} [k]$ at random and prepare a L-OWPF challenge for $\mathcal{A}$ containing each y_i in a location t with $j_t = i$ (not including the challenge y_i if there is no such location). This perfectly simulates the OWPF security game for $\mathcal{A}$, and in the event that WIN $\wedge$ DISTINCT occurs, $\mathcal{A}'$ obtains inverses to all k of its challenges. Thus, we conclude the claim by the security of $\mathcal{F}$. □

Claim 3.6.2. $\Pr[\text{WIN} \mid \neg\text{DISTINCT}] \leq \delta \cdot 2^{(k-1)n}$.

Proof. This also follows from the $(s+\text{poly}(n),\delta)$-security of $\mathcal{F}$. Namely, a k-OWPF adversary $\mathcal{A}'$ given $(\mathcal{I}, y_1,\ldots,y_k)$ can select $j_1,\ldots,j_L \xleftarrow{\$} [k]$ subject to the event $\neg$DISTINCT (this can be done efficiently) and prepare a L-OWPF challenge for $\mathcal{A}$ containing y_{j_1} in location 1. Whenever $\mathcal{A}$ successfully inverts its first challenge, $\mathcal{A}'$ can guess its other $k-1$ challenges uniformly at random and win with probability $2^{-(k-1)n}$. Thus, we conclude the claim by the security of $\mathcal{F}$. □

Finally, we note the combinatorial fact that $\Pr[\text{DISTINCT}] \leq k(1-\frac{1}{k})^L$. Combining the two claims and this fact, we obtain the statement of Theorem 3.6. □

Remark 3.3. *Setting $L \approx k\log(\frac{1}{\delta}) < k^2 n$, we see that the family $\mathcal{F}'$ defined above is a $(s, \delta(1+o(1)))$-secure family of symmetric L-OWPFs.*

Remark 3.4. *If we instead set $k=2$, $\log^{1.1}(n) < L < \frac{n}{\log^{1.1}(n)}$, and $\delta = 2^{-2n+\frac{L}{2}}$, we obtain a construction of $2^{-n-L\log(L)}\cdot\text{negl}(n)$-secure symmetric L-OWPFs from suitably strong (asymmetric) 2-OWPFs. This is sufficient for L-multi-collision resistant hash functions (MCRHFs) if the original family $\mathcal{F}$ is also statistically injective. While this requires almost perfect security from the original OWPF family, we see this as a proof of concept that the most general notion of OWPF can be used without obfuscation to build more expressive primitives, such as MCRHFs.*

We now give a construction of symmetric OWPFs from one-way power families that has a milder security loss than the construction of Theorem 3.6; in the event that the one-way power family is *public coin*, the security loss can be improved even further.

Theorem 3.7. *Let $\mathcal{F}^k = \{(f_{I_1},\ldots,f_{I_k})\}_{(I_1,\ldots,I_k)\in\mathcal{I}^k}$ be a public coin $(ks+\text{poly}(n),\delta)$-secure one way k-power family with domain $\{0,1\}^n$. Moreover, for any $N = 2^{\nu(n)}$, and suppose that $\mathcal{H}$ is a family of programmable k-wise independent hash functions from $[N]\to\mathcal{I}$, where $\mathcal{I}$ is the key space*

for $\mathcal{F}$.[7] *Then, for any L, the function family $\mathcal{F}'^L = \{(f'_h, f'_h, \ldots, f'_h)\}_{h\in\mathcal{H}}$ is a (s, δ')-secure family of L-OWPFs with domain $\{0,1\}^{n+\nu(n)}$, where*

$$f'_h(x||\rho) = \rho||f_{h(\rho)}(x)$$

and

$$\delta' = \delta + (k-1)\cdot \max_{1\le d\le k-1}\left[\frac{d^d}{d!}\left(\frac{d}{N}\right)^{L-d}\delta^{\frac{1}{\lceil k/d\rceil}}\right].$$

In the special case that $N = \mathrm{poly}(n,k)$ and a member of the "hash family" $\mathcal{H}$ consists of N independently sampled $I_1, \ldots, I_N \in \mathcal{I}$, we obtain the same conclusion when $\mathcal{F}$ is not public coin.

Remark 3.5. *Note that if all members of the family $\mathcal{F}$ are injective, then f'_h is injective for every choice of hash function h.*

Proof. Suppose that some size s adversary $\mathcal{A}(h, y'_1, \ldots, y'_L)$ wins the OWPF security game for $\mathcal{F}'^k$ with probability ϵ, where $y'_i = \rho_i||f_{I_i}(x_i)$ and $I_i = \mathcal{F}.\mathsf{Samp}(\rho_i)$ for each i. Let Win denote the event that $\mathcal{A}$ produces L valid inverses (i.e. it wins the security game), and let d-Distinct be the event that $\{\rho_1, \ldots, \rho_L\}$ contains exactly d distinct elements. We prove the following claim about the behavior of $\mathcal{A}$.

Claim 3.7.1. $\Pr[\textsc{Win} \mid d\text{-}\textsc{Distinct}] \le \delta^{\frac{1}{\lceil k/d\rceil}}$.

Proof. This follows from a two-part argument. First, we note that the family $\mathcal{F}^d$ is a public coin $(s, \delta^{\frac{1}{\lceil k/d\rceil}})$-secure one-way d-power family. This is because any algorithm breaking $\mathcal{F}^d$ could be used $\lceil k/d\rceil$ times independently to break $\mathcal{F}^k$.

Thus, we prove the claim by reducing from the one-wayness of $\mathcal{F}^d$. In particular, an adversary $\mathcal{A}'$ given d independently drawn indices $I_1, \ldots, I_d$ and values $y_i = f_{I_i}(x_i)$ to invert could use $\mathcal{A}$ to break $\mathcal{F}^d$ in the following way.

- First, sample L uniformly random values $\rho_1, \ldots, \rho_L \leftarrow [N]$ such that there are exactly d distinct ρ_i. Call these values $\rho^*_1, \ldots, \rho^*_d$.
- Sample a hash function $h \leftarrow \mathcal{H}.\mathsf{CondGen}(\rho^*, \mathbf{I})$, i.e., a hash function subject to the constraints that $h(\rho^*_i) = I_i$. Here, we think of the indices I_i as public coins so that this sampling is possible.
- Run $\mathcal{A}(h, y'_1, \ldots, y'_L)$, where $y'_i = \rho_i||y_i$.

By the conditional sampling property of $\mathcal{H}$ (and k-wise independence), the input distribution to $\mathcal{A}$ in this experiment is exactly the correct input distribution (a random input subject to the constraint d-Distinct), so by the $\delta^{\frac{1}{\lceil k/d\rceil}}$-hardness of $\mathcal{F}^d$, we conclude the claim. □

Finally, we note that by a counting argument,

$$\Pr[d\text{-}\textsc{Distinct}] = \binom{N}{d}\left(\frac{d}{N}\right)^L \le \frac{d^d}{d!}\left(\frac{d}{N}\right)^{L-d}.$$

Thus, we conclude Theorem 3.7 by a standard probability calculation.

□

[7] This is possible when either (1) $\mathcal{F}$ is public coin, or (2) $N = \mathrm{poly}(n,k)$, in which case sampling from $\mathcal{H}$ consists of sampling N independent keys from $\mathcal{I}$.

Corollary 3.8. *Consider the case when $\mathcal{F}$ is public coin, $k = 2$ and $L = 3$, and set $\nu(n) = \frac{1}{4}\log(\frac{1}{\delta})$. Then, we have*

$$\delta' = \delta + \frac{1}{N^2}\delta^{\frac{1}{2}} = 2\delta,$$

with a new security parameter of $n' = n + \frac{1}{4}\log(\frac{1}{\delta})$. For example, this yields a $2^{-n'} \cdot \mathrm{negl}(n')$-secure symmetric 3-OWPF family from $2^{-4n/3} \cdot \mathrm{negl}(n)$-secure (public coin) 2-one-way power families (which suffice for 3-MCRHFs if $\mathcal{F}$ is also injective), and a $2^{(-4/3+o(1))n} \cdot \mathrm{negl}(n)$-secure symmetric 3-OWPF family from $2^{(-2+o(1))n}$-secure (public coin) one-way 2-power families.

Corollary 3.9. *Consider the case when $\mathcal{F}$ is public coin, $L = k(1 + \log(n))$ and set $\nu(n) = \frac{n}{\log(n)}$. Then, we have*

$$\delta' < \delta + k^L 2^{-kn},$$

yielding essentially a (s, δ)-secure symmetric L-OWPF family from δ-secure (public coin) one-way k-power families. This reduction suffices for many of the applications in Section 5 if $\mathcal{F}$ is injective.

Corollary 3.10. *Consider the case $N = ek \cdot (nk)^c$; then, setting $L = k + \frac{1}{c\log(kn)}\log(\frac{1}{\delta})$, we obtain $\delta' < (1 + o(1))\delta$. This yields L-MCRHFs from any statistically injective[8], δ-secure one-way k-power families with $\delta < \left(2^{-n-k\log(k)}\right)^{\frac{1}{1-1/c}}$.[9] We therefore also obtain L-MCRHFs from sufficiently secure (not necessarily injective) one-way k-power families by first applying Construction 3.4 and then applying the construction of Theorem 3.7.*

4 Collision Resistance from OWPFs

Having defined and explored the foundations of OWPFs in Section 3, we now turn to *applications* of OWPFs. In this section, we prove our main theorem on collision resistance. As usual, we assume that all OWPFs used have domain $\{0,1\}^n$.

Theorem 4.1. *Suppose that $\mathcal{F}$ is an $(s(n), \delta(n))$-secure symmetric 2-OWPF-family $\mathcal{F}$ that is injective with probability $1 - \eta$. Then, for every $m = m(n)$, the hash family $\mathcal{H} := \mathcal{H}_{\mathcal{F},n,m(n)}$ in Construction 4.2 is (s', δ')-collision-resistant for $s' = s + \mathrm{poly}(n)$ and $\delta' = \eta + \delta \cdot 2^{2n-m}$.*

Construction 4.2. *Given input and output lengths n and m, and a symmetric 2-OWPF-family $\mathcal{F} = \{(f_I, f_I)\}_{I\in\mathcal{I}}$ given by algorithms $(\mathcal{F}.\mathsf{Gen}, \mathcal{F}.\mathsf{Eval})$, define the hash family $\mathcal{H} = \mathcal{H}_{\mathcal{F},n,m}$ by $(\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Eval})$ as follows. Let $\ell = \mathrm{poly}(\lambda)$ denote a bound on the output length of f_I for I in the support of $\mathcal{F}.\mathsf{Gen}(1^\lambda)$.*

$\mathcal{H}.\mathsf{Gen}$**:** *On input 1^λ sample $I \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda)$, and sample $H_{\mathsf{out}} : \{0,1\}^\ell \to \{0,1\}^m$ from a programmable pairwise independent hash family. Output (I, H_{out}) as the hash function description.*

$\mathcal{H}.\mathsf{Eval}$**:** *On input $((I, H_{\mathsf{out}}), x)$, output $H_{\mathsf{out}}(f_I(x))$.*

[8] If $\mathcal{F}$ is statistically injective, then $\mathcal{F}'$ is statistically injective because with overwhelming probability, all N of the sampled function keys $I_1, \ldots, I_n$ will be injective.

[9] This follows from the inequality $\delta < 2^{-(n+\nu(n))}2^{-L\log(L)}$.

Corollary 4.3 (Follows from Theorem 4.1 and Section 3.3)**.** *If there exists a* $(\mathrm{poly}(n), 2^{-n}\cdot\mathrm{negl}(n))$*-secure symmetric* 2*-OWPF family that is injective with probability* $1-\mathrm{negl}(n)$*, then collision resistant hash families exist. Also, if there exists a* $(\mathrm{poly}(n), 2^{-1.6n}\cdot\mathrm{negl}(n))$*-secure symmetric* 2*-OWPF family (with no injectivity hypothesis), then there exist collision-resistant hash families. Finally, if symmetric* 2*-OWPFs with* nearly optimal security *(i.e., security* $(s, 2^{-2n(1+o(1))}s^c)$*) exist, then CRHFs with nearly optimal security also exist.*

Informally, we will define $H_{i,h_{\mathsf{out}}}(x) := h_{\mathsf{out}}(f_i(x))$ for $i \in \mathcal{I}$ and $h_{\mathsf{out}} \in \mathcal{H}_{\mathsf{out}}$ and refer to Construction 4.2 as the "outer hash construction."

For any 2-OWPF family $\mathcal{F}$ and associated outer hash construction $\mathcal{H} = \mathcal{H}_{\mathcal{F},m}$, we first prove that it is hard to find a *certain type* of "outer" collisions in $\mathcal{H}$.

Definition 4.1 (Outer and Inner Collisions)**.** *Let* $\mathcal{F}$ *be a 2-OWPF family and* $\mathcal{H} = \mathcal{H}_{\mathcal{F},m}$ *be an associated outer hash construction. We say that* $(x_0, x_1) \in \{0,1\}^n$ *is an* outer collision *with respect to* (i, h_{out}) *if* $H_{i,h_{\mathsf{out}}}(x_0) = H_{i,h_{\mathsf{out}}}(x_1)$ *but* $f_i(x_0) \neq f_i(x_1)$*. We say that* (x_0, x_1) *is an* inner collision *if* $f_i(x_0) = f_i(x_1)$*.*

Our result is as follows.

Theorem 4.4 (Outer Hash Lemma)**.** *For any polynomial* $m(n)$*, there exists*[10] *a polynomial* $p(n)$ *such that for any* $(s(n)+p(n), \delta(n))$*-secure family* $\mathcal{F}$ *of symmetric 2-OWPFs, it is* $(s(n), \delta(n)\cdot 2^{2n-m(n)})$*-hard to find any* outer collision *in* $\mathcal{H}_{\mathcal{F},m}$*, given* $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$*.*

Proof. Suppose for the sake of contradiction that there is an adversary $\mathcal{A} = \{\mathcal{A}_n\}$ that violates the $(s(n), \delta(n)\cdot 2^{2n-m(n)})$-hardness of finding outer collisions for $\mathcal{H}_{f,m}$. That is, (1) the size of $\mathcal{A}_n$ is at most $s(n)$, and (2) for infinitely many n, the probability that (X_0, X_1) is an outer collision with respect to (I, H_{out}) is some $\epsilon(n) > \delta(n)\cdot 2^{2n-m(n)}$ in the probability space defined by sampling $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$ and $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

We let $\mathrm{Expt}_n^{(0)}$ and $\Pr_n^{(0)}$ respectively denote the experiment described above and the probability measure that it induces. In $\mathrm{Expt}^{(0)}$, let $\textsc{Win}$ denote the event that (X_0, X_1) is an outer collision with respect to (I, H_{out}). We now define a sequence of related probability experiments $\left\{\mathrm{Expt}_n^{(j)}\right\}_{j\in\{1,2,3\}}$, and let $\{\Pr_n^{(j)}\}$ denote the probability measures that they induce.

- Let $\mathrm{Expt}_n^{(1)}$ denote the following modification of $\mathrm{Expt}_n^{(0)}$:
  1. Sample $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$
  2. Sample $X_0^*, X_1^* \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^n$.
  3. Compute $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

  In $\mathrm{Expt}_n^{(1)}$, let $\textsc{Win}$ denote the event that (X_0, X_1) is an outer collision with respect to (I, H_{out}). It holds that

$$\Pr_n^{(1)}\left[\textsc{Win} \wedge ((X_0, X_1) = (X_0^*, X_1^*))\right] = \Pr_n^{(0)}\left[\textsc{Win}\right]\cdot 2^{-2n} \geq \frac{\epsilon(n)}{2^{2n}}.$$

- Let $\mathrm{Expt}_n^{(2)}$ denote the following further modification.

[10] In fact, $p(n) = \mathrm{poly}(n, m(n))$ for some polynomial poly that depends only on the programmable pairwise hash family $\mathcal{H}_{\mathsf{out}}$.

1. Sample (I, X_0^*, X_1^*) as in $\mathrm{Expt}_n^{(1)}$. If $f_I(X_0^*) = f_I(X_1^*)$, abort.
2. Sample $Z_0^*, Z_1^* \xleftarrow{\text{i.i.d.}} \{0,1\}^{m(n)}$.
3. Sample $H_{\mathsf{out}} \leftarrow \mathcal{H}.\mathsf{CondGen}(\mathbf{Y}^*, \mathbf{Z}^*)$, where $\mathbf{Y}^* = (f_I(X_0^*), f_I(X_1^*))$ and $\mathbf{Z}^* = (Z_0^*, Z_1^*)$.
4. Compute $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

In $\mathrm{Expt}_n^{(2)}$, let $\textsc{Win}$ denote the event that (1) $f_I(X_0^*) \neq f_I(X_1^*)$ (so that the experiment proceeds to completion) and (2) (X_0, X_1) is an outer collision with respect to (I, H_{out}). Then, it holds that

$$\Pr_n^{(2)}\left[\textsc{Win} \wedge ((X_0, X_1) = (X_0^*, X_1^*))\right] = \Pr_n^{(1)}\left[\textsc{Win} \wedge ((X_0, X_1) = (X_0^*, X_1^*))\right] \geq \frac{\epsilon}{2^{2n}}.$$

by the pairwise independence and programmability of $\mathcal{H}_{\mathsf{out}}$ (and the fact that $\mathbf{Z}^*$ was chosen uniformly at random).

- Let $\mathrm{Expt}^{(3)}$ denote the following further modification.

1. Sample (I, X_0^*, X_1^*) as in $\mathrm{Expt}^{(1)}$. If $f_I(X_0^*) = f_I(X_1^*)$, abort.
2. Sample $Z^* \leftarrow \{0,1\}^{m(n)}$ and define $Z_0^* = Z_1^* = Z^*$.
3. Sample $H_{\mathsf{out}} \leftarrow \mathcal{H}.\mathsf{CondGen}(\mathbf{Y}^*, \mathbf{Z}^*)$, where $\mathbf{Y}^* = (f_I(X_0^*), f_I(X_1^*))$ and $\mathbf{Z}^* = (Z_0^*, Z_1^*)$.
4. Compute $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

In $\mathrm{Expt}_n^{(3)}$, let the event $\textsc{Win}$ be defined as in $\mathrm{Expt}_n^{(2)}$. Then

$$\begin{aligned}
\Pr_n^{(3)}\left[(X_0, X_1) = (X_0^*, X_1^*)\right] &\geq \Pr_n^{(3)}\left[\textsc{Win} \wedge (X_0, X_1) = (X_0^*, X_1^*)\right] \\
&= \Pr_n^{(2)}\left[\textsc{Win} \wedge (X_0, X_1) = (X_0^*, X_1^*) | Z_0^* = Z_1^*\right] \\
&= \frac{\Pr_n^{(2)}\left[\textsc{Win} \wedge (X_0, X_1) = (X_0^*, X_1^*)\right]}{\Pr_n^{(2)}\left[Z_0^* = Z_1^*\right]} \qquad (6) \\
&\geq \frac{\epsilon \cdot 2^{-2n}}{2^{-m(n)}} = \epsilon \cdot 2^{m(n)-2n}. \qquad (7)
\end{aligned}$$

where Eq. (6) follows because the event "$\textsc{Win} \wedge (X_0, X_1) = (X_0^*, X_1^*)$" occurs *only* when $Z_0^* = Z_1^*$.

We now deduce the existence of an $(s(n) + \mathrm{poly}(n), \epsilon(n) \cdot 2^{m(n)-2n})$-attack on the 2-OWPF security of $\mathcal{F}$. The attack is given by the following algorithm $\mathcal{B} = \{\mathcal{B}_n\}$. On input (I, Y_0^*, Y_1^*), $\mathcal{B}_n$ does the following:

1. Sample $Z^* \leftarrow \{0,1\}^{m(n)}$
2. Sample $H_{\mathsf{out}} \leftarrow \mathcal{H}_{\mathsf{out}}.\mathsf{CondGen}(\mathbf{Y}^*, (Z^*, Z^*))$
3. Compute and output $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

Suppose that as in the 2-OWPF security game, $\mathcal{B}_n$'s input (I, Y_0^*, Y_1^*) is generated by sampling $I \leftarrow \mathcal{F}.\mathsf{Gen}(1^n)$; $X_0^*, X_1^* \xleftarrow{\text{i.i.d.}} \{0,1\}^n$; and $Y_b^* = f_I(X_b^*)$ for each $b \in \{0,1\}$. Then all of our named random variables are jointly distributed exactly as in $\mathrm{Expt}_n^{(3)}$. Thus the output (X_0, X_1) of $\mathcal{B}_n$ is equal to (X_0^*, X_1^*) (and in particular $\mathcal{B}_n$ has inverted both Y_0^* and Y_1^*) with probability at least $\epsilon(n) \cdot 2^{m(n)-2n} > \delta(n)$.

This concludes the proof of Theorem 4.4. □

Finally, we give a proof of Theorem 4.1.

Proof of Theorem 4.1. Suppose there is some size s adversary $\mathcal{A}$ that on input (I, H_{out}) outputs $x_1 \neq x_2$ such that $H_{I,H_{\mathsf{out}}}(x_1) = H_{I,H_{\mathsf{out}}}(x_2)$ with probability δ'; that is, $\mathcal{A}$ finds a collision with this probability. Note that with probability $1 - \eta$ over the randomness of $\mathcal{H}.\mathsf{Gen}$, *no inner collisions exist* in $H_{I,H_{\mathsf{out}}}$. Moreover, note that by Theorem 4.4, $\mathcal{A}$ outputs an outer collision with probability at most $\delta \cdot 2^{2n-m}$. We conclude Theorem 4.1 by a union bound. □

4.1 Parameter Settings and Discussion

When we aim for polynomially-secure CRHFs from $\{0,1\}^n \to \{0,1\}^m$, the 2-OWPF assumption required by Theorem 4.1– namely, $2^{m-2n} \cdot \mathrm{negl}(n)$-secure injective symmetric 2-OWPFs – is plausible for any $m = \omega(\log(n))$.

We also obtain "optimally hard" collision resistant hash functions under plausible assumptions (which, for example, are satisfied by our "double discrete logarithm" candidate). The relevant result is sketched in Corollary 4.3, but to be more specific, our hash function is collision-resistant with $2^{-m(1-\epsilon)}$-security assuming the existence of a $2^{-2n+\epsilon m} \cdot \mathrm{negl}(n)$-secure (injective symmetric) 2-OWPF, which is plausible for any $m = \omega(\frac{\log(n)}{\epsilon})$. This yields (for any super-logarithmic output length) a collision resistant hash family with security nearly matching the trivial attack of outputting two uniformly random points x_1, x_2. Moreover, by Section 3.3, the injectivity requirement on the 2-OWPF family can be removed (with slightly more security loss).

In terms of optimality, we recall that by Theorem 1.7 (see Appendix A), the construction of CRHFs from $2^{-n} \cdot \mathrm{negl}(n)$-secure injective symmetric 2-OWPFs cannot be quantitatively improved (with black box techniques); indeed, even one-way permutations with security $\frac{2^{-n}}{\mathrm{negl}(n)}$ do not imply CRHFs in a black-box way. Thus, constructing $2^{-n} \cdot \mathrm{negl}(n)$-secure injective symmetric 2-OWPFs from $2^{-\frac{n}{2}} \cdot \mathrm{negl}(n)$-secure one-way permutations (or even $2^{-.99n}$-secure one-way permutations) is an extremely interesting open question.

As a final note on collision resistance, recall that Corollary 4.3 shows that CRHFs exist as long as sufficiently secure *symmetric* 2-OWPFs exist (without having to assume injectivity), but none of our OWPF transformations currently suffice to build CRHFs from asymmetric OWPFs. We leave the question of whether CRHFs can be constructed from (sufficiently secure) arbitrary 2-OWPFs open.

5 Output Intractability from OWPFs

In this section, we generalize the proof strategy of Section 4 to build correlation intractable hash functions for all k-ary output relations ("k-output intractable hash functions") assuming suitably secure k-OWPF families exist. The hardness that we need depends quantitatively on the *sparsity* of the relation R.

We now define the relevant objects and assumptions for our construction.

Definition 5.1 (Correlation Intractability)**.** *A hash family $\mathcal{H} = \{h_I\}_{I \in \mathcal{I}}$ (as in Definition 2.3) is said to be (s, δ)-multi-input correlation intractable for a class $\mathcal{R}$ of relations if for every $2k$-ary relation $R \in \mathcal{R}$ and every size-$s(\cdot)$ circuit ensemble $\{\mathcal{A}_n\}$,*

$$\Pr_{\substack{I \leftarrow \mathsf{Gen}(1^\lambda) \\ (x_1,\ldots,x_k) \leftarrow \mathcal{A}_\lambda(I)}} [(x_1, \ldots, x_k, h_I(x_1), \ldots, h_I(x_k)) \in R] \leq O(\delta(\lambda)).$$

Additionally, $\mathcal{H}$ is said to be δ-multi-input correlation intractable with respect to $\mathcal{R}$ if $\mathcal{H}$ is (n^c, δ) multi-input correlation intractable for every $c > 0$, and $\mathcal{H}$ is said to be multi-input correlation intractable with respect to $\mathcal{R}$ if $\mathcal{H}$ is $(n^c, m(n)^{-c})$ multi-input correlation intractable for every $c > 0$.

Correlation intractability is a useful and versatile property of random oracles that we would like to guarantee in the standard model. However, even a random oracle $\mathcal{O}$ is not correlation intractable with respect to relations R whose accepting inputs are sufficiently dense. To avoid this problem, we restrict our relations R to be *sparse* as in Definition 5.2 below.

Definition 5.2 (Sparsity)**.** *For any relation $R \subseteq (\{0,1\}^*)^k \times (\{0,1\}^*)^k$, we say that R is $p(\cdot)$-sparse if for any $\mathbf{x} \in (\{0,1\}^*)^k$,*

$$\Pr_{\mathbf{y} \leftarrow (\{0,1\}^m)^k} [(\mathbf{x}, \mathbf{y}) \in R] \leq p(m).$$

When p is a negligible function, we say simply that R is sparse.

Ideally, we would construct a hash family that is correlation intractable for all sparse relations. However, our OWPF-based construction is only able to handle k-ary relations R that depend only on the *outputs* of $\mathcal{H}$ rather than the inputs.

Definition 5.3 (Output Intractability)**.** *We say that a hash family $\mathcal{H}$ is (s, δ)-output intractable for a class $\mathcal{R}$ of relations if $\mathcal{H}$ is (s, δ)-multi-input correlation intractable for $\mathcal{R}$, and every relation in $\mathcal{R}$ (1) requires that $x_1, \ldots, x_k$ are distinct, and (2) is otherwise only a function of the* outputs *y_i of $\mathcal{H}$ (and not the inputs).*

We note that requiring distinct inputs $x_1, \ldots, x_k$ is necessary in order for our notion of sparsity to be applicable; this is because the random variable $(H(x_1), \ldots, H(x_k))_{H \leftarrow \mathcal{H}}$ cannot be a uniformly random k-tuple if $x_i = x_j$ for some $i \neq j$. However, every k-ary relation $R(y_1, \ldots, y_k)$ can be thought of as a union of at most k^k relations to which Definition 5.3 can be applied.

Moreover, we note that in Section 6.3, we are able to construct hash functions that go beyond output intractability, at the cost of introducing indistinguishability obfuscation as an additional assumption.

Finally, we discuss the notion of *samplability* of a relation, which will prove useful in our security proof.

Definition 5.4 (t-Samplability of a relation R)**.** *An output relation $R \subset (\{0,1\}^m)^k$ is samplable in time t if there is a sampling algorithm S such that (1) $S(1^n, 1^k)$ runs in time $t = t(n)$, and (2) for every $\mathbf{y} \in R$,*

$$\Pr[S(1^n, 1^k) = y_i \text{ for all } i] = \Pr_{\mathbf{Y} \leftarrow (\{0,1\}^m)^k} [\mathbf{Y} = \mathbf{y} \mid R(\mathbf{Y}) = 1].$$

In other words, the distribution sampled by $S(1^n, 1^k)$ is the uniform distribution on the set of $\mathbf{y}$ for which $R(\mathbf{y}) = 1$.

We say that R is efficiently samplable if it is samplable in time $\mathrm{poly}(n, k)$.

Remark 5.1. *Any output relation $R \subset (\{0,1\}^m)^k$ is samplable by a* non-uniform *algorithm running in time $t = 2^{2km} \cdot \mathrm{poly}(m)$ by enumerating over all outputs $(y_1, \ldots, y_k)$, computing each $R(y_1, \ldots, y_k)$ (using a circuit of size 2^{km}), and selecting a uniformly random k-tuple out of those satisfying R.*

Let $\mathcal{R}^{\text{out}}_{k,p,t}$ denote the class of k-ary output relations

$$R = \{R_n \subset (\{0,1\}^n)^{k(n)} \times (\{0,1\}^m)^{k(n)}\}$$

that are p-sparse and samplable in time t,[11] and let $\mathcal{R}^{\text{out}}_{k,p} = \bigcup_t \mathcal{R}^{\text{out}}_{k,p,t}$ denote the class of k-ary output relations that are p-sparse. For any $R \in \mathcal{R}^{\text{out}}_{k,p}$, we will abuse notation and think of R as both a relation on $(\{0,1\}^m)^k$ (i.e. the outputs) and a relation on $(\{0,1\}^n)^k \times (\{0,1\}^m)^k$ (the output relation along with the constraint that the inputs x_i are all distinct).

We now state our results on output intractability.

Theorem 5.1. *Suppose that $\mathcal{F} = \{f_I : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{I \in \mathcal{I}}\}$ is a family of symmetric k-OWPFs with security $(s + \text{poly}(n), \delta)$, and suppose further that $\mathcal{F}$ is injective with probability $1 - \eta$. For every $m = m(n)$, let $\mathcal{H} := \mathcal{H}_{\mathcal{F},n,m(n)}$ denote the hash family in Construction 4.2. Then, for every sparsity p, $\mathcal{H}$ is (s, δ')-output intractable for $\mathcal{R}^{\text{out}}_{k,p}$ with $\delta' = \eta + \delta \cdot p \cdot 2^{kn}$.*

Moreover, if a relation $R \in \mathcal{R}^{\text{out}}_{k,p,t}$ is samplable in uniform time t, then there is a uniform *reduction to OWPF security with an additional loss of t time.*

Construction 5.2. *Suppose we are given a symmetric k-OWPF family $\mathcal{F} = \{f_I\}$ with input space $\{0,1\}^n$ and output space $\{0,1\}^{\ell(n)}$ given by algorithms $(\mathcal{F}.\mathsf{Gen}, \mathcal{F}.\mathsf{Eval})$. We define the hash family $\mathcal{H} = \mathcal{H}_{\mathcal{F},n,m}$ by $(\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Eval})$ as follows.*

$\mathcal{H}.\mathsf{Gen}$**:** *On input 1^λ sample $I \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda)$, and sample $H_{\mathsf{out}} : \{0,1\}^\ell \to \{0,1\}^m$ from a programmable k-wise independent hash family $\mathcal{H}_{\mathsf{out}}$. Output (I, H_{out}) as the hash function description.*

$\mathcal{H}.\mathsf{Eval}$**:** *On input $((I, H_{\mathsf{out}}), x)$, output $H_{\mathsf{out}}(\mathcal{F}.\mathsf{Eval}(I, x))$.*

Remark 5.2. *For various parameter settings, Theorem 5.1 can be combined with the reductions of Section 3.3 and Section 3.4 to obtain constructions from certain asymmetric and/or non-injective OWPFs. See Proposition 3.3 and Section 3.4 for some examples.*

Informally, we will define $H_{I,h_{\mathsf{out}}}(x) := h_{\mathsf{out}}(f_I(x))$ for $I \in \mathcal{I}$ and $h_{\mathsf{out}} \in \mathcal{H}_{\mathsf{out}}$ and call Construction 5.2 the "(generalized) outer hash construction."

We will prove Theorem 5.1 by generalizing the outer hash lemma (Theorem 4.4) to the case of general output intractability. That is, for any k-OWPF family $\mathcal{F}$ and associated outer hash construction $\mathcal{H} = \mathcal{H}_{\mathcal{F},n,m}$, and for any output relation $R \in \mathcal{R}^{\text{out}}_{k,p,t}$, we prove that $\mathcal{H}$ is correlation intractable with respect to a *modified* relation $R_{\mathcal{F}}$:

Definition 5.5 (Post-Composed Relation R_f)**.** *For any output relation $R \subset (\{0,1\}^m)^k$ and any function $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$, we define the post-composed relation R_f by*

$$R_f(\mathbf{x}, \mathbf{y}) = 1 \text{ if and only if } R(\mathbf{y}) = 1 \text{ and } f(x_1), \ldots, f(x_k) \text{ are distinct.}$$

In the case of collision resistance, this definition corresponds to the notion of an "outer collision."

Our result is as follows.

[11] We may consider both uniform and non-uniform versions of this definition.

Theorem 5.3 (Generalized Outer Hash Lemma)**.** *Let t' be the runtime of the sampling algorithm $\mathcal{H}_{\mathsf{out}}.\mathsf{CondGen}$. If $\mathcal{F}$ is a $(s+t'+t,\delta)$-secure k-OWPF against uniform adversaries and $R \in \mathcal{R}^{\text{out}}_{k,p,t}$, then $\mathcal{H}$ is $(s, \delta \cdot \frac{2^{kn}}{p})$-correlation intractable with respect to $R_{\mathcal{F}}$ (i.e. the relation R_{f_I} depends on the hash function $(I, \mathcal{H}_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}$). Moreover, the same conclusion holds if $\mathcal{F}$ is a $(s+t',\delta)$-secure k-OWPF with respect to nonuniform adversaries.*

Proof. Suppose that some s-time adversary $\mathcal{A}$, on input $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$, produces with probability ϵ an input $\mathbf{x}$ such that $R_{f_I}(\mathbf{x}, \mathbf{y}) = 1$, where $y_i = H_{I,H_{\mathsf{out}}}(x_i)$ for each i. Let the random variable $\mathbf{X} = (X_1, \ldots, X_k)$ denote the output of $\mathcal{A}(I, H_{\mathsf{out}})$, let $Y_i = H_{I,H_{\mathsf{out}}}(X_i)$ for all i, and let the random variable Win denote the event that $R_{f_I}(\mathbf{X}, \mathbf{Y}) = 1$. We will call $\text{Expt}^{(0)}$ the security game described above.

- Consider the following modified experiment $\text{Expt}^{(1)}$. A challenger generates $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$, chooses uniformly random $\mathbf{X}^* \xleftarrow{\$} (\{0,1\}^n)^k$, and sends (I, H_{out}) to $\mathcal{A}$, which in turn outputs $\mathbf{X}$. Then, we have

$$\Pr^{(1)}[\text{Win} \wedge (\mathbf{X} = \mathbf{X}^*)] = \Pr^{(0)}[\text{Win}] \cdot 2^{-kn} \geq \frac{\epsilon}{2^{kn}}.$$

  We note that if the variables $Y_i^* := f_I(X_i^*)$ are not distinct in $\text{Expt}^{(1)}$ then $\mathcal{A}$ necessarily loses, so we redefine the game to immediately end if this occurs.

- Consider the further modified experiment $\text{Expt}^{(2)}$, defined as follows. The challenger generates $(I, \mathbf{X}^*)$ as above, and additionally generates $\mathbf{Z}^* \xleftarrow{\$} (\{0,1\}^m)^k$ uniformly at random. The challenger then samples $H_{\mathsf{out}} \leftarrow \mathcal{H}.\mathsf{CondGen}(\mathbf{Y}^*, \mathbf{Z}^*)$ and sends (I, H_{out}) to $\mathcal{A}$. Then, we have

$$\Pr^{(2)}[\text{Win} \wedge (\mathbf{X} = \mathbf{X}^*)] = \Pr^{(1)}[\text{Win} \wedge (\mathbf{X} = \mathbf{X}^*)] \geq \frac{\epsilon}{2^{kn}}$$

  by the programmability correctness of $\mathcal{H}_{\mathsf{out}}$ (and the fact that $\mathbf{Z}^*$ was chosen uniformly at random).

- Consider an experiment $\text{Expt}^{(3)}$ which differs from $\text{Expt}^{(2)}$ only in that $\mathbf{Z}^*$ is instead sampled by $S(1^n, 1^k)$, the sampling algorithm associated to R. Then

$$\begin{aligned}
\Pr^{(3)}[\mathbf{X} = \mathbf{X}^*] &\geq \Pr^{(3)}[\text{Win} \wedge \mathbf{X} = \mathbf{X}^*] \\
&= \Pr^{(2)}[\text{Win} \wedge \mathbf{X} = \mathbf{X}^* | R(\mathbf{Z}^*) = 1] && (8) \\
&= \frac{\Pr^{(2)}[\text{Win} \wedge \mathbf{X} = \mathbf{X}^*]}{\Pr^{(2)}[R(\mathbf{Z}^*) = 1]} && (9) \\
&\geq \frac{\epsilon \cdot 2^{-kn}}{p}. && (10)
\end{aligned}$$

  where Eq. (8) follows from our correctness requirement of the sampling algorithm S, Eq. (9) follows because the event "$\text{Win} \wedge \mathbf{X} = \mathbf{X}^*$" occurs *only* when $R(\mathbf{Z}^*) = 1$, and Eq. (10) follows from the p-sparsity of R.

- Finally, we note that $\text{Expt}^{(3)}$ leads to a $(s + t + t', \epsilon \cdot \frac{2^{-kn}}{p})$-attack on the k-OWPF security of $\mathcal{F}$. The attack is as follows: an adversary $\mathcal{A}'$ given $(I, \mathbf{Y}^*)$ as in the k-OWPF security

game can sample $\mathbf{Z}^* \leftarrow S(1^n, 1^k)$, sample $H_{\mathsf{out}} \leftarrow \mathcal{H}_{\mathsf{out}}.\mathsf{CondGen}(\mathbf{Y}^*, \mathbf{Z}^*)$, and run $\mathcal{A}(I, H_{\mathsf{out}})$. This perfectly simulates $\mathrm{Expt}^{(3)}$, and hence $\mathcal{A}'$ recovers $\mathbf{X}^*$ (which in particular satisfies $f_I(X_i^*) = Y_i^*$ for all i) with probability at least $\epsilon \cdot \frac{2^{-kn}}{p}$.

As a uniform algorithm, the OWPF adversary $\mathcal{A}'$ runs in time $s + t' + t$, but since the sampling step $\mathbf{Z}^* \leftarrow S(1^n, 1^k)$ is oblivious to the OWPF challenge $\mathbf{Y}^*$, by an averaging argument there *exists* some string $\mathbf{z}^* \in (\{0,1\}^m)^k$ such that $\mathcal{A}'$ with $\mathbf{Z}^* := \mathbf{z}^*$ hardwired also inverts $\mathbf{Y}^*$ with probability $\epsilon \cdot \frac{2^{-kn}}{p}$, which yields a nonuniform attack running in time $s + t'$. This concludes the proof of Theorem 5.3. □

Finally, we give a proof of Theorem 5.1.

Proof of Theorem 5.1. Suppose there is some size s adversary $\mathcal{A}$ that on input (I, H_{out}) outputs $\mathbf{x}$ such that all x_i are distinct and $R(\mathbf{y}) = 1$, where $y_i = H_{I,H_{\mathsf{out}}}(x_i)$ for all i, with probability δ'. Note that with probability $1-\eta$ over the randomness of $\mathcal{H}.\mathsf{Gen}$, the function f_I is injective, so by a union bound, $\mathcal{A}$ wins its security game *and* f_I is injective with probability at least $\delta' - \eta$. However, if $\mathcal{A}$ wins its security game and f_I is injective, then $\mathcal{A}$ has produced an input $\mathbf{x}$ such that $R_{f_I}(\mathbf{x}, \mathbf{y}) = 1$. But by Theorem 5.3, this can happen with probability at most $\delta \cdot 2^{kn} \cdot p$. Thus, we conclude that $\delta' \le \eta + \delta \cdot 2^{kn} \cdot p$, as desired. □

5.1 Examples Arising from Theorem 5.1

We now we describe some of the consequences of Theorem 5.1 for particular relations R of interest.

5.1.1 Collision Resistance

As a direct consequence of Theorem 5.1, we recover our theorem on collision resistance, Theorem 4.1. The relevant output relation is defined as follows: $R(y_1, y_2) = 1$ if and only if $y_1 = y_2$. R has sparsity 2^{-m}, where m is the output length of our hash function $H : \{0,1\}^n \to \{0,1\}^m$. Moreover, the set $\{\mathbf{y} : R(\mathbf{y}) = 1\}$ is exactly $\{(r, r) : r \in \{0,1\}^m\}$ and is therefore polynomial-time samplable (meaning that we do not have to rely on a non-uniform reduction).

Thus, we recover Theorem 4.1 from Theorem 5.1, as the distinguishing advantage δ produced by Theorem 5.1 is $\delta = 2^{-2n} \cdot 2^m \cdot \mathrm{negl}(n)$ for any negligible function $\mathrm{negl}(n)$.

5.1.2 Multi-Collision Resistance

By considering the k-ary output relation

$$R(\mathbf{y}) = 1 \text{ if and only if } y_1 = y_2 = \ldots = y_k,$$

we obtain a result on k-collision resistance [BDRV18, BKP18, KNY18] for any k. This relation has sparsity $2^{-(k-1)m}$ (if the hash function has output length m), and we can efficiently sample a random $\mathbf{y}$ such that $R(\mathbf{y}) = 1$ by choosing a uniformly random $r \leftarrow \{0,1\}^m$ and outputting $(r, \ldots, r)$. Thus, by Theorem 5.1, we have the following result.

Corollary 5.4. *If there exists an injective symmetric OWPF family with security (s, δ), then there exists a family of k-MCRHFs mapping $\{0,1\}^n \to \{0,1\}^m$ with security roughly $(s, \delta \cdot 2^{kn} \cdot 2^{-(k-1)m})$. (Moreover, this is proved by a uniform reduction.)*

In particular, for any $m = \omega(\log n)$, a plausible setting of δ yields a k-collision resistant hash family whose security matches the trivial attack of outputting k uniformly random points $x_1, \ldots, x_k$.

Finally, we consider the special case $m = n - \log(k)$ (the minimal compression to guarantee k-collisions) and polynomial security, in which case we require an injective OWPF that is $2^{-n-k\log(k)} \cdot \mathrm{negl}(n)$-secure. For example, in the case of $k = \frac{\alpha n}{\log(n)}$, we require $2^{-(1+\alpha)n}$-hardness of a problem for which the naive algorithm has success probability $2^{-\frac{\alpha^2 n^2}{\log^2(n)}}$. This is a substantially weaker OWPF assumption than is required for collision resistance.

We note that by [BDRV18, KNY18], k-collision-resistant hash functions (for any k) suffice to build constant-round statistically-hiding commitments, another primitive which we currently do not know how to construct from IO and one-way functions alone.

Additionally, the quantitatively weaker (injective symmetric) OWPF requirements for k-MCRHFs allow us to use reductions from both Section 3.3 and Section 3.4 to obtain constructions from various kinds of asymmetric and/or non-injective OWPFs. We refer the reader to these previous sections for details.

6 Constructions from IO and OWPFs

In this section, we combine OWPFs with the powerful notion of *indistinguishability obfuscation* in the hopes of obtaining better constructions of hash functions. We successfully obtain:

- A better quantitative tradeoff than in constructions based on general (i.e. *asymmetric*) OWPFs, avoiding a costly intermediate reduction such as Theorem 3.6. For example, we obtain a construction of CRHFs from IO and $2^{-n} \cdot \mathrm{negl}(n)$-secure injective 2-OWPFs (without any symmetry requirement).
- A hash family that is correlation intractable with respect to a broader class of relations than achievable with (symmetric) OWPFs alone. As described later, this includes an instantiation of the Fiat-Shamir transform for an expressive class of interactive proofs.

Moreover, our constuction is extremely simple: our hash function is an obfuscated (puncturable) PRF $\mathcal{O}(F_s(\cdot))$, and we only require the existence of OWPFs in the security proofs. As a byproduct, this construction confirms our intuition that obfuscated (puncturable) PRFs should satisfy many random oracle properties (including collision-resistance, despite the negative result of [AS15]). Our work in this section extends the proof technique of [KRR16], who show that an obfuscated puncturable PRF suffices for Fiat-Shamir assuming the existence of strong point function obfuscation.

6.1 Preliminaries

6.1.1 Indistinguishability Obfuscation

An *obfuscator for all circuits* is a ppt algorithm $\mathcal{O}$ such that for every circuit C, $\mathcal{O}(C)$ is with probability 1 a circuit $\tilde{C}$ with the same functionality as C. Various security properties may be defined for an obfuscator; the one most relevant to us is *indistinguishability obfuscation* [BGI+01].

Definition 6.1 (Indistinguishability Obfuscation)**.** *$\mathcal{O}$ is a (s, δ)-secure indistinguishability obfuscator (IO) if for all pairs of functionally equivalent circuits C_0 and C_1 of size $|C_0| = |C_1| = \lambda$, and all*

circuits $\mathcal{A}$ of size $s(\lambda)$, it holds that

$$\Pr[\mathcal{A}(\mathcal{O}(C_0)) = 1] - \Pr[\mathcal{A}(\mathcal{O}(C_1)) = 1] \leq O(\delta(\lambda)).$$

6.1.2 Puncturable PRFs

Definition 6.2 (Puncturable PRF [BW13, BGI14, KPTZ13, SW14])**.** *A PPRF family is a family of functions*

$$\mathcal{F} = \left\{F_{n,s} : \{0,1\}^n \to \{0,1\}^{m(n)}\right\}_{n\in\mathbb{N}, s\in\{0,1\}^{\ell(n)}}$$

with associated (deterministic) polynomial-time algorithms $(\mathcal{F}.\mathsf{Eval}, \mathcal{F}.\mathsf{Puncture}, \mathcal{F}.\mathsf{PuncEval})$ *satisfying*

- *For all $x \in \{0,1\}^n$ and all $s \in \{0,1\}^{\ell(n)}$, $\mathcal{F}.\mathsf{Eval}(s, x) = F_{n,s}(x)$.*
- *For all distinct $x, x' \in \{0,1\}^n$ and all $s \in \{0,1\}^{\ell(n)}$, $\mathcal{F}.\mathsf{PuncEval}(\mathcal{F}.\mathsf{Puncture}(s, x), x') = \mathcal{F}.\mathsf{Eval}(s, x')$.*

For ease of notation, we write $F_s(x)$ and $\mathcal{F}.\mathsf{Eval}(s, x)$ interchangeably, and we write $s\{x\}$ to denote $\mathcal{F}.\mathsf{Puncture}(s, x)$.

$\mathcal{F}$ is said to be (s, δ)-secure if for every $\{x^{(n)} \in \{0,1\}^n\}_{n\in\mathbb{N}}$, the following two distribution ensembles (indexed by n) are $\delta(n)$-indistinguishable to circuits of size $s(n)$:

$$(S\{x^{(n)}\}, F_S(x^{(n)})) \text{ where } S \leftarrow \{0,1\}^{\ell(n)}$$

and

$$(S\{x^{(n)}\}, U) \text{ where } S \leftarrow \{0,1\}^{\ell(n)}, U \leftarrow \{0,1\}^{m(n)}.$$

Theorem 6.1 ([GGM86, KPTZ13, BW13, BGI14, SW14])**.** *If {polynomially secure, subexponentially secure, subexponential advantage-secure} one-way functions exist, then for all functions $m : \mathbb{N} \to \mathbb{N}$ (with $1^{m(n)}$ polynomial-time computable from 1^n), and all $\delta : \mathbb{N} \to [0, 1]$ with $\delta(n) \geq 2^{-\mathrm{poly}(n)}$, there is a polynomial $\ell(n)$ and a {polynomially secure, $(\frac{1}{\delta}, \delta)$-secure, δ-secure} PPRF family*

$$\mathcal{F}_m = \left\{F_{n,s} : \{0,1\}^n \to \{0,1\}^{m(n)}\}_{n\in\mathbb{N}, s\in\{0,1\}^{\ell(n)}}\right\}.$$

6.2 Warm-Up: Target Collision Resistance

To demonstrate the power of our technique, we first show that an obfuscated PPRF $\mathcal{O}(F_s)$ is target collision-resistant (i.e. a UOWHF), only making use of the additional assumption that injective one-way functions exist. This result may be of independent interest – although one-way functions imply UOWHFs without additional assumptions [Rom90], we are not aware of any prior proof that $\mathcal{O}(F_s)$ (with suitable padding) is a UOWHF.[12] This result also demonstrates that the planting technique can be used without making any exponential assumptions.

Theorem 6.2. *Let $m : \mathbb{N} \to \mathbb{N}$ be a polynomial time computable function such that $n > m(n) \geq n - O(\log n)$. Suppose that*

[12] In contrast, a standard puncturing argument suffices to prove that $\mathcal{O}(G \circ F'_s)$ *is* target collision-resistant, where G denotes a PRG and F'_s denotes a PRF with output length $\frac{m}{2}$.

- $\mathcal{O}$ *is a sub-exponential advantage-secure indistinguishability obfuscator.*
- $\mathcal{F} = \left\{\{F_{n,s} : \{0,1\}^n \to \{0,1\}^{m(n)}\}_{s\in\{0,1\}^{\ell(n)}}\right\}_{n\in\{0,1\}^*}$ *is a family of* 2^{-2n}*-secure puncturable PRFs. We will use the notation* $F_s(\cdot)$ *as shorthand.*
- *There* exists *a family* $\mathcal{F}_{inj}$ *of (polynomially secure) injective one-way functions.*

Then, there is a polynomial $p : \mathbb{N} \to \mathbb{N}$ *such that the hash family* $\mathcal{H}$ *defined by* $H \leftarrow \mathcal{O}(P_s)$ *is a UOWHF family, where* P_s *is a program padded to have size* $p(n)$ *which on input* $x \in \{0,1\}^n$ *outputs* $F_s(x)$*.*

Proof Overview We will show that if an adversary $\mathcal{A}$ finds collisions in H with noticeable probability, then it also finds a random *planted* collision in H with noticeable probability. On the other hand, we hide the planted collision with an special-purpose obfuscator (based on any injective one-way function), which exactly prevents $\mathcal{A}$ from finding the planted collision with noticeable probability.

Proof. The polynomial $p(n)$ is chosen to be large enough so that $\mathcal{O}$ is 2^{-2n}-secure for programs of length $p(n)$, and so that all circuits obfuscated in our proof's hybrids have size at most $p(n)$ (in particular, $p(n)$ must be at least as large as the description of a function in $\mathcal{F}_{\text{inj}}$).

Suppose that $\mathcal{H}$ is not a UOWHF – namely, for some ppt $(\mathcal{A}_0, \mathcal{A}_1)$, some $c > 0$, and infinitely many n, in the experiment $\text{Expt}^{(0)}$ defined by sampling $(X, \mathsf{st}) \leftarrow \mathcal{A}_0(1^n)$, $S \leftarrow \{0,1\}^{\ell(n)}$, $H \leftarrow \mathcal{O}(P_S)$ and $X' := \mathcal{A}_1(H, \mathsf{st})$, it holds that

$$\Pr^{(0)}[\textsc{Win}] > m(n)^{-c} := m^{-c},$$

where $\textsc{Win}$ denotes the event that $X \neq X'$ but $H(X) = H(X')$.

- Consider an experiment $\text{Expt}^{(1)}$ which differs from $\text{Expt}^{(0)}$ only in that we additionally (and independently) sample $X^* \leftarrow \{0,1\}^n$. Then clearly

$$\Pr^{(1)}[\textsc{Win} \wedge (X' = X^*)] = \Pr^{(0)}[\textsc{Win}] \cdot 2^{-n} > \frac{1}{2^n m^c}.$$

- Consider an experiment $\text{Expt}^{(2)}$ which differs from $\text{Expt}^{(1)}$ only in the definition of H. Namely, H is defined not as $\mathcal{O}(P_S)$, but as $\mathcal{O}(P_{S,X^*,F_S(X^*)})$, where P_{s,x^*,y^*} is the appropriately padded circuit (with $s\{x^*\}$, x^*, and y^* hard-coded) that computes

$$P_{s,x^*,y^*}(x) = \begin{cases} y^* & \text{if } x = x^* \\ \mathsf{PuncEval}(s\{x^*\}, x) & \text{otherwise.} \end{cases}$$

  Because $P_{S,X^*,F_s(X^*)}$ is functionally equivalent to P_S, the 2^{-2n} security of $\mathcal{O}$ implies that

$$\Pr^{(2)}[\textsc{Win} \wedge (X' = X^*)] \geq \Pr^{(1)}[\textsc{Win} \wedge (X' = X^*)] - 2^{-2n} > \frac{1}{2^n m^c} - 2^{-2n}.$$

- Consider an experiment $\text{Expt}^{(3)}$ which differs from $\text{Expt}^{(2)}$ only in the definition of H. Namely, H is now sampled as $\mathcal{O}(P_{S,X^*,Y^*})$ for independently and uniformly random $Y^* \leftarrow \{0,1\}^m$. Now the 2^{-2n} punctured pseudorandomness of F_s at X^* implies that

$$\Pr^{(3)}[\textsc{Win} \wedge (X' = X^*)] \geq \Pr^{(2)}[\textsc{Win} \wedge (X' = X^*)] - 2^{-2n} > \frac{1}{2^n m^c} - 2 \cdot 2^{-2n}.$$

- Consider an experiment $\text{Expt}^{(4)}$ which differs from $\text{Expt}^{(3)}$ only in that Y^* is now defined as $Y^* := F_S(X)$. Then

$$\begin{aligned}\Pr^{(4)}\left[X' = X^*\right] &\geq \Pr^{(4)}\left[\text{WIN} \wedge (X' = X^*)\right] \\ &= \Pr^{(3)}\left[\text{WIN} \wedge (X' = X^*) \middle| Y^* = F_s(X)\right] \\ &= \frac{\Pr^{(3)}\left[\text{WIN} \wedge (X' = X^*)\right]}{\Pr^{(3)}\left[Y^* = F_s(X)\right]} \\ &> \left(\frac{1}{2^n m^c} - 2 \cdot 2^{-2n}\right) 2^m \geq \frac{1}{m^c \cdot 2^{O(\log(n))}} = \text{non-negl}(n),\end{aligned} \tag{11}$$

  where Eq. (11) follows because the event "$\text{WIN} \wedge (X' = X^*)$" occurs *only* when $Y^* = F_S(X)$.

- Finally, consider an experiment $\text{Expt}^{(5)}$ which differs from $\text{Expt}^{(4)}$ only in that H is now sampled as $\mathcal{O}(\tilde{P}_{S,f_I(X^*),Y^*})$, where $f_I \leftarrow \mathcal{F}_{\text{inj}}$ is sampled from the family of injective one-way functions, and $\tilde{P}_{s,w^*,y^*}$ is the circuit (with s, w^*, and y^* hard-coded) that computes

$$\tilde{P}_{s,w^*,y^*}(x) = \begin{cases} y^* & \text{if } f_I(x) = w^* \\ F_s(x) & \text{otherwise.} \end{cases}$$

  Since f_I is injective, we know that $\tilde{P}_{S,f_I(X^*),Y^*}$ is functionally equivalent to P_{S,X^*,Y^*}. We then have that $\Pr^{(5)}[X' = X^*] = \text{non-negl}(n)$ by the security of $\mathcal{O}$.

- However, this constitutes a polynomial-time inversion attack on $\mathcal{F}_{\text{inj}}$. Even if $\mathcal{A}$ were given $\tilde{P}_{S,f_{\text{inj}}(X^*),Y^*}$ in the clear, $\mathcal{A}$ should be unable to produce an inverse to $f_{\text{inj}}(X^*)$, as X^* is uniformly random and independent of S and Y^*. This contradicts the one-wayness of the family $\mathcal{F}_{\text{inj}}$, and so we have proved that $\mathcal{H}$ is a UOWHF. □

6.3 Multi-Input Correlation Intractability

In this section, we generalize the proof strategy of Section 6.2 to build multi-input correlation intractable hash functions – for a special class of relations that we define below – assuming the existence of IO, puncturable PRFs, and suitably secure injective k-OWPF families. The hardness that we need depends quantitatively on the *sparsity* of the relation R. Our proof relies on the observation that injective k-OWPFs allow us to obfuscate programs of the form

$$P_{x_1,\ldots,x_k}(x) = \begin{cases} i & x = x_i \text{ for some } i \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, by combining our result here with Construction 3.4, we obtain a construction from suitably secure (asymmetric and non-injective) k-OWPFs.

We refer the reader to Section 5 for the relevant definitions about correlation intractability. We again note that ideally, we would prove that an obfuscated (puncturable) PRF is correlation intractable for all sparse relations. Indeed, our proof reduces correlation intractability for any sparse R to the *existence* of an (extremely secure) special-purpose obfuscator that depends on R.[13] When R is a k-ary relation that satisfies a "local sampleability" property, we construct such an obfuscator from injective OWPFs.

[13] In general, it is not clear when such obfuscators exist, and upon which assumptions they can be based.

Definition 6.3 (Local Approximate Sampling with Setup)**.** *A relation $R \subseteq (\{0,1\}^n)^k \times (\{0,1\}^m)^k$ is locally ϵ-approximately samplable with t-setup if there are k polynomial time algorithms $S_1, S_2, \ldots, S_k$ and a probabilistic algorithm* Setup *such that:*

- $\mathsf{Setup}(1^n, 1^k)$ *runs in time t and outputs a string* CRS *of length* $\mathrm{poly}(n,k)$.
- *For every $(\mathbf{x}, \mathbf{y}) \in R$,*

$$\Pr_{\mathsf{CRS}}[S_i(x_i; \mathsf{CRS}) = y_i \text{ for all } i] \geq \epsilon \cdot \Pr_{\mathbf{Y} \leftarrow (\{0,1\}^m)^k}[\mathbf{Y} = \mathbf{y} \mid R(\mathbf{x}, \mathbf{Y}) = 1].$$

  In other words, for every $\mathbf{x}$, the distribution $(S_i(x_i; \mathsf{CRS}))_i$ approximates the uniform distribution on the set of $\mathbf{y}$ for which $R(\mathbf{x}, \mathbf{y}) = 1$ as long as this set is non-empty.

We further restrict our attention to "distinct-input" relations, as we do in Section 5.

Definition 6.4 (Distinct-Input Relation)**.** *A k-ary relation $R \in (\{0,1\}^n)^k \times (\{0,1\}^m)^k$ is a distinct-input relation if $R(\mathbf{x}, \mathbf{y}) = 0$ whenever $x_i = x_j$ for some $i \neq j \in [k]$.*

Let $\mathcal{R}_{k,t,\epsilon,p}$ denote the class of distinct-input k-ary relations

$$R = \{R_n \subset (\{0,1\}^n)^{k(n)} \times (\{0,1\}^m)^{k(n)}\}$$

that are p-sparse and locally ϵ-approximately samplable with t-setup, and let $\mathcal{R}_{k,\epsilon,p} := \bigcup_t \mathcal{R}_{k,t,\epsilon,p}$ denote the class of distinct-input k-ary relations that are p-sparse and locally ϵ-approximately samplable (with any setup time).

We now state our most general result on multi-input correlation intractability.

Theorem 6.3. *Let $\nu : \mathbb{N} \to \mathbb{R}$ be a function satisfying $\nu(n) \geq 2^{-\mathrm{poly}(n)}$, let $k : \mathbb{N} \to \mathbb{N}$ be any polynomial, let $T, t : \mathbb{N} \to \mathbb{N}$ satisfy $t(n) \leq 2^{\mathrm{poly}(n)}$ and $T(n) \leq k(n) \cdot 2^n$. Suppose also that*

- *$\mathcal{O}$ is a sub-exponentially secure[14] indistinguishability obfuscator.*
- *$\mathcal{F} = \left\{\{F_{n,s} : \{0,1\}^n \to \{0,1\}^{m(n)}\}_{s \in \{0,1\}^{\ell(n)}}\right\}_{n \in \mathbb{N}}$ is a family of $(2^{2kn}, \nu(m(n)) \cdot 2^{-2kn})$-secure puncturable PRFs. We will use the notation $F_s(\cdot)$ as shorthand for $F_{n,s}(\cdot)$.*
- *There* exists *a $(T + \mathrm{poly}(n), \delta)$-secure injective k-OWPF family $\mathcal{F}_{inj}$ for some $\delta = 2^{-kn} \cdot \frac{\epsilon}{p} \cdot \nu(m)$.*

Then, there is a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that the hash family $\mathcal{H}$ defined by $H \leftarrow \mathcal{O}(P_s)$ is $(T, \nu(m(\cdot))$-correlation intractable for $\mathcal{R}_{k,\epsilon,p}$, where P_s is a circuit that evaluates F_s (padded to size $p(n)$).

Moreover, for the restricted class $\mathcal{R}_{k,t,\epsilon,p}$, the reduction to OWPF security can be made uniform with an additional loss of t time.

Remark 6.1. *The restriction to distinct-input relations is primarily for ease of presentation; in particular, any $2k$-ary relation R is a union of at most k^k distinct-input relations, so at the cost of parameters that are worse by a factor of k^k, Theorem 6.3 can be applied to sparse relations not necessarily satisfying the distinct-input condition.*

[14] Correlation intractability for any fixed relation R can be achieved from a potentially weaker assumption; $\mathcal{O}$ and $\mathcal{F}$ must be secure against circuits of size that depends on t and the time to decide R.

Proof. The polynomial $p(n)$ is chosen to be large enough so that $\mathcal{O}$ is $(t + 2^{2kn}, \nu(m(n)) \cdot 2^{-2kn} \cdot \epsilon)$-secure for programs of length $p(n)$, and so that all circuits obfuscated in our proof's hybrids have size at most $p(n)$.

Let R be any relation in $\mathcal{R}_{k,t,\epsilon,p}$, and suppose that an adversary $\mathcal{A}$ breaks the $(T, \nu(m(\cdot)))$-correlation intractability of $\mathcal{H}$ for R. We define $\mathrm{Expt}^{(0)}$ to be the R-correlation intractability game: $S \leftarrow \{0,1\}^{\ell(n)}$, $H \leftarrow \mathcal{O}(P_S)$, and $\mathbf{X} := (X_1, \ldots, X_k) \leftarrow \mathcal{A}(H)$. Moreover, we define $\mathbf{Y} := Y_1 || \ldots || Y_k := H(X_1) || \ldots || H(X_k)$, and define Win to be the event that $R(\mathbf{X}, \mathbf{Y}) = 1$. We then argue as follows.

- Consider an experiment $\mathrm{Expt}^{(1)}$ which differs from $\mathrm{Expt}^{(0)}$ only in that we additionally (and independently) sample $\mathbf{X}^* := (X_1^*, \ldots, X_k^*) \leftarrow (\{0,1\}^n)^k$. Then,

$$\Pr^{(1)}[\textsc{Win} \wedge (\mathbf{X} = \mathbf{X}^*)] = \Pr^{(0)}[\textsc{Win}] \cdot 2^{-kn} > \omega(\nu(m)) \cdot 2^{-kn}.$$

  Note that when $(X_1^*, \ldots, X_i^*)$ are not distinct in $\mathrm{Expt}^{(1)}$, $\mathcal{A}$ necessarily loses, so we re-define the game to immediately end if this event occurs.

- Consider an experiment $\mathrm{Expt}^{(2)}$ which differs from $\mathrm{Expt}^{(1)}$ only in the definition of H. Namely, H is sampled not as $\mathcal{O}(P_S)$, but as $\mathcal{O}(P_{S,\mathbf{X}^*,\mathbf{Y}^*})$, where $\mathbf{Y}^* := (Y_1^*, \ldots, Y_k^*) \leftarrow (\{0,1\}^m)^k$ is drawn uniformly at random, and $P_{s,\mathbf{x}^*,\mathbf{y}^*}$ is the appropriately padded circuit (with s, $\mathbf{x}^*$, and $\mathbf{y}^*$ hard-coded) that computes

$$P_{s,\mathbf{x}^*,\mathbf{y}^*}(x) = \begin{cases} y_1^* & \text{if } x = x_1^* \\ \vdots & \vdots \\ y_k^* & \text{if } x = x_k^* \\ F_s(x) & \text{otherwise.} \end{cases}$$

  Then, we have that

$$\begin{aligned} \Pr^{(2)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*] &\geq \Pr^{(1)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*] - O(k \cdot 2^{-2kn}) \\ &> \frac{\omega(\nu(m))}{2^{kn}} - O(k \cdot \nu(m) \cdot 2^{-2kn}) = \frac{\omega(\nu(m))}{2^{kn}}. \end{aligned}$$

  where we have invoked the $(2^{2kn}, \nu(m(n)) \cdot 2^{-2kn})$ security[15] of $\mathcal{O}$ ($k+1$ times) and the $(2^{2kn}, \nu(m(n)) \cdot 2^{-2kn})$ security of $\mathcal{F}$ (k times) to puncture the program P_S at each X_i^*.

- Consider an experiment $\mathrm{Expt}^{(3)}$ which differs from $\mathrm{Expt}^{(2)}$ only in how $\mathbf{Y}^*$ is sampled. Specifically, conditioned on $\mathbf{X}^* = \mathbf{x}^*$, its distribution is uniform on $\{\mathbf{y} \in (\{0,1\}^m)^k : R(\mathbf{x}^*, \mathbf{y}) = 1\}$

[15] This level of security is required because determining whether Win occurs requires deciding R.

whenever this set is non-empty. Then,

$$\begin{aligned}
\Pr^{(3)}[\mathbf{X} = \mathbf{X}^*] &\geq \Pr^{(3)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*] \\
&= 2^{-kn} \sum_{\mathbf{x}^*} \Pr^{(3)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^* | \mathbf{X}^* = \mathbf{x}^*] \\
&= 2^{-kn} \sum_{\mathbf{x}^*} \Pr^{(2)}[\textsc{Win} \wedge R(\mathbf{x}^*, \mathbf{Y}^*) = 1 \wedge \mathbf{X} = \mathbf{X}^* | \mathbf{X}^* = \mathbf{x}^*] \\
&= 2^{-kn} \sum_{\mathbf{x}^*} \frac{\Pr^{(2)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^* \mid \mathbf{X}^* = \mathbf{x}^*]}{\Pr^{(2)}[R(\mathbf{x}^*, \mathbf{Y}^*) = 1]} \qquad (12) \\
&\geq 2^{-kn} \sum_{\mathbf{x}^*} \frac{\Pr^{(2)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^* \mid \mathbf{X}^* = \mathbf{x}^*]}{p} \qquad (13) \\
&= \frac{\Pr^{(2)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*]}{p} = \frac{\omega(\nu(m))}{2^{kn} \cdot p}
\end{aligned}$$

where Eq. (12) follows because the event "$\textsc{Win} \wedge \mathbf{X} = \mathbf{x}^*$" occurs *only* when $R(\mathbf{x}^*, \mathbf{Y}^*) = 1$, and Eq. (13) follows from the p-sparsity of R.

- Consider an experiment $\text{Expt}^{(4)}$ which differs from $\text{Expt}^{(3)}$ only in how $\mathbf{Y}^*$ is sampled. Specifically, conditioned on $\mathbf{X}^* = \mathbf{x}^*$, $\mathbf{Y}^*$ is equal to $(S_i(x_i^*, \mathsf{CRS}))_{i=1}^k$, where $\mathsf{CRS} \leftarrow \mathsf{Setup}(1^n || 1^k)\}$. Then,

$$\begin{aligned}
\Pr^{(4)}[\mathbf{X} = \mathbf{X}^*] &= \sum_{\mathbf{x}^*, \mathbf{y}^*} \Pr^{(4)}\Big[\mathbf{X} = \mathbf{X}^* \mid (\mathbf{X}^*, \mathbf{Y}^*) = (\mathbf{x}^*, \mathbf{y}^*)\Big] \Pr^{(4)}\Big[(\mathbf{X}^*, \mathbf{Y}^*) = (\mathbf{x}^*, \mathbf{y}^*)\Big] \\
&= 2^{-kn} \sum_{\mathbf{x}^*, \mathbf{y}^*} \Pr^{(4)}\Big[\mathbf{X} = \mathbf{X}^* \mid (\mathbf{X}^*, \mathbf{Y}^*) = (\mathbf{x}^*, \mathbf{y}^*)\Big] \Pr^{(4)}\Big[\mathbf{Y}^* = \mathbf{y}^* \mid \mathbf{X}^* = \mathbf{x}^*\Big] \\
&\geq \epsilon \cdot 2^{-kn} \sum_{\mathbf{x}^*, \mathbf{y}^*} \Pr^{(3)}\Big[\mathbf{X} = \mathbf{X}^* \mid (\mathbf{X}^*, \mathbf{Y}^*) = (\mathbf{x}^*, \mathbf{y}^*)\Big] \Pr^{(3)}\Big[\mathbf{Y}^* = \mathbf{y}^* \mid \mathbf{X}^* = \mathbf{x}^*\Big] \qquad (14) \\
&= \epsilon \cdot \Pr^{(3)}[\mathbf{X} = \mathbf{X}^*] = \frac{\epsilon \cdot \omega(\nu(m))}{2^{kn} \cdot p}
\end{aligned}$$

where Eq. (14) follows from the approximate sampling condition for $(S_1, \ldots, S_k)$ and the fact that $\text{Expt}^{(4)}$ and $\text{Expt}^{(3)}$ only differ in the sampling of $\mathbf{y}^*$.

- Finally, consider an experiment $\text{Expt}^{(5)}$ which differs from $\text{Expt}^{(4)}$ only in that H is now sampled as $\mathcal{O}(\tilde{P}_{S,\mathbf{W}^*,\mathsf{CRS}})$, where $W_i^* := f_i(X_i^*)$, $(f_1, \ldots, f_k) \leftarrow \mathcal{F}_{\text{inj}}$ is sampled from the OWPF family, and $\tilde{P}_{s,\mathbf{w}^*,\mathsf{crs}}$ is the circuit (with s, $\mathbf{w}^*$, and crs hard-coded) that computes

$$\tilde{P}_{s,\mathbf{w}^*,\mathsf{crs}}(x) = \begin{cases} S_i(x; \mathsf{crs}) & \text{if } f_i(x) = w_i^* \text{ for some } i \\ F_s(x) & \text{otherwise.} \end{cases}$$

Since the f_i are all injective, we know that $\tilde{P}_{S,\mathbf{W}^*,\mathsf{crs}}$ is functionally equivalent to $P_{S,\mathbf{X}^*,\mathbf{Y}^*}$ for $\mathbf{Y}^* = (S_i(X_i^*; \mathsf{CRS}))_{i=1}^k$. We then have that $\Pr^{(5)}[\mathbf{X} = \mathbf{X}^*] = \frac{\epsilon \cdot \omega(\nu(m))}{2^{kn} \cdot p}$ by the $(T + t + \text{poly}(n), \epsilon \cdot \nu(m) \cdot 2^{-2kn})$-security of $\mathcal{O}$.

- However, the adversary's success in $\text{Expt}^{(5)}$ contradicts the $(t + \text{poly}(n), 2^{-kn} \cdot \frac{\epsilon}{p} \cdot \nu(m))$-security of $\mathcal{F}_{\text{inj}}$. In particular, a modified adversary $\mathcal{B}$ given only $W_i^* := f_i(X_i^*)$ for all i could sample $\tilde{P}_{S,\mathbf{W}^*,\mathsf{CRS}}$ itself in time $t + \text{poly}(n)$ and feed this output to $\mathcal{A}$, solving the batch inversion problem with probability $\frac{\epsilon \cdot \omega(\nu(m))}{2^{kn} \cdot p}$. This constitutes a $(T + t + \text{poly}(n), \frac{\epsilon \cdot \omega(\nu(m))}{2^{kn} \cdot p})$ attack on the OWPF family, which completes the claimed uniform reduction. Moreover, we note that the CRS sampling algorithm $\mathsf{Setup}(1^n, 1^k)$ is oblivious to the OWPF challenge, so by an averaging argument there *exists* some string crs such that $\mathcal{B}$ with $\mathsf{CRS} := \mathsf{crs}$ hardcoded wins the OWPF security game with the same probability. This completes the nonuniform reduction, proving correlation intractability for every $R \in \mathcal{R}_{k,\epsilon,p}$. □

6.4 Examples Arising from Theorem 6.3

We now we describe some of the consequences of Theorem 6.3 for particular relations R of interest.

6.4.1 Collision Resistance

As a direct consequence of Theorem 6.3, we obtain a second construction of collision-resistant hash functions. Similarly to before, the relevant relation is defined as follows: $R(x_1, x_2, y_1, y_2) = 1$ if and only if $x_1 \neq x_2$ and $y_1 = y_2$. As noted in Section 5.1, R has sparsity 2^{-m}, and the set $\{\mathbf{y} : R(\mathbf{x}, \mathbf{y}) = 1\}$ is efficiently sampleable in a way that is *oblivious* to the input $\mathbf{x}$. Thus R is clearly locally 1-sampleable with polynomial-time setup.

Thus, we obtain the following corollary.

Corollary 6.4. *If $\mathcal{O}$ is a sub-exponential advantage-secure indistinguishability obfuscator, $\mathcal{F}$ is a sub-exponential advantage-secure puncturable PRF, and there exists a δ-secure injective 2-OWPF family, then an $\mathcal{O}$-obfuscation of a (sufficiently padded) PRF chosen from $\mathcal{F}$ is $\delta \cdot 2^{2n} \cdot 2^{-m}$- collision resistant (by a uniform reduction).*

This exactly matches the quantitative parameters of Theorem 4.1. However, there are significant differences between the two results, namely:

- Corollary 6.4 requires the existence of sub-exponential advantage-secure IO, but
- Corollary 6.4 only requires (injective) OWPFs rather than *symmetric* (injective) OWPFs. Moreover, Corollary 6.4 only requires that such OWPFs exist; they are not required in the construction itself. Theorem 4.4, even when combined with the reductions of Section 3, was unable to produce a construction of CRHFs from (injective) asymmetric OWPFs.

Since the quantitative parameters of Corollary 6.4 match those of Theorem 4.1, this also yields CRHFs with optimal security under plausible OWPF assumptions (and IO).

6.4.2 Output Intractability

We also obtain an analog to Theorem 5.1; that is, a result on output intractability.

Corollary 6.5. *If $\mathcal{O}$ is a sub-exponentially secure indistinguishability obfuscator, $\mathcal{F}$ is a sub-exponentially secure puncturable PRF, and there exists a δ-secure injective k-OWPF family, then a $\mathcal{O}$-obfuscation of a (sufficiently padded) PRF chosen from $\mathcal{F}$ is $\delta \cdot 2^{kn} \cdot p$-output intractable for all k-ary output relations R. Moreover, this reduction can be made uniform (with a time t loss) if R is t-samplable.*

Again, this involves the same quantitative OWPF parameters as in Theorem 5.1, with the same tradeoff as in the collision resistance example above.

6.4.3 An Example Falling Outside the Output Intractability Framework

All of our previous examples are special cases of output intractability as defined in [Zha16] (albeit with possibly unbounded relations, unlike [Zha16]). On the other hand, consider the following relation on $(\{0,1\}^n)^2 \times (\{0,1\}^m)^2$, parametrized by a matrix $\mathbf{A} \in \mathbb{F}_2^{m \times n}$:

$$R_{\mathbf{A}}(x_1, x_2, y_1, y_2) = 1 \text{ if } x_1 \neq x_2 \text{ and } y_1 \oplus y_2 = \mathbf{A}(x_1 \oplus x_2).$$

This is clearly not a special case of output intractability (the relation depends explicitly on both the inputs and outputs). However, it falls into the framework captured by Theorem 6.3. The relation $R_{\mathbf{A}}$ has sparsity 2^{-m}. We can also sample, for any $x_1 \neq x_2 \in \{0,1\}^m$, a random (y_1, y_2) such that $R_{\mathbf{A}}(x_1, x_2, y_1, y_2) = 1$ with the algorithms

$$S_i(x_i; r) = r \oplus \mathbf{A}x_i.$$

Thus, we see that an obfuscated PRF is correlation intractable for these relations assuming an injective 2-OWPF family with the exact same parameters as those required for collision resistance.

In fact, this example extends to the following relation on $(\{0,1\}^n)^2 \times (\{0,1\}^m)^2$, parametrized by a matrix $\mathbf{A} \in \mathbb{F}_2^{d \times n}$ and a full-rank matrix $\mathbf{B} \in \mathbb{F}_2^{d \times m}$, as long as $2^{-d} = \text{negl}(n)$:

$$R_{\mathbf{A},\mathbf{B}}(x_1, x_2, y_1, y_2) = 1 \text{ if } x_1 \neq x_2 \text{ and } \mathbf{B}(y_1 \oplus y_2) = \mathbf{A}(x_1 \oplus x_2).$$

6.4.4 The Fiat-Shamir Transform for Commit-Challenge-Response Proofs

Theorem 6.3 is also applicable in the case $k = 1$: we give new sufficient conditions for the provably secure instantiation of the Fiat-Shamir heuristic [FS86], for an expressive class of interactive proof systems. Namely, we consider the familiar example of "commit-challenge-response" proofs.

Definition 6.5 (Commit-Challenge-Response Proof System)**.** *A 3-message proof system $\Pi = (P, V)$ is called a* commit-challenge-response *proof system for a language L if it satisfies the following properties.*

1. *The first message is sent by the prover to the verifier. This message, which we denote by a, consists of a block-wise commitment (under a statistically binding commitment scheme) to a string y that is a function of both the common input x and the prover's private input w.*

2. *The second message, which we denote by e and refer to as the verifier's "challenge", is sent by the verifier to the prover and is sampled uniformly at random from a* $\text{poly}(\lambda)$*-size alphabet Σ.*

3. *The third and final message, which we denote by z, is sent by the prover to the verifier, and consists of a* decommitment *to y_T, i.e., a subset T of the blocks of y. Here, T is a function of the challenge e.*

4. *The verifier V accepts if and only if (1) z is a valid decommitment of a_T, and (2) the tuple (x, y_T, e) passes some efficient test* Check*, where y_T is the value to which a_T was decommitted.*

Examples of commit-challenge-response proof systems include the classical 3-message zero knowledge protocol for 3-coloring [GMW91] as well as the 3-message zero knowledge protocol for Hamiltonicity given by [FLS99] (with a slight modification).

As we will see shortly, it is possible to use Theorem 6.3 to instantiate the Fiat-Shamir heuristic for any commit-challenge-response protocol (repeated in parallel). The key advantage to using our approach over that of [KRR16], or the more recent work of [CCRR18], is that we prove security only assuming that IO and exponentially secure one-way functions exist, rather than needing (exponentially secure) input-hiding obfuscation for arbitrary multi-bit point functions (for [KRR16]) or exponentially secure KDM-secure secret key encryption with respect to arbitrary functions (for [CCRR18]).

Theorem 6.6. *Let $\Pi = (P, V)$ be a commit-challenge-response proof system for some language $L \in \mathsf{NP}$ with soundness error $\mu = \mu(n)$, where n denotes the length of a first message a. Moreover, let $|\Sigma| = |\Sigma(n)|$ be the number of possible challenges associated to a single commit message $a \in \{0,1\}^n$, let $N = \lambda|\Sigma|n$ (for arbitrarily related $n = \mathrm{poly}(\lambda)$), and suppose that*

- *$\mathcal{O}$ is a sub-exponential advantage secure indistinguishability obfuscator.*
- *$\mathcal{F} = \left\{ \{F_{n,\lambda,s} : \{0,1\}^N \to \{0,1\}^{\lambda|\Sigma|\log|\Sigma|}\}_{s\in\{0,1\}^{\ell(n)}} \right\}_{n\in\mathbb{N}}$ is a family of $(\mathrm{poly}(N), 2^{-2N})$-secure puncturable PRFs. We will use the notation $F_s(\cdot)$ as shorthand for $F_{n,s}(\cdot)$.*
- *There* exists *a δ-secure injective OWF family $\mathcal{F}_{inj}$ for some $\delta = 2^{-N} \cdot (\frac{1}{\mu})^{\lambda|\Sigma|} \cdot \mathrm{negl}(N)$ taking inputs of length N.*

Then, if Π is instantiated using a public key encryption scheme to commit (where the public key is provided as a common reference string and commitment is encryption), then there is a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a such that the hash family $\mathcal{H}$ defined by $H \leftarrow \mathcal{O}(P_s)$ instantiates the Fiat-Shamir heuristic[16] for a $\lambda|\Sigma|$-wise parallel repetition of Π, where P_s is a circuit (padded to size $p(n)$) that evaluates F_s.

Moreover, if Π is honest verifier zero-knowledge, then the new 1-message proof system Π' is also zero knowledge (with a programmable CRS).

Remark 6.2. *By Section 3.3, the same result holds if there exists a δ'-secure (not necessarily injective) OWF family for some $\delta'(N) = 2^{-N} \cdot 2^{\frac{N}{3}} \delta(\frac{N}{3})$.*

Applying Theorem 6.6 to either the 3-colorability protocol of [GMW91] or the Hamiltonicity protocol of [FLS99] yields a construction of NIZK arguments (in the common reference string model). While NIZK proofs from IO and OWFs are already known by [BP15], this yields a construction of NIZK arguments through the Fiat-Shamir transform.

Proof of Theorem 6.6. Let x be any string *not* in the language L, and let crs be a random CRS for the commitment scheme used in Π. We would like to apply Theorem 6.3 to the single input-output relation

$$R = \left\{ R_\lambda := \left\{ (\mathbf{a}, \mathbf{e}) : \text{ there exists } \mathbf{z} \text{ such that } (\mathbf{a}, \mathbf{e}, \mathbf{z}) \text{ is an accepting transcript for } \Pi^{\lambda|\Sigma|} \right\} \right\},$$

[16] in the common reference string model

which is a $\mu^{\lambda|\Sigma|}$-sparse relation for any $x \notin L$. Unfortunately, it is not clear that R satisfies the hypotheses of Theorem 6.3; namely, it is unclear whether R is efficiently samplable. This issue can be fixed with two modifications:

- We instantiate the commitment scheme using a public key encryption scheme, where the public key is provided as a common reference string.
- We replace the relation R_λ with a relaxed relation $\tilde{R}_{\lambda,\mathsf{sk}}$ that is in $\mathcal{R}_{1,0,\mu^{\lambda|\Sigma|}}$.

More specifically, the modified relation $\tilde{R}_{\lambda,\mathsf{sk}}$ is defined as follows:

$$\tilde{R}_{\lambda,\mathsf{sk}} = \left\{(\mathbf{a},\mathbf{e}) : \text{Check}\left(x, y^{(i)}_{T(e^{(i)})}, e^{(i)}\right) = 1 \text{ for all } i, \text{ where } \mathbf{y} = \mathsf{Dec}(\mathsf{sk},\mathbf{a})\right\}.$$

We first note that $\tilde{R}_{\lambda,\mathsf{sk}}$ is a strict relaxation (superset) of R_λ when the commitment scheme for Π is instantiated with a public key encryption scheme. This follows from (1) the definition of a commit-challenge-response protocol, and (2) the fact that given a first message $a^{(i)}$, the only possible valid decommitment to any block of $a^{(i)}$ is the corresponding block of $\mathsf{Dec}(\mathsf{sk}, a^{(i)})$.

Moreover, it is easy to see that $\tilde{R}_{\lambda,\mathsf{sk}}$ is efficiently (locally) samplable. The sampling algorithm is as follows: given $\mathbf{a} = a^{(1)}||\ldots||a^{(\lambda|\Sigma|)}$ and sk, compute $\mathbf{y} = \mathsf{Dec}(\mathsf{sk},\mathbf{a})$. Then, for every $i \in [\lambda]$, do the following procedure: for every $e \in \Sigma$, run $\text{Check}(x, y^{(i)}_{T(e)}, e)$, and then sample $e^{(i)}$ uniformly at random from the set of e for which Check outputs 1. The sampling algorithm outputs $\mathbf{e} = e^{(1)}||\ldots||e^{(\lambda|\Sigma|)}$.

Since $\tilde{R}_{\lambda,\mathsf{sk}}$ is efficiently (locally) samplable and has sparsity $\mu^{\lambda|\Sigma|}$, we conclude that the hash family $\mathcal{H}$ is correlation intractable for $\tilde{R}_{\lambda,\mathsf{sk}}$ by Theorem 6.3. Moreover, since $\tilde{R}_{\lambda,\mathsf{sk}}$ is a relaxation of the relation R_λ, we conclude that it is hard for an efficient adversary $\mathcal{A}(H)$ to produce any message $\mathbf{a}$ such that $(\mathbf{a}, H(\mathbf{a}), \mathbf{z})$ is an accepting transcript for any possible $\mathbf{z}$. Thus, the Fiat-Shamir 1-message protocol is sound, as desired.

To show that the protocol is zero knowledge (if Π is honest verifier zero knowledge), we define the simulator Sim' for the 1-message protocol in terms of an honest-verifier simulator Sim for Π:

1. Sample a public key pk for the public key encryption scheme.
2. Run $\mathsf{Sim}(x, \mathsf{pk})$ independently $\lambda|\Sigma|$ times to obtain simulated transcripts $(\tilde{a}^{(i)}, \tilde{e}^{(i)}, \tilde{z}^{(i)})_{i \le \lambda|\Sigma|}$.
3. Letting $\tilde{\mathbf{a}} = (\tilde{a}^{(1)}, \ldots, \tilde{a}^{(\lambda|\Sigma|)})$, $\tilde{\mathbf{e}} = (\tilde{e}^{(1)}, \ldots, \tilde{e}^{(\lambda|\Sigma|)})$, and $\tilde{\mathbf{z}} = (\tilde{z}^{(1)}, \ldots, \tilde{z}^{(\lambda|\Sigma|)})$, compute the obfuscated program $\tilde{H} = \mathcal{O}(P_{s,\tilde{\mathbf{a}},\tilde{\mathbf{e}}})$, where

$$P_{s,\tilde{\mathbf{a}},\tilde{\mathbf{e}}}(x) = \begin{cases} \tilde{\mathbf{e}} & \text{if } x = \tilde{\mathbf{a}} \\ F_s(x) & \text{otherwise.} \end{cases}$$

4. Output $(\widetilde{\mathsf{CRS}}, \tilde{\pi}) = ((\mathsf{pk}, \tilde{H}), (\tilde{\mathbf{a}}, \tilde{\mathbf{z}}))$.

The proof that Sim' samples from a distribution computationally indistinguishable from an honest proof follows by a hybrid argument: first, convert $(\tilde{\mathbf{a}}, \tilde{\mathbf{e}}, \tilde{\mathbf{z}})$ to a collection $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ of $\lambda|\Sigma|$ honest Π-proofs by the security of Sim, and then convert the obfuscated program $\mathcal{O}(P_{s,\mathbf{a},\mathbf{e}})$ into an obfuscated program $\mathcal{O}(P_s)$ by obfuscation and puncturing security. □

Acknowledgements

The authors thank Zvika Brakerski, Yael Kalai, Omer Paneth, and Vinod Vaikuntanathan for helpful discussions and comments.

References

[AS15] Gilad Asharov and Gil Segev, *Limits on the Power of Indistinguishability Obfuscation and Functional Encryption*, 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, 2015.

[BDRV18] Itay Berman, Akshay Degwekar, Ron D Rothblum, and Prashant Nalini Vasudevan, *Multi collision resistant hash functions and their applications*, Annual International Conference on the Theory and Applications of Cryptographic Techniques – EUROCRYPT 2018, Springer, 2018.

[BDSG+13] Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana López-Alt, and Daniel Wichs, *Why âĂIJfiat-shamir for proofsâĂİ lacks a proof*, Theory of Cryptography, Springer, 2013, pp. 182–201.

[BDV17] Nir Bitansky, Akshay Degwekar, and Vinod Vaikuntanathan, *Structure vs. hardness through the obfuscation lens*, Annual International Cryptology Conference – CRYPTO 2017, Springer, 2017, pp. 696–723.

[BGI+01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang, *On the (im) possibility of obfuscating programs*, Annual International Cryptology Conference – CRYPTO 2001, Springer, 2001, Journal version appears in JACM 2012, pp. 1–18.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan, *Functional signatures and pseudorandom functions*, Public Key Cryptography, Lecture Notes in Computer Science, vol. 8383, Springer, 2014, pp. 501–519.

[BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth, *Multi-collision resistance: A paradigm for keyless hash functions*, STOC, 2018.

[BLVW18] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs, *Cryptographic hashing and worst-case hardness for lpn via code smoothing.*

[BP15] Nir Bitansky and Omer Paneth, *Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation*, TCC (2), Lecture Notes in Computer Science, vol. 9015, Springer, 2015, pp. 401–427.

[BR93] Mihir Bellare and Phillip Rogaway, *Random oracles are practical: A paradigm for designing efficient protocols*, Proceedings of the 1st ACM conference on Computer and communications security, ACM, 1993, pp. 62–73.

[BW13] Dan Boneh and Brent Waters, *Constrained pseudorandom functions and their applications*, ASIACRYPT (2), Lecture Notes in Computer Science, vol. 8270, Springer, 2013, pp. 280–300.

[Can97] Ran Canetti, *Towards realizing random oracles: Hash functions that hide all partial information*, IACR Cryptology ePrint Archive **1997** (1997), 7.

[CCR16] Ran Canetti, Yilei Chen, and Leonid Reyzin, *On the correlation intractability of obfuscated pseudorandom functions*, Theory of Cryptography Conference, Springer, 2016, pp. 389–415.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron Rothblum, *Fiat-shamir and correlation intractability from strong kdm-secure encryption*, Annual International Conference on the Theory and Applications of Cryptographic Techniques – EUROCRYPT 2018, Springer, 2018.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi, *The random oracle methodology, revisited*, Journal of the ACM (JACM) **51** (2004), no. 4, 557–594.

[CK18] Henry Corrigan-Gibbs and Dmitry Kogan, *The discrete-logarithm problem with preprocessing*, Annual International Conference on the Theory and Applications of Cryptographic Techniques – EUROCRYPT 2018, 2018.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold, *Perfectly one-way probabilistic hash functions (preliminary version)*, STOC, ACM, 1998, pp. 131–140.

[Dam87] Ivan Damgård, *Collision free hash functions and public key signature schemes*, EUROCRYPT, Lecture Notes in Computer Science, vol. 304, Springer, 1987, pp. 203–216.

[DJMW12] Yevgeniy Dodis, Abhishek Jain, Tal Moran, and Daniel Wichs, *Counterexamples to hardness amplification beyond negligible*, Theory of Cryptography Conference, Springer, 2012, pp. 476–493.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir, *Multiple noninteractive zero knowledge proofs under general assumptions*, SIAM Journal on Computing **29** (1999), no. 1, 1–28.

[FS86] Amos Fiat and Adi Shamir, *How to prove yourself: Practical solutions to identification and signature problems*, Conference on the Theory and Application of Cryptographic Techniques, Springer, 1986, pp. 186–194.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters, *Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits*, Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on, 2013, pp. 40–49.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali, *How to construct random functions*, J. ACM **33** (1986), no. 4, 792–807.

[GK03] Shafi Goldwasser and Yael Tauman Kalai, *On the (in) security of the fiat-shamir paradigm*, Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on, IEEE, 2003, pp. 102–113.

[GK16] ———, *Cryptographic assumptions: A position paper*, Theory of Cryptography Conference, Springer, 2016, pp. 505–522.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson, *Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems*, Journal of the ACM (JACM) **38** (1991), no. 3, 690–728.

[Gol04] Oded Goldreich, *The foundations of cryptography - volume 2, basic applications*, Cambridge University Press, 2004.

[GR07] Shafi Goldwasser and Guy N Rothblum, *On best-possible obfuscation*, Theory of Cryptography Conference, Springer, 2007, pp. 194–213.

[GW11] Craig Gentry and Daniel Wichs, *Separating succinct non-interactive arguments from all falsifiable assumptions*, Proceedings of the forty-third annual ACM symposium on Theory of computing, ACM, 2011, pp. 99–108.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby, *A pseudorandom generator from any one-way function*, SIAM Journal on Computing **28** (1999), no. 4, 1364–1396.

[Kil94] Joe Kilian, *On the complexity of bounded-interaction and noninteractive zero-knowledge proofs*, Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on, IEEE, 1994, pp. 466–477.

[KNY18] Ilan Komargodski, Moni Naor, and Eylon Yogev, *Collision resistant hashing for paranoids: Dealing with multiple collisions*, Annual International Conference on the Theory and Applications of Cryptographic Techniques – EUROCRYPT 2018, Springer, 2018.

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias, *Delegatable pseudorandom functions and applications*, IACR Cryptology ePrint Archive **2013** (2013), 379.

[KRR16] Yael Tauman Kalai, Guy Rothblum, and Ron Rothblum, *From Obfuscation to the Security of Fiat-Shamir for Proofs*, Advances in Cryptology - CRYPTO 2017, 2016.

[LP09] Yehuda Lindell and Benny Pinkas, *A proof of security of yaoâĂŹs protocol for two-party computation*, Journal of Cryptology **22** (2009), no. 2, 161–188.

[Nao91] Moni Naor, *Bit commitment using pseudorandomness*, Journal of cryptology **4** (1991), no. 2, 151–158.

[Nao03] ———, *On cryptographic assumptions and challenges*, Annual International Cryptology Conference – CRYPTO 2003, Springer, 2003, pp. 96–109.

[NY89] Moni Naor and Moti Yung, *Universal one-way hash functions and their cryptographic applications*, Proceedings of the twenty-first annual ACM symposium on Theory of computing, ACM, 1989, pp. 33–43.

[PW11] Chris Peikert and Brent Waters, *Lossy trapdoor functions and their applications*, SIAM Journal on Computing **40** (2011), no. 6, 1803–1844.

[Rom90] John Rompel, *One-way functions are necessary and sufficient for secure signatures*, STOC, ACM, 1990, pp. 387–394.

[Sho97] Victor Shoup, *Lower bounds for discrete logarithms and related problems*, International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 1997, pp. 256–266.

[Sim98] Daniel R Simon, *Finding collisions on a one-way street: Can secure hash functions be based on general assumptions?*, International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 1998, pp. 334–345.

[SW14] Amit Sahai and Brent Waters, *How to use indistinguishability obfuscation: deniable encryption, and more*, STOC, ACM, 2014, pp. 475–484.

[Wee05] Hoeteck Wee, *On obfuscating point functions*, Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, ACM, 2005, pp. 523–532.

[Wic18] Daniel Wichs, personal communication, April 2018.

[Yao86] Andrew Chi-Chih Yao, *How to generate and exchange secrets*, Foundations of Computer Science, 1986., 27th Annual Symposium on, IEEE, 1986, pp. 162–167.

[YZW+17] Yu Yu, Jiang Zhang, Jian Weng, Chun Guo, and Xiangxue Li, *Learning parity with noise implies collision resistant hashing*, https://eprint.iacr.org/2017/1260.pdf.

[Zha16] Mark Zhandry, *The magic of elfs*, Proceedings, Part I, of the 36th Annual International Cryptology Conference on Advances in Cryptology—CRYPTO 2016-Volume 9814, Springer-Verlag New York, Inc., 2016, pp. 479–508.

A A Proof of the Refined Asharov-Segev Bound

In this section, we prove Theorem 1.7 (a refined analysis of the Asharov-Segev impossibility result [AS15]). We now formally state Theorem 1.7.

Theorem A.1. *There exists an oracle* Γ' *and an oracle* $\Gamma = (\Gamma', \mathsf{CollFind}^{\Gamma'})$ *such that no hash function built relative to* Γ' *is collision-resistant relative to* Γ*, and such that the following cryptographic primitives* can be built *relative to* Γ' *(and are secure relative to* Γ*):*

1. $(2^{\frac{n}{15}}, 2^{-\frac{n}{40}})$*-secure indistinguishability obfuscation*
2. $(2^{\frac{n}{50}}, 2^{-\frac{n}{50}})$*-secure one-way permutations.*
3. *A one-way permutation which is* $(q(n), q(n)^c \cdot n \cdot 2^{-n})$*-secure for every polynomial* q *(for some absolute constant* c*).*

As in [AS15], the oracle Γ is defined as follows.

Definition A.1 (Asharov-Segev Oracle)**.** *The Asharov-Segev oracle* $\Gamma = (\Gamma', \mathsf{CollFind}^{\Gamma'}) = (f, \mathcal{O}, \mathsf{Eval}^{f,\mathcal{O}}, \mathsf{CollFind}^{f,\mathcal{O},\mathsf{Eval}})$ *consists of four parts:*

1. *A uniformly random permutation* $f = f^{(n)} : \{0,1\}^n \to \{0,1\}^n$ *for every input length* n*.*
2. *A uniformly random permutation* $\mathcal{O} = \mathcal{O}^{(n)} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ *for every input length* n*.*

3. *The function* $\mathsf{Eval}^{f,\mathcal{O}}$*, on input* $(z,x) \in \{0,1\}^* \times \{0,1\}^*$*, finds the unique string* $D||r$ *such that* $\mathcal{O}(D||r) = z$ *and outputs* $D^f(x)$*. The combination of* $\mathcal{O}$ *and* Eval *will serve as our indistinguishability obfuscator.*

4. *A collision-finding oracle* $\mathsf{CollFind}^{f,\mathcal{O},\mathsf{Eval}}$*: on any input* $C^{f,\mathcal{O},\mathsf{Eval}}$ *(which is a circuit with* $f, \mathcal{O}$*, and* Eval*-gates),* $\mathsf{CollFind}$ *outputs a random* $w \xleftarrow{\$} \{0,1\}^t$ *(where* t *is the input length of* C*), as well as a uniformly random* w' *of the same input length subject to the condition that* $C^{f,\mathcal{O},\mathsf{Eval}}(w) = C^{f,\mathcal{O},\mathsf{Eval}}(w')$.

We refer the reader to [AS15] for details on Γ (in particular, on the specific implementation of $\mathsf{CollFind}$).

In [AS15], it is shown that CRHFs (implementable relative to Γ') do not exist relative to Γ (Claim 3.5 in [AS15]), $(2^{\frac{n}{15}}, 2^{-\frac{n}{40}})$-secure indistinguishability obfuscation exists relative to Γ (Theorem 3.8 in [AS15]), and $(2^{\frac{n}{50}}, 2^{-\frac{n}{50}})$-secure one way permutations exist relative to Γ (Theorem 3.20 in [AS15]). In particular, the one-way permutation they prove secure is f itself. We now strengthen their result to show that f is (nearly) 2^{-n}-secure.

Lemma A.2. *Let* $q(n)$ *denote any polynomial function of* n*. Then, any adversary* $\mathcal{A}^\Gamma$ *which is given* $y = f(x)$ *(for* $x \xleftarrow{\$} \{0,1\}^n$*) and makes at most* $q(n)$ *queries to* Γ *(each of size at most* $q(n)$*) will output* x *with probability at most* $q(n)^c \cdot n \cdot 2^{-n}$*, for some absolute constant* c*. The probability here is taken over the choice of* x *as well as the choice of oracles* $(f^{(n)}, \mathsf{CollFind})$ *(but holds for any oracle* $\mathcal{O}$*).*

The rest of this section is devoted to establishing Lemma A.2 with the help of [AS15]. The proof proceeds as follows.

Suppose that some adversary $\mathcal{A}^\Gamma$ is given $y = f(x)$ (for $x \xleftarrow{\$} \{0,1\}^n$) and outputs x with probability ϵ. Define $\textsc{Win}_{\mathcal{A}}$ to be the event that $f(\mathcal{A}^\Gamma(y)) = y$. Moreover, define the $\mathsf{CollHit}_{y,\mathcal{A}}$ to be the event that $\mathcal{A}$ makes *some call to the* $\mathsf{CollFind}$ *oracle* which outputs (w, w') with one of the following two properties:

1. Some f-gate in the circuit evalutation $C^{f,\mathcal{O},\mathsf{Eval}}(w)$ or $C^{f,\mathcal{O},\mathsf{Eval}}(w')$ has output y, OR
2. Some Eval-gate in $C^{f,\mathcal{O},\mathsf{Eval}}(w)$ or $C^{f,\mathcal{O},\mathsf{Eval}}(w')$ has input $(\hat{D}, a)$ such that $D^f(a)$ has an f-gate with output y, where D is the unique circuit such that $\mathcal{O}(D, r) = \hat{D}$ for some r.

Our refined analysis (as compared to [AS15]) is the following claim (and proof).

Claim A.2.1. *Given* $\mathcal{A}$ *as above, there exists an algorithm* $\mathcal{B}^{f,\mathcal{O},\mathsf{Eval},\mathsf{CollFind}}$ *which makes at most* $3q(n)^3$ *queries to* f*,* $q(n)$ *queries to* Eval*, and* $q(n)$ *queries to* $\mathsf{CollFind}$*, such that*

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \geq \frac{\epsilon}{6}.$$

Proof. We may assume that

$$\Pr[\textsc{Win}_{\mathcal{A}} \wedge \mathsf{CollHit}_{y,\mathcal{A}}] \geq \frac{\epsilon}{2},$$

because otherwise we may just set $\mathcal{B} = \mathcal{A}$. In the remaining case, we define $\mathcal{B}$ as follows: $\mathcal{B}^{f,\mathcal{O},\mathsf{Eval},\mathsf{CollFind}}(y)$ executes $\mathcal{A}^{f,\mathcal{O},\mathsf{Eval},\mathsf{CollFind}}(y)$, except that whenever $\mathcal{A}$ would make a query C to $\mathsf{CollFind}$, it first samples a random $z \leftarrow \{0,1\}^t$ (where t is the input length of the circuit C),

explicitly evaluates $C^{f,\mathcal{O}}(z)$ *without invoking* Eval[17], and checks if this evaluation has any f-gate with output y. If so, $\mathcal{B}$ returns the input to this f-gate and halts; otherwise, $\mathcal{B}$ continues the execution of $\mathcal{A}$.

It was already noted in [AS15] that $\mathcal{B}$ makes at most $3q(n)^3$ queries to f, and at most $q(n)$ queries to Eval and $\mathsf{CollFind}$, respectively. To prove the desired inequality, we define $\mathsf{Guess}_{y,\mathcal{B}}$ to be the event that $\mathcal{B}$ successfully inverts y in one of its z-experiments as described above. Then, we see that

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge (\mathsf{Guess}_{y,\mathcal{B}} \vee \mathsf{CollHit}_{y,\mathcal{B}})] \geq \Pr[\textsc{Win}_{\mathcal{A}} \wedge \mathsf{CollHit}_{y,\mathcal{A}}] \geq \frac{\epsilon}{2}.$$

This inequality follows by considering a third algorithm $\mathcal{C}$ which acts as $\mathcal{B}$ does but *does not halt* after any z-experiment ($\mathcal{C}$ instead entirely ignores the outcome of this experiment); it is clear that

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge (\mathsf{Guess}_{y,\mathcal{B}} \vee \mathsf{CollHit}_{y,\mathcal{B}})] \geq \Pr[\textsc{Win}_{\mathcal{C}} \wedge \mathsf{CollHit}_{y,\mathcal{C}}] = \Pr[\textsc{Win}_{\mathcal{A}} \wedge \mathsf{CollHit}_{y,\mathcal{A}}].$$

Next, we show that

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{Guess}_{y,\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \geq \frac{1}{2}\Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{CollHit}_{y,\mathcal{B}}].$$

To see this, we write

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{Guess}_{y,\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] = \sum_{i=1}^{q} \Pr[\mathsf{Guess}_i],$$

where Guess_i is the event that $\mathcal{B}$ does *not* invert y in the first $i-1$ z-experiments it runs, does *not* invert y with one of the first $i-1$ $\mathsf{CollFind}$ queries, but *does* invert y in the ith z-experiment. Similarly, we write

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{CollHit}_{y,\mathcal{B}}] \leq \sum_{i=1}^{q} [\mathsf{CollHit}_i],$$

where $\mathsf{CollHit}_i$ is the event that $\mathcal{B}$ does *not* invert y in the first $i-1$ z-experiments it runs, does *not* invert y with one of the first $i-1$ $\mathsf{CollFind}$ queries, but *does* invert y in its ith $\mathsf{CollFind}$ query. Our claim now follows from the inqualities

$$\Pr[\mathsf{Guess}_i] \geq \frac{1}{2}\Pr[\mathsf{CollFind}_i],$$

which holds because given that no inversion has occurred in the first $i-1$ z-experiments and $\mathsf{CollFind}$ queries, the probability that the ith $\mathsf{CollFind}$ query produces (w, w') leading to a y-inversion is at most twice the probability that w (the first input) leads to a y-inversion, which is identical to the probability that the ith z-experiment leads to a y-inversion (because z and w are both just uniformly random inputs to C_i, the ith $\mathsf{CollFind}$ query).

Finally, we conclude the desired result by the calculation

$$\begin{aligned}
\Pr[\textsc{Win}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] &\geq \Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{Guess}_{y,\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \\
&\geq \frac{1}{3}\Pr[\textsc{Win}_{\mathcal{B}} \wedge (\mathsf{Guess}_{y,\mathcal{B}} \vee \mathsf{CollHit}_{y,\mathcal{B}})] \\
&\geq \frac{1}{3}\Pr[\textsc{Win}_{\mathcal{A}} \wedge \mathsf{CollHit}_{y,\mathcal{A}}] \\
&\geq \frac{\epsilon}{6}.
\end{aligned}$$

□

[17] In other words, for every query $(\hat{D}, a)$ to Eval, $\mathcal{B}$ will make exponentially many calls to $\mathcal{O}$ to brute-force recover D from $\hat{D}$, and then evaluate $D^f(a)$.

To conclude Theorem A.1, we combine Claim A.2.1 with the following additional claim from [AS15] (minimally modified).

Claim A.2.2 ([AS15], Claim 3.27)**.** *If any algorithm $\mathcal{B}$ makes at most Q queries to f,* Eval*, and* CollFind *(each), then*

$$\Pr[\text{WIN}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \leq \delta + 2^{\frac{-\delta 2^n}{3Q(n)^3}}$$

for every $\delta > 0$.

In particular, setting $\delta = 3n \cdot Q(n)^3 2^{-n}$, we see that

$$\Pr[\text{WIN}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \leq (3n \cdot Q(n)^3 + 1)2^{-n}$$

for any such $\mathcal{B}$. Using the $\mathcal{B}$ we produced from $\mathcal{A}$ in Claim A.2.1, we conclude that

$$\frac{\epsilon}{6} \leq \Pr[\text{WIN}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \leq (81n \cdot q(n)^3 + 1)2^{-n},$$

yielding the desired bound on $\epsilon = \Pr[\text{WIN}_{\mathcal{A}}]$, and hence Theorem A.1.