Electronic Colloquium on Computational Complexity, Report No. 17 (2022)

Collision-Resistance from Multi-Collision-Resistance

Ron D. Rothblum[*] Prashant Nalini Vasudevan[†]

February 15, 2022

Abstract

Collision-resistant hash functions (CRH) are a fundamental and ubiquitous cryptographic primitive. Several recent works have studied a relaxation of CRH called *t-way multi-collision-resistant hash functions* (t-MCRH). These are families of functions for which it is computationally hard to find a t-way collision, even though such collisions are abundant (and even $(t-1)$-way collisions may be easy to find). The case of $t = 2$ corresponds to standard CRH, but it is natural to study t-MCRH for larger values of t.

Multi-collision-resistance seems to be a qualitatively weaker property than standard collision-resistance. In particular, Komargodski *et al.* (Eurocrypt, 2018) showed that there does not exist a blackbox transformation of MCRH into CRH. Nevertheless, in this work we show a *non-blackbox* transformation of any moderately shrinking t-MCRH, for $t \in \{3, 4\}$, into an (infinitely often secure) CRH. This transformation is non-constructive – we can prove the existence of a CRH but cannot explicitly point out a construction.

Our result partially extends to larger values of t. In particular, we show that for suitable values of $t > t'$, we can transform a t-MCRH into a t'-MCRH, at the cost of reducing the shrinkage of the resulting hash function family and settling for infinitely often security. This result utilizes the list-decodability properties of Reed-Solomon codes.

1 Introduction

Collision-Resistant Hashing (CRH) is a fundamental primitive that is important throughout cryptography. These are functions that shrink their input but for which it is computationally infeasible to find two inputs (called "colliding" inputs) that map to the same output, even though many such pairs exist.

Recently, natural relaxations of such hash functions, called Multi-Collision-Resistant Hash Functions (t-MCRH for some integer t) have been studied [KNY17, BDRV18, BKP18, KNY18, KY18]. These are functions where it is computationally infeasible to find a set of t distinct inputs that are all mapped to the same output, even though many such collisions exist and moreover, it might even be possible to find sets of $(t-1)$ colliding inputs efficiently. Clearly, a CRH is a t-MCRH for any value of $t \geq 2$. In this paper, we address the question of whether the existence of a t-MCRH for some $t > 2$ implies the existence of a CRH.

The existing evidence in this regard is ambiguous. In some important applications like constant-round statistically hiding commitments, CRH may be replaced by MCRH [BDRV18, KNY18]. Further, MCRH imply a different relaxation of CRH called distributional CRH [KY18]. Similar to CRH, there is also a blackbox separation between MCRH and one-way permutations [BDRV18, KNY18]. These suggest that MCRH might be as powerful as CRH.

On the other hand, there is also a blackbox separation between CRH and MCRH [KNY18]. Further, CRH have properties that MCRH are not known to possess. For instance, it is well-known that for CRH shrinkage of even a single bit suffices to construct a CRH of essentially any desired shrinkage (see [Gol04, Section 6.2.3] for details). Such a transformation for t-MCRH that preserves the the number t of collisions resisted is not

[*] Technion, Israel. Email: rothblum@cs.technion.ac.il. Supported in part by the Israeli Science Foundation (Grants No. 1262/18 and 2137/19), and grants from the Technion Hiroshi Fujiwara cyber security research center and Israel cyber directorate.

[†] National University of Singapore. Email: prashant@comp.nus.edu.sg. Supported by funds from an NUS Presidential Young Professorship.

ISSN 1433-8092

known. A non-trivial transformation that somewhat increases the t is known, however, if starting with a t-MCRH that already has substantial shrinkage [BKP18].

1.1 Our Results

Loosely speaking, we show that the existence of t-MCRH for $t = 3$ or 4 that are sufficiently shrinking implies the existence of CRH. Our proof of this is non-constructive and non-blackbox. It is non-constructive because, even when given an explicit t-MCRH, we can only prove that a CRH exists but cannot explicitly point out a specific construction. It is non-blackbox because we make non-blackbox use of a potential CRH adversary.

Before stating our results formally, we define these primitives. Throughout this work, for a function $h : \{0,1\}^n \to \{0,1\}^*$, integer $t \in \mathbb{N}$ and set $X \subseteq \{0,1\}^n$, we denote by $t\text{-coll}_h(X)$ the event that (1) $|X| = t$ and (2) $h(x) = h(x')$ for every $x, x' \in X$.

Definition 1. *For functions $t = t(n)$ and $\ell = \ell(n)$, a (t,ℓ)-*$\mathsf{multi\text{-}collision\text{-}resistant\ hash\ function}$ *((t,ℓ)-*MCRH*) consists of a probabilistic polynomial-time algorithm* Gen *that on input 1^n outputs a circuit $h : \{0,1\}^n \to \{0,1\}^{n-\ell(n)}$ such that the following holds. For every family of polynomial-size circuits $A = (A_n)_{n\in\mathbb{N}}$, every polynomial p and all sufficiently large $n \in \mathbb{N}$, it holds that:*

$$\Pr_{\substack{h \leftarrow \mathsf{Gen}(1^n) \\ X \leftarrow A_n(h)}} \left[t\text{-coll}_h(X)\right] < 1/p(n). \tag{1}$$

Observe that for a (t,ℓ)-MCRH to be non-trivial, we need $\ell(n) \geq \log t(n)$. The standard definition of CRH is equivalent to $(2,1)$-MCRH. As noted earlier, while a $(2,1)$-MCRH can be used to construct a $(2,cn)$-MCRH for any $c < 1$, this is not known to be true for a $(t, \log t)$-MCRH for $t > 2$. This potentially qualitative difference between t-MCRH with different levels of shrinkage also shows up in the theorems we are able to prove in this paper. Thus, it is important to be explicit about the shrinkage ℓ of an MCRH in our terminology. (Nevertheless, in some informal discussions we may use the terminology t-MCRH without explicitly stating the shrinkage.)

Variants of MCRH. We also consider certain variants of the definition of MCRH. In an *infinitely-often* MCRH we require every adversary to fail on infinitely many n's (rather than *all* sufficiently large n's). More precisely, we say that Gen is a (t,ℓ)-ioMCRH if Eq. (1) only holds for infinitely many n's (rather than all sufficiently large n's). Every MCRH is also an ioMCRH but the converse is not necessarily true.

We say that a hash function family is *non-uniform* if the sampling algorithm is non-uniform. That is, instead of an algorithm Gen that samples the hash functions h when run as $\mathsf{Gen}(1^n)$, there is a family of probabilistic circuits $(\mathsf{Gen}_n)_{n\in\mathbb{N}}$ such that Gen_n has size $\mathsf{poly}(n)$ and outputs h. In this work, we follow the standard practice of modeling adversaries as non-uniform circuits. Jumping ahead, as some of our constructions make use of a potential non-uniform adversary, the hash functions we construct will also be non-uniform.

Remark 2. *Hsiao and Reyzin [HR04] consider a variant of* CRH *in which the adversary is given also the coins used by the generator to sample the hash function h. An analogous variant may be considered for* MCRH *(with or without the infinitely-often and non-uniform qualifiers). We remark that all of our results can be easily adapted to the [HR04] setting as well.*

Main Results. With the above definitions in hand, we are ready to state our main results. The first result, which is easiest to state, is the construction of an (infinitely-often and non-uniform) CRH from a sufficiently shrinking 3-MCRH. Similar to standard CRH, the shrinkage of non-uniform infinitely often secure CRH can also be generically increased from a single bit to cn for any $c < 1$, so we often do not specify it. The parameters stated in the theorems below result in CRH with shrinkage $\Omega(\log n)$.

Theorem 3. *Suppose there exists a $\big(3, n/2 + \omega(\log n)\big)$-*$\mathsf{MCRH}$*. Then there exists a non-uniform* ioCRH.

The same conclusion also holds under the weaker assumption that the 3-MCRH in the hypothesis above is non-uniform and/or only infinitely-often secure (this is true for the remaining theorems as well). Given Theorem 3, it is natural to wonder for what values of t we can construct ioCRH from t-MCRH. Curiously, while we are able to show such an implication from a sufficiently shrinking 4-MCRH, our techniques stop working when $t \geq 5$.

Theorem 4. *Suppose there exists a $\left(4, \frac{5}{6}n + \omega(\log n)\right)$-*MCRH*. Then there exists a non-uniform* ioCRH*.*

We discuss the limitation of our techniques to $t \leq 4$ in Section 4. Getting around this and constructing ioCRH from t-MCRH for larger constants t (let alone all constants or even super constant values of t) is an interesting open problem. Despite this restriction, for large enough constants t, we are able to show that t-MCRH generically implies t'-ioMCRH for many values of $t' < t$.

Theorem 5. *Consider any constants t, k, $t_f \geq \max\left[(2t\sqrt{k-1})^{2/3}, 24\right]$, and function $\ell = \ell(n)$. If there exists a (t, ℓ)-*MCRH* then there exists a (t_f, ℓ_f)-*ioMCRH*, for $\ell_f(n) = \min\left[(\ell(n) - n/k), (\ell(kn) - n(k-1) - O(\log n))\right]$.*

Theorem 5 is only meaningful if t is larger than t_f (and thus larger than 24). These bounds are not optimized, and our construction works for some smaller values of t and t_f as well. Starting with any (t, ℓ), the parameter k controls a tradeoff between the best values of t_f and ℓ_f we can obtain from the above theorem. It may be verified that, for the $\ell_f(n)$ above to be positive for some value of k, we need to start with an ℓ such that $\ell(n) > n/2$. With appropriate choices of the parameters, the theorem can be applied multiple times in sequence to get a (t'_f, ℓ'_f)-ioMCRH from the (t_f, ℓ_f)-ioMCRH for some $t'_f < t_f$, etc.. For $t > 4$, however, there is no sequence of parameters that can be used to get all the way down to a $(2, 1)$-ioMCRH.

For an example of an instantiation, consider a 100-MCRH that has output of length $n/10$ – that is, a $(100, 9n/10)$-MCRH. With $k = 2$, noting that $(2 \cdot 100 \cdot 1)^{2/3} \approx 34$, the above theorem gives us a $(35, 4n/10)$-ioMCRH (ignoring additive $O(\log n)$ terms). Similarly, with $k = 4$ and $k = 9$, we can get a $(50, 6n/10)$-ioMCRH and a $(69, n/10)$-ioMCRH, respectively. Values of k outside the range $[2, 9]$ lead to negative values for $\ell_f(n)$ and thus do not result in shrinking hash functions.

Following the work of Komargodski *et al.* [KNY18], it is known that none of the above implications can be proven using blackbox techniques – the proofs in their paper can be extended to show a blackbox separation between (t, cn)-ioMCRH and $(t+1, c'n)$-MCRH for any constants t and $0 < c, c' < 1$.

1.2 Our Techniques

In this overview we focus on Theorem 3, that is, our approach for constructing an (infinitely-often and non-uniform) CRH from a 3-MCRH. Suppose we have a $(3, \ell)$-MCRH, for a shrinkage parameter $\ell = \ell(n)$ to be determined below. At a high-level, we will construct two families of functions such that if neither of them is a CRH, then 3-way collisions can be found in the original hash function family.

Our approach is inspired by a recent construction of Komargodski and Yogev [KY18] of *distributional* CRH from MCRH. Distributional CRH (or DCRH), introduced by Dubrov and Ishai [DI06], are a different relaxation of CRH in which it should be hard to find *random* collisions (although it may be easy to find some specific collisions). In contrast to [KY18] who construct (infinitely-often) DCRH from MCRH, we show that MCRH imply *worst-case* collision-resistance. We defer a thorough comparison of our techniques and results with those of [KY18] to Section 1.3.1.

The candidate CRHs. Fix some input length n. Let $\mathcal{H} = \left\{h : \{0,1\}^n \to \{0,1\}^{n-\ell}\right\}$ be a $(3, \ell)$-MCRH (for simplicity we assume that the hash functions are sampled uniformly at random from this family). Since it may be possible to find 2-way collisions for functions in $\mathcal{H}$, we will have to modify $\mathcal{H}$. Toward this end we introduce an additional *non-cryptographic* function family $\mathcal{G} = \{g : \{0,1\}^n \to \{0,1\}^m\}$, with $m = m(n) < \ell(n)$. The exact properties that we need from $\mathcal{G}$, as well as setting of the parameter $m = m(n)$, will be specified below.

Thus, our first family of hash functions is $\mathcal{F} = \left\{f_{h,g} : \{0,1\}^n \to \{0,1\}^{n-\ell+m}\right\}$, where $h \in \mathcal{H}$, $g \in \mathcal{G}$. The evaluation $f_{h,g}(x)$ is simply the concatenation $f_{h,g}(x) = (h(x), g(x))$. There are two possibilities: either $\mathcal{F}$ is a CRH or it is not. If the former is true then we are done and so we might as well assume the latter. Namely, assume that there exists an efficient (non-uniform) adversary A that, given $f_{h,g} \in \mathcal{F}$ as input, outputs (x_0, x_1) such that $x_0 \neq x_1$ but $f_{h,g}(x_0) = f_{h,g}(x_1)$. For simplicity, let us assume that A is *perfect* – that is, that A finds a valid collision for *any* $f_{h,g} \in \mathcal{F}$.

We will use A – an adversary for $\mathcal{F}$ – to construct a second family of hash functions. We denote the family by $\mathcal{F}_A = \left\{f_{h,A} : \mathcal{G} \to \{0,1\}^{n-\ell}\right\}$. Each function $f_{h,A}$ takes as its input the description of a function g from $\mathcal{G}$, runs $A(f_{h,g})$ to get (x_0, x_1) – a collision for $f_{h,g}$ – and outputs $h(x_0)$. The fact that $\mathcal{F}_A$ depends on an adversary A is what makes our construction non-blackbox, non-uniform, and non-constructive. In particular, as the description of the family $\mathcal{F}_A$ involves the description of a purported adversary A for $\mathcal{F}$, unless this adversary were explicitly given, we would be unable to point out an explicit construction of $\mathcal{F}_A$ (even given $\mathcal{H}$).

What makes $\mathcal{F}_A$ interesting for our purposes is that, intuitively, a pairwise collision $g_0, g_1 \in \mathcal{G}$ for $\mathcal{F}_A$ actually specifies four inputs (namely, $(x_{00}, x_{01}) \leftarrow A(f_{h,g_0})$ and $(x_{10}, x_{11}) \leftarrow A(f_{h,g_1})$) that all collide under h. We will attempt to leverage this fact to argue that $\mathcal{F}_A$ must be collision-resistant.

Thus, assume toward a contradiction that $\mathcal{F}_A$ is not a CRH. That is, that there exists an efficient adversary A' that finds collisions for $\mathcal{F}_A$. We assume again that A' is also perfect in the same manner as A, and show how to use A' to find a 3-way collision for $\mathcal{H}$.

Finding 3-way collisions. For any $h \in \mathcal{H}$, given A and A' as above, we can find a collision for h as follows:

1. Run $A'(f_{h,A})$ to get (g_0, g_1).
2. Run $A(f_{h,g_0})$ to get (x_{00}, x_{01}).
3. Run $A(f_{h,g_1})$ to get (x_{10}, x_{11}).
4. Identify three distinct elements among $\{x_{00}, x_{01}, x_{10}, x_{11}\}$ and output them if they exist.

We make the following observations about this procedure:

1. The fact that A finds valid collisions implies that $x_{00} \neq x_{01}$ and $x_{10} \neq x_{11}$.
2. The fact that g_0 and g_1 are a collision for $f_{h,A}$ implies that, whether given f_{h,g_0} or f_{h,g_1} as input, A will find collisions that have the same output under h – that is, $h(x_{00}) = h(x_{01}) = h(x_{10}) = h(x_{11})$.
3. The definition of f_{h,g_0} and f_{h,g_1} implies that $g_0(x_{00}) = g_0(x_{01})$ and $g_1(x_{10}) = g_1(x_{11})$. Further, the fact that A' finds valid collisions implies that $g_0 \neq g_1$.

Property 2 above implies that the set $X = \{x_{00}, x_{01}, x_{10}, x_{11}\}$ forms a collision under h, while Property 1 implies that X contains at least 2 distinct elements. Unfortunately though, nothing so far guarantees that this set contains *more* than 2 elements. A particularly alarming, but so far possible, scenario is that $x_{00} = x_{10}$ and $x_{01} = x_{11}$. Thus, it is not at all immediate that the set X contains a 3-way collision. This is the point where we will need to use special properties of the family of functions $\mathcal{G}$. In particular, we will choose $\mathcal{G}$ in such a way that Property 3 above will ensure that X does indeed contain a 3-way collision for h.

The family $\mathcal{G}$. Let $\mathbb{F}$ denote the finite field of size $2^{n/2}$. Functions in $\mathcal{G}$ correspond to elements of $\mathbb{F}$. Thus, for each $\alpha \in \mathbb{F}$, there is a function $g_\alpha \in \mathcal{G}$, which is computed as follows. Given input $x \in \{0,1\}^n$, divide x into two halves $x_L, x_R \in \{0,1\}^{n/2}$ and interpret them as elements of $\mathbb{F}$ in the natural way. The evaluation of $g_\alpha(x)$ is simply the value of the line specified by (x_L, x_R) at the point α – that is, $g_\alpha(x) = x_L + \alpha \cdot x_R$ (computations performed over $\mathbb{F}$).

If for some $x_0, x_1 \in \{0,1\}^n$ and some $g_\alpha \in \mathcal{G}$ we have $g_\alpha(x_0) = g_\alpha(x_1)$, this implies that the lines specified by x_0 and x_1 intersect at $(\alpha, g_\alpha(x_0))$. Since any two distinct lines can intersect at at most one point, $\mathcal{G}$ has the following property: for any two distinct $x_0, x_1 \in \{0,1\}^n$, there is at most one function $g \in \mathcal{G}$ such that $g(x_0) = g(x_1)$.

Consider now the two pairwise collisions that we have: both $\{x_{00}, x_{01}\}$ and $\{x_{10}, x_{11}\}$ are pairs of distinct inputs such that $g_0(x_{00}) = g_0(x_{01})$ and $g_1(x_{10}) = g_1(x_{11})$. Suppose that these two sets are identical to one another: for example that $x_{00} = x_{10}$ and $x_{01} = x_{11}$. Since $g_0 \neq g_1$, this implies that there are two distinct functions in G such that x_{00} and x_{01} collide on them, a contradiction of the above property of G.

Thus, these two sets cannot be identical, implying that the set of collisions X above contains at least 3 distinct elements. This gives us a 3-way collision for h. We conclude that if $\mathcal{H}$ is a 3-MCRH, then either A or A' cannot exist. That is, either $\mathcal{F}$ is collision-resistant, or $\mathcal{F}_A$, constructed using the corresponding adversary A, is collision-resistant.

Shrinkage. It remains to argue that both $\mathcal{F}$ and $\mathcal{F}_A$ are in fact shrinking. As noted earlier, a CRH with one bit of shrinkage is sufficient to construct a CRH with essentially any desired shrinkage (and the same holds for non-uniform ioCRH). So it would be sufficient for $\mathcal{F}$ and $\mathcal{F}_A$ to shrink by even one bit.

By construction, functions in $\mathcal{G}$ map n-bit inputs to $n/2$-bit outputs. This means that $\mathcal{F}$ maps n bits to $(\frac{3}{2}n - \ell)$ bits and is shrinking as long as $\ell > n/2$. As noted above, each member of $\mathcal{G}$ is described by an element of $\mathbb{F}$, in other words a string of length $n/2$. Thus, functions in $\mathcal{F}_A$ map $n/2$ bits to $(n - \ell)$ bits. So again, if $\ell > n/2$, this is shrinking.

Coping with imperfect adversaries. Above, we assumed that the adversaries A and A' work perfectly – given a hash function, they always find a collision for it. This was done for simplicity of presentation here. In the actual construction, there are several difficulties that arise from dealing with imperfect adversaries. First, if A and A' are standard CRH adversaries, this would only imply that they find collisions for an infinite set of input lengths n, rather than all large enough n. We can only make the above arguments for the set of n's for which both of them work, and this set could well be empty. This is the reason that we can only argue that $\mathcal{F}$ or $\mathcal{F}_A$ is an *infinitely often* CRH rather than a standard CRH.

In addition, in the actual construction we only know that A succeeds with non-negligible probability, rather than with probability 1 as assumed above. This means that $\mathcal{F}_A$ might only be defined for a relatively small (but non-negligible) fraction of its domain. We resolve this second difficulty by showing how, in general, to transform collision-resistant hash functions that only work on a small subset of their domain, to full-fledged CRH. This transformation, which we find to be of independent interest, is based on the so-called "reverse randomization" technique, introduced by Lautemann [Lau83] and used in several works in cryptography since [Nao89, DNR04, DN07, BV17]. We defer the details to Section 2. We remark that this transformation introduces a small overhead and in particular leads to our hypothesis being that ℓ is larger than $n/2 + \omega(\log n)$ rather than just $n/2$ as above.

Improving Collision Resistance in General t-MCRH. A simple generalization of the above approach to getting a t_f-MCRH from a t-MCRH for some $t_f < t$ is to keep the construction as is and just change the arguments in the proof. Let $t_f = \lceil (t+1)/2 \rceil$, and let the families $\mathcal{F}$, $\mathcal{G}$, and $\mathcal{F}_A$ be just as defined above. If $\mathcal{F}$ were not a t_f-MCRH and $\mathcal{F}_A$ were not a CRH, then we can find a t-wise collision for functions in $\mathcal{H}$ in the same manner we found 3-wise collisions above – given $h \in \mathcal{H}$, find a pairwise collision (g_0, g_1) for $f_{h,A} \in \mathcal{F}_A$, and then for each g_b, find a t_f-wise collision $(x_{b1}, \ldots, x_{bt_f})$ for f_{h,g_b}. By the same argument as above, the sets $\{x_{0i}\}$ and $\{x_{1i}\}$ can have at most one element in common, and they all have the same value of $h(x_{bi})$. This gives a $(2t_f - 1)$-wise collision for h, which is a contradiction. Thus, either $\mathcal{F}$ is a t_f-MCRH or $\mathcal{F}_A$ is a CRH. The only guarantee we have, however, is that the weaker of these statements holds, meaning that a t_f-MCRH exists.

The price of this transformation is that the shrinkage of the resulting hash functions decreases by at least $n/2$ from that of $\mathcal{H}$, as this is the size of the output of functions in $\mathcal{G}$. For one, this precludes the transformation from being applied twice in order to get a t'_f-MCRH for some $t'_f < t_f$. In order to obtain better shrinkage and also to improve how much smaller t_f can be than t, we generalize our construction. For any $k \geq 2$, denote by $\mathbb{F}_k$ the finite field of size $2^{n/k}$ (assume that k divides n). Now, instead of $\mathcal{G}$ being the set of functions representing evaluations of lines in $\mathbb{F}_2$, we set it to be the functions representing evaluations of polynomials of degree $(k-1)$ over $\mathbb{F}_k$. That is, each function $g \in \mathcal{G}$ corresponds to an element $\lambda \in \mathbb{F}_k$, and given input $x \in \{0,1\}^n$, interprets it as a list of elements $x_0, \ldots, x_{k-1} \in \mathbb{F}_k$, and outputs $\sum_{i=0}^{k-1} x_i \lambda^i$.

Notice that the shrinkage of $\mathcal{F}$ is now $(\ell(n) - n/k)$, as opposed to the $(\ell(n) - n/2)$ earlier. The shrinkage of $\mathcal{F}_A$ can be computed to be $(\ell(kn) - n(k-1))$, which can be made better than $(\ell(n) - n/2)$ by an appropriate choice of k. We claim now that, for certain values of t_f, either $\mathcal{F}$ is a t_f-MCRH, or the $\mathcal{F}_A$ constructed using the corresponding adversary A is a t_f-MCRH. If they were not, given an $h \in \mathcal{H}$, we can proceed along the same lines as earlier to first get a set of functions $g_1, \ldots, g_{t_f} \in \mathcal{G}$ that collide under $\mathcal{F}_A$. Then, we can use A on each f_{h,g_i} to get t_f sets $X_i = \left\{x_{i1}, \ldots, x_{it_f}\right\}$, each of size t_f, such that all the x_{ij}'s have the same value under h and all the elements of each X_i have the same value under g_i.

If we can also prove that there are at least t distinct x_{ij}'s in the union of these sets, we would have a t-wise collision for h and thus a contradiction. Notice that each set X_i corresponds to a set of t_f polynomials (given by $x_{i1}, \ldots, x_{it_f}$) that all have the same evaluation at the field element, say λ_i, corresponding to g_i. Thus we end up with the following question: given t_f sets X_i of t_f polynomials each and t_f pairs (λ_i, y_i) with the guarantee that for each $x \in X_i$ we have $x(\lambda_i) = y_i$, what is the smallest possible number of distinct polynomials in the union $\cup_{i=1}^{t_f} X_i$?

This is closely related to bounds on the list-decodability of Reed-Solomon codes, which we use to show that as long as t_f is at least roughly $(2t\sqrt{k-1})^{2/3}$, there have to be at least t distinct elements among the above sets. This gives us a transformation from t-MCRH to t_f-MCRH for such values of t, which is again much better than the transformation to $\lceil (t+1)/2 \rceil$-MCRH that followed from our original construction. We elaborate on this in Section 3.3. By paying attention to details, we show that this transformation can be used to go from a 4-MCRH to a 3-MCRH with a loss of $n/3$ in shrinkage, and then on to a CRH with an additional loss of $n/2$. This approach, however, cannot be used to get a CRH starting from a 5-MCRH. We discuss this barrier in Section 4.

1.3 Related Work

Multi-Collision-Resistance was first studied by Joux [Jou04], who showed that for a certain class of hash functions called iterated hash functions, certain collision-finding attacks can be augmented to find multi-collisions without much overhead. Subsequent work has studied similar attacks on some other specific classes of hash functions [NS07, YW07, . . .]. The formal theoretical study of MCRH began with the work of Komargodski et al [KNY17], who defined MCRH and showed connections to problems arising from Ramsey theory.

A more detailed study of MCRH was done later in three concurrent and independent works [BKP18, KNY18, BDRV18]. Berman *et al.* [BDRV18] showed that $(n^2, \sqrt{n})$-MCRH can be constructed from the hardness of a variant of the Entropy Approximation problem [DGRV11]. Both Berman *et al.* and Komargodski *et al.* [BDRV18, KNY18] showed that constant-round statistically hiding commitment schemes can be constructed from MCRH with various parameters, which implies a blackbox separation between such MCRH and one-way permutations [HHRS15]. Komargodski *et al.* also showed two other results. First, they showed how to use MCRH to construct succinct argument-systems. Second, they showed a blackbox separation between CRH and $(3, n/2)$-MCRH. These separations extend the well-known separation between CRH and one-way permutations [Sim98].

Bitansky *et al.* [BKP18] studied MCRH and also considered a keyless version of MCRH. They used both variants to construct round-efficient succinct zero-knowledge arguments. Notably, they use the keyless version of MCRH to construct 3-message zero-knowledge arguments.

The paper closest to ours is that of Komargodski and Yogev [KY18] on distributional CRH (DCRH). DCRH, first defined by Dubrov and Ishai [DI06], is a relaxation of CRH where the adversary's task is to sample a *random* collision – given a function h, to sample (x, x') where x is a uniformly random input and x' is uniformly random conditioned on $h(x) = h(x')$. Whereas with some primitives like one-way functions the distributional version implies the full-fledged one [IL89], this is not known to be the case with CRH. See also Bitansky *et al.* [BHKY19] for more recent work on DCRH.

1.3.1 Detailed Comparison with [KY18]

Komargodski and Yogev show that the existence of a $(t,\Omega(n))$-MCRH for any constant t implies the existence of an infinitely often DCRH.[1] Their construction is also non-explicit and non-blackbox, and their approach is quite similar to ours. Our results are technically incomparable – they obtain a weaker primitive (DCRH as opposed to our CRH), but they can work with any t-MCRH, whereas we are limited to 4-MCRH. We describe their approach at a high level here and discuss the salient differences.

Let $\mathcal{H} = \left\{h : \{0,1\}^n \to \{0,1\}^{n/2}\right\}$ be a $(3, n/2)$-MCRH. They also construct two families of hash functions such that at least one of them has to be a DCRH. The first family is $\mathcal{H}$ itself. Suppose $\mathcal{H}$ is not a DCRH and there is an adversary A that samples uniformly random collisions for $h \in \mathcal{H}$. Note that A is necessarily randomized. Without loss of generality (by padding), we can assume that the number ρ of random bits that A uses is larger than n. The second family of hash functions is then defined as $\mathcal{H}_A = \left\{f_{h,A} : \{0,1\}^\rho \to \{0,1\}^{n/2}\right\}$, where $h \in \mathcal{H}$. The function $f_{h,A}(r)$ is computed by first running $A(h;r)$ to get a collision (x_0, x_1), and then outputting $h(x_0)$.

If $\mathcal{H}_A$ is also not a DCRH, then there is another adversary A' that finds random collisions for $f_{h,A} \in \mathcal{H}_A$. This A' can be used to find a pair of uniformly random (r_0, r_1) such that $A(h;r_0)$ and $A(h;r_1)$ both find collisions that have the same output under h. That is, if $(x_{00}, x_{01}) \leftarrow A(h;r_0)$ and $(x_{10}, x_{11}) \leftarrow A(h;r_1)$, then $h(x_{00}) = h(x_{01}) = h(x_{10}) = h(x_{11})$. Further, as r_0 and r_1 are uniformly random upto this condition, and A also samples uniformly random collisions, this set of x's is also random conditioned on colliding under h. Thus, with very high probability, they will all be distinct, giving a 3-way collision for h.

Essentially, the work of our family of functions $\mathcal{G}$ is here performed by the randomness of the distributional collision-finding adversary A. Such a distributional adversary is much more powerful than the normal collision-finding adversary that we have access to. The distinctness of the collisions found comes for free with a distributional adversary, whereas we have to use $\mathcal{G}$ to get it. It also enables the constructed DCRH above to not lose any shrinkage compared to the original 3-MCRH. This allows them to start from $(t,\Omega(n))$-MCRH for any constant t and iteratively perform the above process to eventually get a DCRH, while the best we can do is start from a $(4, 5n/6)$-MCRH.

1.4 Open Questions

We show using non-blackbox techniques that CRH exist assuming the existence of sufficiently shrinking 3-MCRH (or 4-MCRH). This indicates that blackbox separations are not necessarily the last word in classifying the power of cryptographic primitives. Still, our proof is non-constructive. The question that follows immediately from this observation is whether an explicit construction of CRH from MCRH is possible.

Question 1. *Can* explicit CRH *(or even* ioCRH*) be constructed from* 3-MCRH*?*

The answer to this question is unclear to us. If it were positive, such a construction, apart from being useful in obtaining explicit and usable CRH, would likely require novel and interesting techniques. It will likely have to be non-blackbox (necessarily so if it can work with cn shrinkage for $c < 1$ [KNY18]), and likely non-blackbox in a manner that is quite different from the techniques in this paper and in [KY18]. Both these papers use a potential adversary to construct hash functions, but they are agnostic to how exactly the hash function or the adversary work. An explicit construction cannot make use of any such adversary, and will perhaps involve looking into the circuit that computes the MCRH and deriving something from there.

The other direction in which our results can be improved is constructing primitives that are secure in the standard cryptographic sense rather than only infinitely often secure. Infinitely often security (or hardness) comes up regularly in cryptography and complexity theory, and we are not aware of any techniques to convert such security to standard security without additional assumptions. Being able to construct such primitives is also likely to require new and interesting techniques.

[1] Their paper states this theorem for $(t, n/2)$-MCRH, but their proof immediately extends to any $(t,\Omega(n))$-MCRH. They define MCRH security as holding only against uniform adversaries and thus obtain a uniform i.o. DCRH secure against uniform adversaries. Under our definition of MCRH with security against non-uniform adversaries, their approach would also result in a non-uniform construction secure against non-uniform adversaries. They also construct DCRH from the average-case hardness of problems in SZK, but this result is not relevant here.

Question 2. *Can a standard (as opposed to i.o.)* CRH *be constructed from a* 3-MCRH*?*

The third obvious question arising from our work is to construct a CRH from t-MCRH for $t > 4$, even assuming the best possible shrinkage. As discussed in Section 4, our approach itself is not sufficient for this purpose and new techniques, or at least non-trivial modifications to ours, will be needed here.

Question 3. *Can* CRH *be constructed from* $(t, n - \mathsf{polylog}(n))$-$\mathsf{MCRH}$ *for all constant* t*?*

Apart from these, there are several adjacent questions about the primitives we deal with here. As noted above, Berman et al [BDRV18] construct n^2-MCRH from assumptions about problems related to the complexity class SZK. Their construction does not extend to t-MCRH for constant t, and it would be interesting to see whether something like this is possible.

Question 4. *Can* t-MCRH *for some constant* t *be constructed based on the average-case hardness of the Entropy Approximation problem (or the variant used by [BDRV18])?*

Perhaps the most intriguing question is whether the classic separation of CRH from one-way permutations [Sim98] can be side-stepped using non-blackbox techniques such as those in this paper. Even a non-constructive answer to this question would be pivotal to our understanding of the relative power of these key cryptographic primitives.

Question 5. *Can similar non-blackbox techniques be used to construct* CRH *(or even* MCRH*) from One-Way Permutations?*

1.5 Organization

In Section 2 we define *partial domain* MCRH (resp., CRH) and show how to transform such hash functions to standard, full domain MCRH (resp., CRH). This notion, and the transformation, are important for our main results – the transformations from t-MCRH to t_f-MCRH for suitable $t_f < t$, which are presented in Section 3. Finally, in Section 4 we show some inherent barriers to our approach.

2 Partial Domain MCRH

In this section we introduce and study *partial-domain* MCRH. Loosely speaking, these are MCRH defined over only a (potentially small) part of their domain. The main result shown in this section is a transformation from such partial-domain MCRH to full-fledged MCRH – a transformation that will be used to establish our main theorems in Section 3. We remark that an impatient reader can skip directly to Section 3 after reviewing only the definition of partial-domain MCRH.

A *partial domain* MCRH $\mathcal{H} = (\mathcal{H}_n)_{n\in\mathbb{N}}$ is defined similarly to an MCRH except that for every $h \leftarrow \mathsf{Gen}(1^n)$, some of the inputs in the domain of h may be defined as "invalid". On such invalid inputs the hash function outputs $h(x) = \perp$. A collision-finding adversary for such a partial domain MCRH needs to find a tuple of *valid* colliding inputs. We require that the number of valid inputs is a noticeable fraction of the domain. We proceed to the formal definition.

Definition 6. *A* partial-domain (t, ℓ)-MCRH *consists of a probabilistic polynomial-time algorithm* Gen *that on input* 1^n *outputs a circuit* $h : \{0,1\}^n \to (\{0,1\}^{n-\ell} \cup \{\perp\})$ *such that the following holds.*

1. *For every family of polynomial-size circuits* $A = (A_n)_{n\in\mathbb{N}}$*, every polynomial* p *and all sufficiently large* $n \in \mathbb{N}$ *it holds that:*

$$\Pr_{\substack{h\leftarrow\mathsf{Gen}(1^n)\\ X\leftarrow A_n(h)}}\Big[\big(t\text{-coll}_h(X)\big) \text{ and } \big(\forall i \in [t],\ h(x_i) \neq \perp\big)\Big] < 1/p(n). \tag{2}$$

2. *There exists a polynomial* q *such that with all but negligible probability over* $h \leftarrow \mathsf{Gen}(1^n)$ *it holds that* $\big|\{x \in \{0,1\}^n : h(x) \neq \perp\}\big| \geq \frac{1}{q(n)} \cdot 2^n$.

To highlight the distinction from partial domain MCRH, we will sometimes refer to a standard MCRH as a *full domain* MCRH. We also generalize the definition of partial domain to the case of infinitely often MCRH and non-uniform MCRH in the natural way. We emphasize that the extension of Definition 6 to the infinitely often case requires Condition 1 to hold infinitely often, whereas Condition 2 remains unchanged – that is, it should hold for all sufficiently large n.

The following lemma shows how to transform a partial domain MCRH to a full domain MCRH. The proof technique is based on Lautemann's [Lau83] proof that BPP is contained in the polynomial hierarchy (this technique has been used in several works in cryptography since then [Nao89, DNR04, DN07, BV17]).

Lemma 7. *If there exists a partial domain (t, ℓ)-MCRH, then there exists a full domain $(t, \ell - O(\log(n)))$-MCRH. The same is true if both the initial and resulting MCRH are non-uniform and/or merely ioMCRH.*

Proof of Lemma 7. We prove the lemma with respect to standard MCRH. The proof extends readily also to non-uniform and/or ioMCRH.

Let Gen be the sampling algorithm for a partial domain (t, ℓ)-MCRH and let $q = q(n)$ be the polynomial guaranteed in the definition (i.e., for all but a negligible fraction of hash functions at least $2^n/q(n)$ of the inputs are valid). We construct a new *full domain* hash function family using a sampling algorithm Gen' as follows.

On input 1^n, the algorithm Gen' first invokes $\mathsf{Gen}(1^n)$ to obtain a hash function $h : \{0,1\}^n \to \{0,1\}^{n-\ell}$. The algorithm further samples $z_1, \dots, z_k \in \{0,1\}^n$, where $k = 2n \cdot q(n)$. The algorithm constructs a hash function h' that on input x, outputs $h'(x) = \big(h(x \oplus z_i), i\big) \in \{0,1\}^{n-\ell} \times \{0, \dots, k\}$, were i is the minimal index such that $h(x \oplus z_i) \neq \bot$ and in case no such i exists it outputs a default value $(0, 0)$. We will sometimes denote the hash function by $h' = (h, z_1, \dots, z_k)$ and note that $h' : \{0,1\}^n \to \{0,1\}^{n-\ell+O(\log n)}$.

Denote the subset of hash functions in the support of $\mathsf{Gen}(1^n)$ for which at least $1/q(n)$ fraction of the inputs are valid by H. By definition of partial domain MCRH we have that:

Claim 7.1. $\Pr_{h \leftarrow \mathsf{Gen}(1^n)}[h \notin H] = \mathrm{negl}(n)$.

Next we argue that for $h \in H$, with overwhelming probability over the z_i's, no input for the hash function $h' = (h, z_1, \dots, z_k)$ is mapped to the default value.

Claim 7.2. *For every $h \in H$, with all but 2^{-n} probability over $z_1, \dots, z_k$, no input for the hash function $h' = (h, z_1, \dots, z_k)$ is mapped to the default value.*

Proof. For every fixed $x \in \{0,1\}^n$ and every $i \in [k]$, the probability over z_i that $h(x \oplus z_i) = \bot$ is at most $1 - 1/q(n)$. Therefore, the probability that $h(x \oplus z_i) = \bot$ for *all* $i \in [k]$ is at most $(1 - 1/q(n))^{2n \cdot q(n)} \leq 2^{-2n}$. The claim follows by taking a union bound over all $x \in \{0,1\}^n$. □

Consider $h' = (h, z_1, \dots, z_k) \in H$ such that no input is mapped to the default value. In such a case, every t-way collision $\{x_1, \dots, x_t\}$ for h' must satisfy that $h(x_1 \oplus z_i) = h(x_2 \oplus z_i) = \dots = h(x_t \oplus z_i)$ for some $i \in [k]$. Thus, we have a t-way collision $\{x_1 \oplus z_i, \dots, x_t \oplus z_i\}$ of size t also for h.

Applying Claims 7.1 and 7.2, we conclude that a collision finding algorithm wrt Gen', which succeeds with probability $\epsilon = \epsilon(n)$, yields a collision finding algorithm for Gen that succeeds with probability $\epsilon(n) - \mathrm{negl}(n) - 2^{-n}$ and the lemma follows. □

3 Improving Collision-Resistance in MCRH

In this section, we prove Theorems 3 to 5 (which were stated in Section 1.1). We start by setting up a common framework for the proofs of all of the theorems. The proofs of Theorems 3 to 5 will be completed in Sections 3.1 to 3.3, respectively.

Setup. Consider a constant t and a (shrinkage) function $\ell : \mathbb{N} \to \mathbb{N}$. Let t_f and k parameters that will be determined later such that $k < t_f < t$. Define the function $\ell_f(n) = \min\left[\ell(n) - n/k, \ell(kn) - n(k-1)\right]$. Let Gen be (a sampler for) a (t, ℓ)-ioMCRH. We will use Gen to construct a (t_f, ℓ_f)-ioMCRH.[2] Below, when it is clear from the context, we sometimes use ℓ as a shorthand for $\ell(n)$. For simplicity, we will assume that k, whatever it is set to, divides n; our proof can be easily extended to work when this is not the case.

Let $\mathbb{F}$ be the finite field of size $2^{n/k}$.[3] We view an input $x \in \{0,1\}^n$ for a hash function $h \leftarrow \mathsf{Gen}(1^n)$ as representing a degree $(k-1)$ univariate polynomial over $\mathbb{F}$ as follows: x is interpreted as a vector $(x_0, \dots, x_{k-1}) \in \mathbb{F}^k$, and the polynomial is defined as $P_x(\xi) = \sum_{i=0}^{k-1} x_i \cdot \xi^i$ (where the arithmetic is over the field). For ease of notation, for $\lambda \in \mathbb{F}$, we use $x(\lambda)$ to denote the evaluation of the polynomial P_x at the point λ.

The First Hash Family. We construct a new hash function family defined by the sampler Gen' that, on input 1^n, works as follows:

1. Invoke $\mathsf{Gen}(1^n)$ to obtain a hash function $h : \{0,1\}^n \to \{0,1\}^{n-\ell}$.
2. Sample a random $\lambda \in \mathbb{F}$.
3. Output the hash function[4] $h' : \{0,1\}^n \to \{0,1\}^{n-\ell+n/k}$ defined as $h'(x) = \big(h(x), x(\lambda)\big)$.

If Gen' is a (t_f, ℓ')-ioMCRH, where $\ell'(n) = (\ell(n) - n/k)$, then we are done. Thus, we may assume that it is not – namely, that there exists a polynomial-size circuit family $A' = (A'_n)_{n \in \mathbb{N}}$ and a polynomial p' such that for all sufficiently large $n \in \mathbb{N}$ it holds that:

$$\Pr_{\substack{h' \leftarrow \mathsf{Gen}'(1^n) \\ X \leftarrow A'_n(h')}} \left[t_f\text{-coll}_{h'}(X)\right] \geq \frac{1}{p'(n)}. \tag{3}$$

Using the definition of h', Eq. (3) can be rewritten as:

$$\Pr_{\substack{h \leftarrow \mathsf{Gen}(1^n) \\ \lambda \leftarrow \mathbb{F} \\ X \leftarrow A'_n(h,\lambda)}} \Big[\big(t_f\text{-coll}_h(X)\big) \text{ and } \big(\forall x_1, x_2 \in X,\ x_1(\lambda) = x_2(\lambda)\big)\Big] \geq \frac{1}{p'(n)}. \tag{4}$$

For every h in the support of $\mathsf{Gen}(1^n)$, define:

$$\delta_h = \Pr_{\substack{\lambda \leftarrow \mathbb{F} \\ X \leftarrow A'_n(h,\lambda)}} \Big[\big(t_f\text{-coll}_h(X)\big) \text{ and } \big(\forall x_1, x_2 \in X,\ x_1(\lambda) = x_2(\lambda)\big)\Big].$$

Thus, Eq. (4) implies that $\mathsf{E}_{h \leftarrow \mathsf{Gen}(1^n)}[\delta_h] \geq \frac{1}{p'(n)}$. We shall aim to restrict our attention to hash functions h for which δ_h is relatively large (i.e., close to the expectation). The following lemma describes a sampling algorithm for such hash functions.

Lemma 8. *There exists a probabilistic polynomial time algorithm $\widetilde{\mathsf{Gen}}$ that on input 1^n outputs a hash function $h : \{0,1\}^n \to \{0,1\}^{n-\ell}$ in the support of $\mathsf{Gen}(1^n)$ such that the following holds for all sufficiently large n:*

- $\Pr_{h \leftarrow \widetilde{\mathsf{Gen}}(1^n)}\left[\delta_h > \frac{1}{4p'(n)}\right] = 1 - 2^{-\Omega(n)}$.

[2] Actually, it may be the case that the shrinkage of the hash function we construct is *larger* than this ℓ_f. In such a case, we can simply pad the output of the hash function with 0's to ensure that the shrinkage is exactly ℓ' (without any effect on its collision-resistance properties).

[3] We assume the field elements can be represented using $\log_2(|\mathbb{F}|)$ bits (in the natural way) and that field operations (i.e., arithmetic operations as well as sampling of random field elements) can be performed in $\mathsf{polylog}(|\mathbb{F}|)$ time. See, e.g., [Sho88] for details.

[4] Note that for Gen' to be non-trivial we must have $\ell(n) > n/k$.

- *For every event E:*

$$\Pr_{h \leftarrow \mathsf{Gen}(1^n)} \big[h \in E\big] \geq \frac{1}{3p'(n)} \cdot \Pr_{h \leftarrow \widetilde{\mathsf{Gen}}(1^n)} \big[h \in E\big] - 2^{-\Omega(n)}.$$

The first item in Lemma 8 states that with very high probability, a hash function h sampled by $\widetilde{\mathsf{Gen}}$ has relatively large δ_h. The second item relates the distributions Gen and $\widetilde{\mathsf{Gen}}$ and in particular implies that events that happen with non-negligible probability over the latter also happen with non-negligible probability over the former. The proof of Lemma 8 is deferred to Section 3.4 but on first reading, the reader may find it convenient to think of the simpler case in which all h have $\delta_h \geq \frac{1}{4p'(n)}$ in which case we can simply take $\widetilde{\mathsf{Gen}} = \mathsf{Gen}$.

The Second Hash Family. We now use the adversary A' to construct a new *partial domain non-uniform hash function family* defined by a sampler $\mathsf{Gen}'' = (\mathsf{Gen}''_n)_{n \in \mathbb{N}}$ as follows. The sampler[5] $\mathsf{Gen}''_{n/k}$ works as follows:

1. Invoke $\widetilde{\mathsf{Gen}}(1^n)$ to obtain a hash function $h : \{0,1\}^n \to \{0,1\}^{n-\ell}$.
2. Output a hash function[6] $h'' : \{0,1\}^{n/k} \to (\{0,1\}^{n-\ell}) \cup \{\perp\}$ that is computed as follows:
   - The input to h'', which is a vector in $\{0,1\}^{n/k}$, is interpreted as a field element $\lambda \in \mathbb{F}$ in the natural way (recall that $|\mathbb{F}| = 2^{n-k}$).
   - To hash λ, first invoke[7] $A'_n(h, \lambda)$ and then consider two cases:
     (a) Case 1: If $A'_n(h, \lambda)$ outputs $X \subseteq \{0,1\}^n$ such that t_f-$\mathrm{coll}_h(X)$ and $\forall x_1, x_2 \in X$, $x_1(\lambda) = x_2(\lambda)$. In such a case $h''(\lambda)$ outputs $h(x)$ for an arbitrary $x \in X$ (the specific choice does not matter since all elements in X collide under h).
     (b) Case 2: If $A'_n(h, \lambda)$ does not generate an output as above (which can be easily tested in polynomial-time) $h''(\lambda)$ outputs $\perp$.

Recall that we currently have two assumptions in place – Gen is a (t, ℓ)-ioMCRH and Gen' is *not* a (t_f, ℓ')-ioMCRH, with the above A' being the corresponding adversary. Under these assumptions we will prove the following lemma.

Lemma 9. Gen'' *is a partial-domain non-uniform* (t_f, ℓ'')-ioMCRH*, where* $\ell''(n) = \ell(kn) - n(k-1)$*.*

Lemma 9, for various values of t and t_f, together with with transformation of partial-domain MCRH into full-domain MCRH (Lemma 7), implies Theorems 3 to 5. To prove it, we will need to show that Gen'' satisfies the two conditions from Definition 6, and that it has shrinkage ℓ''. The latter follows by construction. We will show in Proposition 10 that Gen'' satisfies Condition 2 of Definition 6 irrespective of the choice of t_f and k. The proof that Gen'' satisfies Condition 1 is where the proofs of the three theorems diverge. For different values of t_f and k, the fact that it does is proven in Sections 3.1 to 3.3, leading to Theorems 3 to 5.

Proposition 10. *There exists a polynomial q such that, for all sufficiently large n, with all but negligible probability over $h'' \leftarrow \mathsf{Gen}''_n$, it holds that* $\big|\{x \in \{0,1\}^n : h''(x) \neq \perp\}\big| \geq \frac{1}{q(n)} \cdot 2^n$*.*

Proof. By the first item in Lemma 8, with all but $2^{-\Omega(n)}$ probability over $h \leftarrow \widetilde{\mathsf{Gen}}(1^n)$ it holds that $\delta_h \geq 1/(4p'(n))$. If $\delta_h \geq 1/(4p'(n))$ then the corresponding h'' (that is output by $\mathsf{Gen}''_{n/k}$ when it samples h from $\widetilde{\mathsf{Gen}}(1^n)$) does not output $\perp$ on an inverse polynomial fraction of its domain. Thus, Gen'' satisfies the requirements of the proposition. □

[5] For sake of consistency we define the hash function w.r.t. "security parameter" n/k, since its domain is $\{0,1\}^{n/k}$.

[6] As in Footnote 4, this is only interesting if $\ell(n) > (n - n/k)$.

[7] This is the point where we use the adversary in a non-blackbox manner. Since the adversary is non-uniform, this also makes the construction non-uniform.

3.1 From 3-MCRH to CRH ($t = 3, t_f = 2$)

In this subsection, we prove that Gen'' satisfies Condition 1 of Definition 6 under the parameter setting $t = 3$, $t_f = 2$, and $k = 2$. This is stated in the following proposition. This proves Lemma 9 under this setting, which, together with Lemma 7, completes the proof of Theorem 3.

Proposition 11. *Let $t = 3$ and $k = 2$. For every family of polynomial-size circuits $A'' = (A''_n)_{n\in\mathbb{N}}$, every polynomial p'' and infinitely many $n \in \mathbb{N}$ it holds that:*

$$\Pr_{\substack{h''\leftarrow\mathsf{Gen}''_n\\(\lambda_1,\lambda_2)\leftarrow A''_n(h'')}}\Big[\big(\lambda_1 \neq \lambda_2\big) \text{ and } \big(h''(\lambda_1) = h''(\lambda_2) \neq \bot\big)\Big] < 1/p''(n).$$

Proof. Fix a hash function $h'' \leftarrow \mathsf{Gen}''_{n/k}(1^{n/k})$ and consider a pair $\lambda_1, \lambda_2 \in \mathbb{F}$ such that $\lambda_1 \neq \lambda_2$ and $h''(\lambda_1) = h''(\lambda_2) \neq \bot$. Let $\{x_{1,1}, x_{1,2}\} = A'_n(h, \lambda_1)$ and $\{x_{2,1}, x_{2,2}\} = A'_n(h, \lambda_2)$. Recall that h'' can be recast as a function $h \leftarrow \widetilde{\mathsf{Gen}}(1^n)$.

Claim 11.1. *The set $\{x_{i,j}\}_{i,j\in\{1,2\}}$ contains a 3-way collision for h.*

Proof. Since $h''(\lambda_1) \neq \bot$ we have that $x_{1,1} \neq x_{1,2}$ but $h(x_{1,1}) = h(x_{1,2})$ and $x_{1,1}(\lambda_1) = x_{1,2}(\lambda_2)$. Similarly, since $h''(\lambda_2) \neq \bot$, we have that $x_{2,1} \neq x_{2,2}$ but $h(x_{2,1}) = h(x_{2,2})$ and $x_{2,1}(\lambda_1) = x_{2,2}(\lambda_2)$. In addition, since $h''(\lambda_1) = h''(\lambda_2)$ we have that $h(x_{1,1}) = h(x_{2,1})$. Overall, this means that $h(x_{1,1}) = h(x_{1,2}) = h(x_{2,1}) = h(x_{2,2})$ so all of the elements do indeed collide.

Thus we only need to show that the set $\{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}\}$ contains at least 3 distinct elements. Suppose that $x_{1,1} = x_{2,1}$ and $x_{1,2} = x_{2,2}$ (the other case is handled similarly). In such a case we have that the line $x_{1,1}$ and the line $x_{1,2}$, which are distinct lines, agree on the distinct points λ_1 and λ_2. But this is a contradiction since two distinct lines (i.e., degree 1 polynomials) can agree on at most one point. □

Thus, the existence of an adversary A'' contradicting the proposition's hypothesis immediately yields a method for finding a 3-way collision for a random $h \leftarrow \widetilde{\mathsf{Gen}}(1^n)$, with probability at least $1/p''(n)$, for all sufficiently large n. By the second item of Lemma 8, this method also works for $h \leftarrow \mathsf{Gen}(1^n)$ with probability at least $\frac{1}{3p'(n)\cdot p''(n)} - 2^{-\Omega(n)}$ (again, for all sufficiently large n) – a contradiction. □

3.2 From 4-MCRH to 3-MCRH ($t = 4, t_f = 3$)

Having handled the case of $t = 3$, we proceed to the special case of $t = 4$. We show how to transform a sufficiently shrinking 4-MCRH into a 3-ioMCRH. If the latter is sufficiently shrinking, we can then apply Theorem 3 to obtain an ioCRH.

Thus, we need to show that Gen'' satisfies Condition 1 of Definition 6 under the parameter setting $t = 4$, $t_f = 3$, and $k = 3$. This is stated in the following proposition. This proves Lemma 9 under this setting, which, together with Lemma 7, completes the proof of Theorem 3.

Proposition 12. *Let $t = 4$ and $k = 3$. For every family of polynomial-size circuits $A'' = (A''_n)_{n\in\mathbb{N}}$, every polynomial p'' and infinitely many $n \in \mathbb{N}$ it holds that:*

$$\Pr_{\substack{h''\leftarrow\mathsf{Gen}''_n\\(\lambda_1,\lambda_2,\lambda_3)\leftarrow A''_n(h'')}}\Big[\big(\lambda_1, \lambda_2, \lambda_3 \text{ are distinct}\big) \text{ and } \big(h''(\lambda_1) = h''(\lambda_2) = h''(\lambda_3) \neq \bot\big)\Big] < 1/p''(n).$$

As the proof mirrors that of Proposition 11, we provide only a sketch.

Proof Sketch. Similarly to Proposition 11, each λ_i yields a 3-way collision $x_{i,1}, x_{i,2}, x_{i,3}$ and the set $\{x_{i,j}\}_{i,\in\{1,2,3\}}$ all collide on h. What remains to be shown is that this set contains 4 distinct elements.

Suppose not. Then, wlog, it must be the case that $x_{1,1} = x_{2,1} = x_{3,1}$, $x_{2,1} = x_{2,2} = x_{2,3}$ and $x_{1,3}, = x_{2,3} = x_{3,3}$. Each one of $x_{1,1}, x_{1,2}, x_{1,3}$ specifies a degree $k - 1$ polynomial, that is, a quadratic polynomial. Thus, we have 3 distinct quadratic polynomials that agree on the 3 points $\lambda_1, \lambda_2, \lambda_3$ – a contradiction.

Overall, we get that a 3-way collision finder for Gen'' yields a 4-way collision finder for $\widetilde{\mathsf{Gen}}$, and therefore, as in the proof of Proposition 11, also for Gen. □

Overall, this yields a $(3, \ell_f - O(\log n))$-ioMCRH from a $(4, \ell)$-MCRH, where $\ell_f = \min[\ell(n) - n/3, \ell(3n) - 2n]$. In particular, if $\ell(n) > \frac{5}{6} \cdot n + \omega(\log n)$, we get that $\ell_f > \frac{1}{2}n + \omega(\log n)$. At this point we can apply Theorem 3 to derive a (non-uniform) ioMCRH, thereby establishing Theorem 4.

3.3 From General t-MCRH to t_f-MCRH

In this subsection, we consider a generic constant t and show that Gen'' satisfies Condition 1 of Definition 6 under the certain settings of t_f and k. This is captured by the following lemma.

Lemma 13. *Consider any t, k, and $t_f \geq \max\left[(2t\sqrt{k-1})^{2/3}, 24\right]$. For every family of polynomial-size circuits $A'' = (A''_n)_{n \in \mathbb{N}}$, every polynomial p, and infinitely many $n \in \mathbb{N}$, it holds that:*

$$\Pr_{\substack{h'' \leftarrow \mathsf{Gen}''(1^n) \\ X \leftarrow A''_n(h'')}} \left[\left(t_f\text{-coll}_{h''}(X)\right) \text{ and } \left(\forall i \in [t],\ h''(x_i) \neq \bot\right)\right] < 1/p(n). \tag{5}$$

Under the above setting of parameters, Lemma 9 follows from Lemma 13. Combined with Lemma 7 (the partial to full domain transformation), this completes the proof of Theorem 5. The proof of Lemma 13 makes use of list-decoding bounds for Reed-Solomon codes.

Proof. Assume toward a contradiction that there exists a polynomial-size circuit family $A'' = (A''_n)_{n \in \mathbb{N}}$ and a polynomial p'' such that for all sufficiently large $n \in \mathbb{N}$ it holds that:

$$\Pr_{\substack{h'' \leftarrow \mathsf{Gen}''_{n/k} \\ \Lambda \leftarrow A''_{n/k}(h'')}} \left[(t_f\text{-coll}_{h''}(\Lambda)) \text{ and } (\forall \lambda \in \Lambda : h''(\lambda) \neq \bot)\right] \geq 1/p''(n).$$

Fix a large enough n such that both $A''_{n/k}$ and A'_n have such non-negligible success probability. Fix also an h in the support of $\widetilde{\mathsf{Gen}}(1^n)$ and the corresponding h'' (that is output by $\mathsf{Gen}''_{n/k}$ when it samples h from $\widetilde{\mathsf{Gen}}(1^n)$) such that for the $\Lambda = \{\lambda_1, \ldots, \lambda_{t_f}\}$ output by $A''_{n/k}(h'')$, the conditions in the above probability statement hold. Denote $X_i = A'_n(h, \lambda_i)$.

Claim 13.1. *It holds that:*

1. *For every $i \in [t_f]$, the set X_i contains t_f distinct elements and for every $x_1, x_2 \in X_i$ it holds that $x_1(\lambda_i) = x_2(\lambda_i)$.*
2. *For every $i, j \in [t_f]$ and $x_1 \in X_i$, $x_2 \in X_j$ it holds that $h(x_1) = h(x_2)$.*

Proof. The fact that the event t_f-$\text{coll}_{h''}(\Lambda)$ holds implies that all of the λ_i's are distinct but $h''(\lambda_1) = \cdots = h''(\lambda_{t_f}) \neq \bot$. By the definition of Gen'', this means that for every $i \in [t_f]$, it holds that $A'(h, \lambda_i)$ outputs a set $X_i = \{x_{i,1}, \ldots, x_{i,t_f}\}$ such that t_f-$\text{coll}_{(h,\lambda_i)}(X_i)$. This implies Item 1 in the claim as well as the fact that $h(x_{i,j}) = h(x_{i,j'})$ for every $i, j, j' \in [t_f]$.

On the other hand, the fact that $h''(\lambda_1) = \cdots = h''(\lambda_t) \neq \bot$ means that $h(x_{1,1}) = \cdots = h(x_{t_f,1})$. Overall, we conclude that *all* of the $x_{i,j}$'s collide under h. This establishes Item 2. □

Let $X \subseteq \{0,1\}^n$ be the multi-set $X = \cup_{i \in [t_f]} X_i$. We emphasize that X is a multi-set, where the *multiplicity* of an element $x \in X$ is equal to the number of $i \in [t_f]$ such that $x \in X_i$. The following proposition shows that X contains a t-way collision for h.

Proposition 14. *t-$\text{coll}_h(X)$ holds.*

Proof. By Item 2 in Claim 13.1, all elements in the set X indeed collide under h and so we only need to show that the set contains at least t *distinct* elements. Define a function $f : \Lambda \to \mathbb{F}$ as $f(\lambda_i) = x_i(\lambda_i)$, where x_i is an arbitrary element in X_i (by Item 1 in Claim 13.1, the specific choice does not matter). Let $d = k - 1$. Let $X_{close} \subseteq X$ denote the set of points $x \in X$ such that x, viewed as a degree d polynomial over $\mathbb{F}$, agrees with f on at least $\sqrt{2t_f d}$ points in Λ. By construction, all $x \in X \backslash X_{close}$ have multiplicity at most $\sqrt{2t_f d}$.

Claim 14.1. *The number of* distinct *elements in X_{close} is at most $\sqrt{2t_f/d}$.*

This claim follows immediately from the following lemma of Sudan [Sud97], which is a special case of an earlier lemma of Goldreich *et al.* [GRS00].[8]

Lemma 15 ([Sud97, GRS00])**.** *Let $\mathbb{F}$ be a finite field and let $\{(x_i, y_i)\}_{i=1}^n \in (\mathbb{F} \times \mathbb{F})^n$ be a sequence of N pairs. The number of degree d polynomials f such that $|\{i : f(x_i) = y_i)\}| \geq \sqrt{2dN}$ is at most $\sqrt{2N/d}$.*

Thus, the multi-set X, which contains $(t_f)^2$ elements overall (counting multiplicities), has at most $\sqrt{2t_f/d}$ elements with multiplicity at least $\sqrt{2t_f d}$. This means that the number of distinct elements in X is at least:

$$\frac{(t_f)^2 - \sqrt{2t_f/d} \cdot t_f}{\sqrt{2t_f d}} \geq \frac{(t_f)^{3/2}}{2\sqrt{d}} \geq t$$

where the first inequality holds for any $t_f \geq 24$ and $d \geq 1$, and the second inequality follows from the condition in the hypothesis that $t_f \geq (2t\sqrt{k-1})^{2/3}$. □

Thus, under the assumption that such an A'' exists, we are able to find a t-way collision for a random $h \leftarrow \widetilde{\mathsf{Gen}}(1^n)$ with probability at least $1/p''(n)$ for all large enough n. By the second item of Lemma 8, this method also works for $h \leftarrow \mathsf{Gen}(1^n)$ with probability at least $\left(\frac{1}{3p'(n) \cdot p''(n)} - 2^{-\Omega(n)}\right)$ for all large enough n – a contradiction to our assumption that Gen is a (t, ℓ)-ioMCRH. So such an A'' cannot exist, which proves Lemma 13. □

3.4 Proof of Lemma 8

Consider the following basic process $\mathsf{Gen}_0(1^n)$ (this is not yet the eventual process $\widetilde{\mathsf{Gen}}$ which we need to show in order to prove Lemma 8).

$\underline{\mathsf{Gen}_0(1^n)}$:

1. Sample $h \leftarrow \mathsf{Gen}(1^n)$.
2. Sample $\lambda_1, \ldots, \lambda_\ell \leftarrow \mathbb{F}$, where $\ell = \Theta((p'(n))^2 \cdot n \cdot r(n))$ where r is a polynomial bounding the number of random coins that $\mathsf{Gen}(1^n)$ uses. Use $\lambda_1, \ldots, \lambda_\ell$ to compute an approximation $\hat{\delta}_h$ for δ_h by setting
$$\hat{\delta}_h = \frac{1}{\ell} \cdot \left|\left\{i \in [\ell] : \big(t_f\text{-coll}_h(X)\big) \text{ and } \big(\forall x_1, x_2 \in X,\ x_1(\lambda_i) = x_2(\lambda_i)\big), \text{ where } X \leftarrow A'_n(h, \lambda_i)\right\}\right|.$$
3. If $\hat{\delta}_h > 1/(3p'(n))$ output h otherwise output $\perp$.

Denote by $p_\perp = \Pr[\mathsf{Gen}_0(1^n) = \perp]$. Let μ denote the distribution obtained by sampling from $\mathsf{Gen}_0(1^n)$ conditioned on *not* getting $\perp$.

Proposition 16. $p_\perp \leq 1 - 1/(3p'(n))$.

[8] Sudan additionally established bounds on the *algorithmic* list-decoding properties of Reed-Solomon codes, whereas for our purposes a combinatorial bound (such as that established in [GRS00]) suffices.

Proof. Since $E_{h\leftarrow \mathsf{Gen}(1^n)}[\delta_h] \geq 1/p'(n)$ (see Eq. (4)), by Markov's inequality, with probability $1/2p'(n)$ over $h \leftarrow \mathsf{Gen}(1^n)$ it holds that $\delta_h \geq 1/(2p'(n))$.

Assume that such an h is sampled in Step 1 of $\mathsf{Gen}_0(1^n)$. By the Chernoff bound, the probability that it passes the check in Step 2 is at least 0.99. In case these two events occur the process outputs $h \neq \perp$ and so we have that $p_\perp \leq 1 - 1/(3p'(n))$. □

Proposition 17. *For every event E it holds that:*

$$\Pr_{h\leftarrow \mathsf{Gen}(1^n)}[h \in E] \geq (1 - p_\perp) \cdot \Pr_{h\leftarrow \mu}[h \in E].$$

Proof. By linearity, it suffices to prove the claim for the case that $E = \{h\}$ is a singleton. Furthermore, we can view the distribution μ as sampling from $\mathsf{Gen}_0(1^n)$ repeatedly until a function $h \neq \perp$ is obtained. With that in mind we have that

$$\begin{aligned}\Pr[\mu = h] &= \sum_{i=0}^{\infty} \Pr[\mu \text{ outputs } h \text{ in iteration } i+1 \text{ and } \perp \text{ in all previous iterations}]\\ &= \sum_{i=0}^{\infty} \Pr[\mathsf{Gen}_0(1^n) = h] \cdot (p_\perp)^i\\ &\leq \Pr[\mathsf{Gen}(1^n) = h] \cdot \frac{1}{1-p_\perp},\end{aligned}$$

where the final inequality follows from the fact that $\Pr[\mathsf{Gen}_0(1^n) = h] \leq \Pr[\mathsf{Gen}(1^n) = h]$ and a standard bound on the sum of a geometric series. □

Consider the "rejection sampling with cutoff" sampler $\widetilde{\mathsf{Gen}}(1^n)$ defined as follows:

1. Repeat $\Theta(p'(n) \cdot n)$ times:
   (a) Sample $h \leftarrow \mathsf{Gen}_0(1^n)$.
   (b) If $h \neq \perp$ output h and abort. Otherwise continue to the next iteration.
2. If this step has been reached, then output some default hash function in the support of $\mathsf{Gen}(1^n)$.

Note that $\widetilde{\mathsf{Gen}}$ can indeed be implemented in probabilistic polynomial-time.

Proposition 18. *The statistical distance between μ and $\widetilde{\mathsf{Gen}}(1^n)$ is at most $2^{-\Omega(n)}$.*

Proof. The statistical distance between the two distributions is equal to the probability that $\widetilde{\mathsf{Gen}}$ gets to Step 2. It follows from Proposition 16 that the latter probability is bounded by $(1 - 1/(3p'(n)))^{\Omega(p'(n)\cdot n)} \leq 2^{-\Omega(n)}$. □

Combining Propositions 16 to 18 we have that for every event E,

$$\begin{aligned}\Pr_{h\leftarrow \mathsf{Gen}(1^n)}[h \in E] &\geq (1 - p_\perp) \cdot \Pr_{h\leftarrow \mu}[h \in E]\\ &\geq \frac{1}{3p'(n)} \cdot \Pr_{h\leftarrow \mu}[h \in E]\\ &\geq \frac{1}{3p'(n)} \cdot \Pr_{h\leftarrow \widetilde{\mathsf{Gen}}(1^n)}[h \in E] - 2^{-\Omega(n)}. \qquad (6)\end{aligned}$$

This establishes the second part of Lemma 8. The following proposition establishes also the first part.

Proposition 19. $\Pr_{h\leftarrow \widetilde{\mathsf{Gen}}(1^n)}\left[\delta_h < \frac{1}{4p'(n)}\right] = 2^{-\Omega(n)}$.

Proof. Fix h with $\delta_h < \frac{1}{4p'(n)}$. For $\mathsf{Gen}_0(1^n)$ to output h, the approximation must deviate by at least an $\frac{1}{12p'(n)}$ factor which, by the Chernoff bound, happens with probability at most $2^{-(2n+p'(n)+r(n))}$.

By taking a union bound over the $O(p'(n) \cdot n)$ iterations in $\widetilde{\mathsf{Gen}}(1^n)$, the probability that an h as above is sampled by the rejection sampling process is at most $\frac{O(p'(n)\cdot n)}{2^{2n+p'(n)+r(n)}} \leq 2^{-(n+r(n))}$. By another application of the union bound we have that:

$$\Pr_{h \leftarrow \widetilde{\mathsf{Gen}}(1^n)}\left[\delta_h < \frac{1}{4p'(n)}\right] = \sum_{h\,:\,\delta_h < \frac{1}{4p'(n)}} \Pr[\widetilde{\mathsf{Gen}}(1^n) = h] \leq 2^{r(n)} \cdot 2^{-(n+r(n))} = 2^{-n}.$$

□

Lemma 8 follows from Eq. (6) and Proposition 19.

4 Limitations of Our Approach

In this section, we discuss why our approach to constructing a CRH (more precisely a non-uniform ioCRH) cannot work when starting from a t-MCRH for $t > 4$. Our discussion will not be completely formal, but should convince the reader of this claim. We will consider, in fact, a generalization of the construction presented in previous sections that uses an unspecified (list-decodable) code rather than the Reed-Solomon code. For simplicity, we go back some of the assumptions made in the presentation in Section 1.2 – that we start with a (t, ℓ)-MCRH that simply samples uniformly random functions from a set $\mathcal{H} = \left\{h : \{0,1\}^n \to \{0,1\}^{n-\ell}\right\}$, and that all collision-finding adversaries below are perfect. Say we wish to construct from this a (t_f, ℓ_f)-ioMCRH for some $t_f \leq t$.

Formalizing our approach. The generalized version of our construction may be described as follows. Let C be a code with message length of n bits and codewords of length N over an alphabet Σ. In particular, C is a subset of Σ^N of size 2^n. (The constructions in Section 3 correspond to taking C to be the Reed-Solomon code of various degrees over fields of characteristic 2.) We will also write $C(x)$ for an $x \in \{0,1\}^n$ to denote the codeword that x is mapped to by the code. Our construction defines the following families of functions:

- $\mathcal{G} = \{g_\lambda : \{0,1\}^n \to \Sigma\}_{\lambda \in [N]}$: for any $x \in \{0,1\}^n$ and $\lambda \in [N]$, $g_\lambda(x)$ is the λ^{th} symbol of $C(x)$.
- $\mathcal{F} = \left\{f_{h,g} : \{0,1\}^n \to \{0,1\}^{n-\ell} \times \Sigma\right\}_{h\in\mathcal{H}, g\in\mathcal{G}}$: $f_{h,g}(x)$ is simply the concatenation $(h(x), g(x))$. Suppose $\mathcal{F}$ is not a t_f-ioMCRH, and the corresponding adversary is A.
- $\mathcal{F}_A = \left\{f_{h,A} : [N] \to \{0,1\}^{n-\ell}\right\}_{h\in\mathcal{H}}$: given input $\lambda \in [N]$, the function $f_{h,A}$ first runs $A(h, g_\lambda)$ to get $x_1, \ldots, x_{t_f} \in \{0,1\}^n$, and outputs $h(x_1)$. (Here g_λ is the function corresponding to λ in $\mathcal{G}$.)

We would like to show then that if $\mathcal{F}$ is not a t_f-ioMCRH and $\mathcal{F}_A$ constructed using the adversary A is also not a t_f-ioMCRH, then we can find t-wise collisions for functions in $\mathcal{H}$, which is a contradiction. In order to do this, we make use of the collision-finding adversary A' for $\mathcal{F}_A$. The process then proceeds as follows:

1. Given an $h \in \mathcal{H}$, first run $A'(f_{h,A})$ to get functions $g_1, \ldots, g_{t_f} \in \mathcal{G}$ that collide under $f_{h,A}$.
2. Then, for each g_i, run $A(f_{h,g_i})$ to get a set $X_i = \left\{x_{i1}, \ldots, x_{it_f}\right\}$ whose elements collide under f_{h,g_i}.
3. If there are t distinct elements in the union $\cup_{i=1}^{t_f} X_i$, output them.

Arguments outlined in Section 1.2 and Section 3 explain why all the x_{ij}'s have the same output under h, and only the following question remains: can we ensure that there are indeed t distinct elements among the X_i's while $\mathcal{F}$ and $\mathcal{F}_A$ are both shrinking? Note that the shrinkage of $\mathcal{F}$ is $(\ell - \log|\Sigma|)$, and that of $\mathcal{F}_A$ is $(\log N - (n - \ell))$.

The question of the existence of t distinct x_{ij}'s may be recast as follows. We are given t_f sets of codewords $C_i = \left\{c_{i1}, \ldots, c_{it_f}\right\}$, where each of the t_f codewords in C_i are distinct. Each C_i corresponds to a statement that, for some $\lambda_i \in [N]$ (where the λ_i's are distinct), all the codewords in C_i agree on the λ_i^{th} coordinate. In other words, there are t_f tuples $(\lambda_i, y_i) \in [N] \times \Sigma$ such that for all $c_{ij} \in C_i$, we have $c_{ij}[\lambda_i] = y_i$. We would then like to claim that there is no set of codewords $T \subseteq C$ such that $|T| < t$, for each i we have $C_i \subseteq T$, and still $c_{ij}[\lambda_i] = y_i$ for all $i, j \in [t_f]$. At the very least, this requires that no set of $(t-1)$ codewords agree on t_f coordinates.

Optimality of current choices. It turns out, however, that (an extension of) the Singleton bound implies that in order for this to happen for $t_f < t$, the alphabet Σ has to be quite large, thus implying an upper bound on the shrinkage of the resulting family $\mathcal{F}$. Let us start with the simple case of $t = 3$ and $t_f = 2$. Here, the condition stated above becomes the following: any 2 codewords agree on at most 1 coordinate. In other words, the distance of the code has to be at least $(N-1)$.

Proposition 20. *In any code $C \subseteq \Sigma^N$ where $|C| = 2^n$ and any 2 codewords agree on at most 1 coordinate, it has to be that $|\Sigma| \geq 2^{n/2}$.*

Proof. This is simply the Singleton bound. Consider truncating all the codewords in C to the first two coordinates. As no two codewords agree on more than one coordinate, this set of truncated codewords still has no repetitions and so has size at least 2^n. This implies that $|\Sigma|^2 \geq 2^n$, which implies that $|\Sigma| \geq 2^{n/2}$. □

Proposition 20 implies that the shrinkage of $\mathcal{F}$ is $(\ell - \log|\Sigma|) \leq (\ell - n/2)$. In particular, this says that using a different code in place of the Reed-Solomon code (of degree 1 in this case) in our transformation from $(3,\ell)$-MCRH to CRH cannot improve the shrinkage ℓ that we can start with.

We can similarly show that our choices in our transformation from 4-MCRH to CRH were also close to optimal. To start with, note that we cannot use our approach to go directly from 4-MCRH to CRH. This would require showing that 2 sets C_i of size 2 each have *no* intersection, which implies that for any codeword $c \in C$, there exists at most one λ for which there is some c' such that $c[\lambda] = c'[\lambda]$. A simple counting argument shows that this cannot happen unless $|\Sigma| \geq 2^n$, at which point all shrinkage is lost.

So to get a CRH from a 4-MCRH, we have to construct a 3-MCRH first. The following proposition implies that the loss in shrinkage in going from a 4-MCRH to a 3-MCRH is at least $n/3$ irrespective of the choice of the code C. So, in order to go from a $(4,\ell)$-MCRH to a CRH, ℓ would have to be at least $(n/3 + n/2) = 5n/6$, which is what we obtained.

Proposition 21. *In any code $C \subseteq \Sigma^N$ where $|C| = 2^n$ and any 3 codewords all agree on at most 2 coordinates, it has to be that $|\Sigma| \geq \Omega(2^{n/3})$.*

Proof. Again, truncate the codewords in C to the first 3 coordinates. This set of truncated codewords has to have at least $2^n/2$ distinct elements. Otherwise, this would mean that some 3 codewords in C agreed on the first 3 coordinates, which is precluded by the hypothesis. Thus, $\Sigma^3 \geq 2^n/2$, which implies that $\Sigma \geq (2^n/2)^{1/3}$. □

Obstructions to improvement. More generally, the above techniques can be used to prove the following general bound.

Proposition 22. *In any code $C \subseteq \Sigma^N$ where $|C| = 2^n$ and any p codewords all agree on at most q coordinates, it has to be that $|\Sigma| \geq (2^n/(p-1))^{1/(q+1)}$.*

Proposition 22 implies, for instance, that going from a 5-MCRH to a 4-MCRH (resp. 3-MCRH) using our approach would incur a loss of at least $n/4$ (resp. $n/3$) in shrinkage. Further, we can show that going from a 5-MCRH to a 3-MCRH in fact incurs a loss of at least $n/2$. In order to do this, we show that if the alphabet Σ is of size somewhat less than $2^{n/2}$, then there actually does exist a $T \subseteq C$ of size 4 such that the sets C_1, C_2, C_3 with their requisite properties are subsets of T. This is implied immediately by the following proposition.

Proposition 23. *For any code $C \subseteq \Sigma^N$ such that $|C| = 2^n$ and $|\Sigma| \leq 2^{n/2}/2$, there exist codewords $c, c_1, c_2, c_3 \in C$ such that on each of the first three coordinates, at least two of the c_i's agree with c.*

Proof. Consider just the first three coordinates of codewords in C. Let S_1 be the set of all codewords c such that there exists another codeword c' such that $c[1] = c'[1]$ and $c[2] = c'[2]$. Let S_2 and S_3 denote similar sets of codewords that instead look at the first and third, and second and third coordinates, respectively. If we can prove that there exists a codeword c that is contained in all of the S_i's then we would be done.

We do this by showing that each S_i has to be large. Take S_1, for instance. By definition, S_1 is the set of all codewords that have some "collision" in the first two coordinates. Since the first two coordinates are supported on Σ^2, the number of codewords that do not have any collisions in these coordinates can be at most $|\Sigma|^2$. Thus, S_1 (and similarly S_2 and S_3) is of size at least $(2^n - |\Sigma|^2) \geq (3/4) \cdot 2^n$. So there has to exist at least one codeword in the intersection of all three S_i's. Take this codeword to be c, and its colliding codeword in each S_i to be the respective c_i. This proves the proposition. □

To go from a 5-MCRH to a CRH, we would first have to go to a 4-MCRH or a 3-MCRH, and then to a CRH from there. As noted above, going from a 4-MCRH (resp. 3-MCRH) to a CRH already incurs a loss of at least $5n/6$ (resp. $n/2$) in shrinkage. Following the above bounds on constructions of 4- or 3-MCRH from 5-MCRH, neither of these routes is viable, and our approach as is cannot be used to construct a CRH from a 5-MCRH (and thus also from t-MCRH for $t > 5$).

Potential workarounds. One possibility to getting a CRH from even a 5-MCRH is to use the hash function h itself to split up codewords that may otherwise appear together in the sets C_i. The codewords in any given c_i correspond to a set of inputs that collide under both h and g, but so far we have only used the fact that they collide under g. Could their collision under h be used meaningfully somehow to improve this approach? Of course, there might also be approaches significantly different from ours that construct CRH from such MCRH.

Acknowledgments

We thank Itay Berman and Akshay Degwekar for being part of much of our exploration of multicollision resistance, for helpful discussions, and for being great collaborators in general.

References

[BDRV18] Itay Berman, Akshay Degwekar, Ron D. Rothblum, and Prashant Nalini Vasudevan. Multi-collision resistant hash functions and their applications. In Nielsen and Rijmen [NR18], pages 133–161.

[BHKY19] Nir Bitansky, Iftach Haitner, Ilan Komargodski, and Eylon Yogev. Distributional collision resistance beyond one-way functions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 667–695. Springer, 2019.

[BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 671–684. ACM, 2018.

[BV17] Nir Bitansky and Vinod Vaikuntanathan. A note on perfect correctness by derandomization. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 592–606, 2017.

[DGRV11] Zeev Dvir, Dan Gutfreund, Guy N. Rothblum, and Salil P. Vadhan. On approximating the entropy of polynomial mappings. In Bernard Chazelle, editor, *Innovations in Computer Science - ICS 2011, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings*, pages 460–475. Tsinghua University Press, 2011.

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

[DN07] Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6):1513–1543, 2007.

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 342–360. Springer, 2004.

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2: Basic Applications*. Cambridge University Press, 2004.

[GRS00] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning polynomials with queries: The highly noisy case. *SIAM J. Discret. Math.*, 13(4):535–570, 2000.

[HHRS15] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - tight lower bounds on the round and communication complexities of statistically hiding commitments. *SIAM J. Comput.*, 44(1):193–242, 2015.

[HR04] Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 92–105. Springer, 2004.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 230–235. IEEE Computer Society, 1989.

[Jou04] Antoine Joux. Multicollisions in iterated hash functions. application to cascaded constructions. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 306–316. Springer, 2004.

[KNY17] Ilan Komargodski, Moni Naor, and Eylon Yogev. White-box vs. black-box complexity of search problems: Ramsey and graph property testing. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 622–632. IEEE Computer Society, 2017.

[KNY18] Ilan Komargodski, Moni Naor, and Eylon Yogev. Collision resistant hashing for paranoids: Dealing with multiple collisions. In Nielsen and Rijmen [NR18], pages 162–194.

[KY18] Ilan Komargodski and Eylon Yogev. On distributional collision resistant hashing. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 303–327. Springer, 2018.

[Lau83] Clemens Lautemann. BPP and the polynomial hierarchy. *Inf. Process. Lett.*, 17(4):215–217, 1983.

[Nao89] Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 128–136. Springer, 1989.

[NR18] Jesper Buus Nielsen and Vincent Rijmen, editors. *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*. Springer, 2018.

[NS07] Mridul Nandi and Douglas R. Stinson. Multicollision attacks on some generalized sequential hash functions. *IEEE Trans. Inf. Theory*, 53(2):759–767, 2007.

[Sho88] Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 283–290, 1988.

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, 1998.

[Sud97] Madhu Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complex.*, 13(1):180–193, 1997.

[YW07] Hongbo Yu and Xiaoyun Wang. Multi-collision attack on the compression functions of MD4 and 3-pass HAVAL. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *Information Security and Cryptology - ICISC 2007, 10th International Conference, Seoul, Korea, November 29-30, 2007, Proceedings*, volume 4817 of *Lecture Notes in Computer Science*, pages 206–226. Springer, 2007.