Collision Resistance from Multi-Collision Resistance for all Constant Parameters

Jan Buzek and Stefano Tessaro

Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle, WA, USA
jan.buzek123@gmail.com, tessaro@cs.washington.edu

Abstract. A *t-multi-collision-resistant hash function* (t-MCRH) is a family of shrinking functions for which it is computationally hard to find t distinct inputs mapped to the same output by a function sampled from this family. Several works have shown that t-MCRHs are sufficient for many of the applications of collision-resistant hash functions (CRHs), which correspond to the special case of $t = 2$.

An important question is hence whether t-MCRHs for $t > 2$ are fundamentally weaker objects than CRHs. As a first step towards resolving this question, Rothblum and Vasudevan (CRYPTO '22) recently gave non-black-box constructions of infinitely-often secure CRHs from t-MCRHs for $t \in \{3, 4\}$ assuming the MCRH is sufficiently shrinking. Earlier on, Komargodski and Yogev (CRYPTO '18) also showed that t-MCRHs for any constant t imply the weaker notion of a *distributional* CRH.

In this paper, we remove the limitations of prior works, and completely resolve the question of the power of t-MCRHs for constant t in the infinitely-often regime, showing that the existence of such a function family always implies the existence of an infinitely-often secure CRH. As in the works mentioned above, our proof is non-black-box and non-constructive. We further give a new domain extension result for MCRHs that enables us to show that the underlying MCRH need only have arbitrarily small linear shrinkage (mapping $(1+\epsilon)n$ bits to n bits for any fixed $\epsilon > 0$) to imply the existence of CRHs.

1 Introduction

Hash functions are central primitives in cryptography. They are meant to satisfy a number of different security requirements, but the by far most widespread requirement is that of *collision resistance*. Informally, a collision-resistant hash function family (or CRH, for short) is a family of shrinking functions in which it is hard to find two inputs mapping to the same output (i.e. a collision) for a randomly selected function from the family.

While CRHs are widely regarded as belonging to symmetric cryptography (i.e., they are considered to be part of Minicrypt in the sense of Impagaliazzo's five worlds), we do not know how build them from *one-way functions* (OWFs). In fact, a black-box separation between the notions was proved by Simon [16]. This is a sharp contrast to most other symmetric cryptographic primitives, including pseudorandom generators [8], pseudorandom functions [4], and secret-key encryption —all of these enjoy black-box constructions from one-way functions.

Multi-collision resistance. The above context naturally motivates the study of notions that are potentially between OWFs and CRHs in order to better understand their relation. A number of papers [10,1,2,11] proposed to study *multi-collision resistance* (MCRH) as such a notion. (The notion had previously been considered in a cryptanalytics context, see e.g. [9].) Informally, a t-MCRH is a family of shrinking functions for which it is hard to find a size t collision, i.e., t distinct inputs mapping to the same output. When the amount of shrinkage plays an important role, we specify the shrinkage parameter k, where a (t, k)-MCRH is a family of functions mapping kn bits to n bits which is also a t-MCRH.

One expects that the notion of a (t, k)-MCRH could become weaker as t increases (because any adversary is required to find larger collisions), and stronger as k increases (because the functions must be more shrinking). In general, t and k can be unbounded functions of the security parameter n, as long as they are at most polynomial. Note however that for the special case of $t = 2$ (i.e., a CRH) shrinkage is less relevant. Indeed, standard domain-extension techniques [12,3] transform a CRH shrinking only a single bit into a CRH with arbitrary (polynomial) shrinkage. In contrast, such a strong result is not known for $t > 2$. A nontrivial result that improves the shrinkage of a t-MCRH at the cost of increasing t is described in [2].

Does multi-collision resistance imply collision resistance? A relevant question in view of the above is therefore whether MCRHs are actually weaker objects than CRHs. As a first step in this direction, Komargodski and Yogev [11] showed that any t-MCRH with constant t implies a *distributional* CRH with infinitely-often security. Infinitely-often security means an adversary fails for infinitely many values of the security parameter n, instead of failing for all values that are sufficiently large. Distributional CRHs are a weaker variant of CRHs, which guarantee that no efficient adversary can sample uniformly random collisions. More precisely, no adversary should be able to sample pairs (x_1, x_2) where x_1 is uniformly random, and x_2 is uniformly random conditioned on colliding with x_1. This is a more difficult task for the adversary than merely finding a collision, making a dCRH a weaker primitive than a CRH. Currently, dCRHs are not known to imply CRHs, leaving open the question of whether fully fledged CRHs can also be built from MCRHs.

Recently, Rothblum and Vasudevan [14] made a first step in this direction by providing constructions of infinitely-often secure CRHs (for short, io-CRHs) from io-(t, k)-MCRHs (and hence from (t, k)-MCRHs). Their techniques inherently require either $t = 3$ and $k \approx 2$, or $t = 4$ and $k \approx 6$ [1].

This leaves therefore the following question open:

[1] They generalize their approach to construct t'-MCRHs from t-MCRHs with $t > 4$ for certain values $t' < t$ (see [14, Theorem 5]). However, there is no sequence of parameters which would allow them to use this to construct an io-CRH from a t-MCRH with $t > 4$.

Can we build io-CRHs from io-(t,k)-MCRHs for any[2] *values of t and k?*

We answer this question affirmatively for any constant t and any constant $k > 1$. Our construction, like those given by [11,14], is non-black-box and non-uniform (alternatively, our construction is uniform if we consider security against uniform adversaries.) An additional benefit of our construction is that it is simpler than prior works, while following their general blueprint.

1.1 Our Results

We now overview our results in greater detail. Below, in Section 1.2, we give a technical overview. Our main result is the following theorem.

Theorem 1. *For any constants t and $\epsilon > 0$, if an io-$(t, 1+\epsilon)$-MCRH exists, then an io-CRH exists.*

We note that this theorem implicitly contains two statements: First, the equivalence of io-t-MCRHs for various constant values of t; second, the equivalence of io-(t,k)-MCRHs for fixed t and various constant values of $k > 1$. Indeed, our proof of the theorem consists of combining two transformations, which we discuss separately.

Improving multi-collision resistance. The first transformation is our main technical result, and is the part of our proof that is non-constructive and non-black-box. In particular we prove the following theorem.

Theorem 2. *For any constant c, equal to a power of two, there exists a constant c_2 such that if an io-$\left(2^{c/2}, c + c_2\frac{\log(n)}{n}\right)$-MCRH exists, then an io-CRH exists.*

This theorem is a substantial improvement over prior results, allowing us to construct an io-CRH starting from a sufficiently shrinking io-t-MCRH for any constant t by choosing the appropriate c. The technique from [14] only allows us to start from $t \in \{3, 4\}$ and that of [11] only constructs distributional io-CRHs. Our proof technique is inspired by those from [11,14], but is in some sense simpler, in that we do not rely on any sophisticated combinatorial tools.

Still, as in [11,14], our construction internally uses an adversary, and therefore if we target security against non-uniform adversaries (which we do), our construction is also non-uniform. (If we instead use uniform adversaries, the construction would also be uniform.) It is an interesting open question whether a black-box proof of Theorem 1 is possible. Previously a black-box separation between various MCRH notions of was claimed [10], but, as mentioned in [14], there is a gap in the proof [13], and such a separation is not currently known.

Removing the "infinitely often" restrictions is also an interesting problem, although, in a pragmatic sense, io-CRHs appear nearly as strong CRHs. For most constructions, it is uncommon for adversaries to only fail on infinitely many security parameters as opposed to all sufficiently large ones.

[2] For extremely small values of k or large values of t this may be impossible; see the note following Definition 1.

Domain extension. We complement the above result with a new a domain extension result for MCRHs, which can be combined with the above theorem to construct a CRH starting with any linearly shrinking t-MCRH with constant t. Although some domain extension results for MCRHs are already known [2], they can only be used to construct t-MCRHs with superconstant t, even when starting from a 3-MCRH. Such transformations cannot be used with Theorem 2 to prove Theorem 1.

This second transformation is a black-box construction of a family of hash functions with larger domain from a given MCRH. We prove the following theorem.

Theorem 3 (Domain Extension for MCRHs). *Given a $(t, 2)$-MCRH for constant t, for any constant λ there exists a (t', λ)-MCRH for $t' = 2t^{\log(\lambda)+2}$. The same holds when both the starting and ending MCRH are merely infinitely often secure.*

The key point in the above theorem is the $O(\log(\lambda))$ dependency in the exponent of t', which allows us to combine this with Theorem 2 and get functions with both improved collision resistance and improved shrinkage compared to our starting MCRH. Our proof of this result uses hash trees and a form of list recoverable code; see the next section for details.

1.2 Technical Overview

In this section, we give a more detailed overview of the proofs of the two main transformations described above.

Prior approach. It is helpful to first start with an overview of the transformation proposed by Rothblum and Vasudevan [14], as it will serve as a starting point for our approach. For simplicity let us start with a $(3, k)$-MCRH, which we now want to transform into a CRH. The idea is to define a new family which contains functions of form

$$f_{g,h}(x) = (h(x), g(x)) ,$$

where $h : \{0,1\}^{kn} \to \{0,1\}^n$ is sampled from the 3-MCRH, and $g : \{0,1\}^{kn} \to \{0,1\}^\ell$ is such that $n+\ell < k \cdot n$, and is sampled from a different function family $\mathcal{G}$ as explained below.

Now, if this new family is a CRH, then we are done. However, if it is not, then there exists an adversary $\mathcal{A}$ which, given g and h finds a collision for $f_{g,h}$. Here, let us just assume that the adversary is perfect, i.e., $\mathcal{A}(g, h) = (x_1, x_2)$ such that $x_1 \neq x_2$, $g(x_1) = g(x_2)$, and $h(x_1) = h(x_2)$. Now, we use $\mathcal{A}$ to define a new function family sampling functions $f_{\mathcal{A},h} : \mathcal{G} \to \{0,1\}^n$ defined by

$$f_{\mathcal{A},h}(g) = h(x_1) ,$$

where $(x_1, x_2) = \mathcal{A}(g, h)$. We choose $\mathcal{G}$ sufficiently large such that this function is also shrinking, and try to argue that it must be a CRH. Now, if we are given a collision $g_1 \neq g_2$ for $f_{\mathcal{A},h}$, it means that

$$h(x_{11}) = h(x_{12}) = h(x_{21}) = h(x_{22})$$

for $\mathcal{A}(g_1, h) = (x_{11}, x_{12})$ and $\mathcal{A}(g_2, h) = (x_{21}, x_{22})$. Further, we know that $x_{11} \neq x_{12}$ and $x_{21} \neq x_{22}$, as well as $g_1(x_{11}) = g_1(x_{12})$ and $g_2(x_{21}) = g_2(x_{22})$.

This does not mean yet that we have a 3-collision for h. In the worst case, we could have $\{x_{11}, x_{12}\} = \{x_{21}, x_{22}\}$. This is exactly what g is meant to prevent. For example, in this case one can relatively easily build a function family where $g : \{0,1\}^{kn} \to \{0,1\}^{kn/2}$ such that for any $x, y \in \{0,1\}^{kn}$ and any $g_1 \neq g_2$, we cannot have $g_1(x) = g_1(y)$ *and* $g_2(x) = g_2(y)$. This implies that $|\{x_{11}, x_{12}, x_{21}, x_{22}\}| \geq 3$.

There are now two immediate challenges. The first one is that $\mathcal{A}$ is only required to work on infinitely many values of the security parameter n. To overcome this, the idea is to instead prove that the resulting hash function is just an io-CRH, thus ensuring that $\mathcal{A}$ works on all sufficiently large values of the security parameter. The second assumption of adversaries succeeding with probability 1 is circumvented by a combination of a technical lemma (which we re-state as Lemma 1) from [14] and further arguments. Of course, an even bigger challenge is to understand how far this approach can be pushed beyond 3-collisions.

How far can we push this? More abstractly, the role of the function g is to 'split up' collisions so that small collisions in the resulting hash function $f_{g,h}$ can be used to reconstruct large ones in h. Intuitively, one should think in the ideal case of $g(x)$ as being truly random, and independent of the actual collisions found, but this is of course not achievable. Therefore, we are required to use combinatorial properties of g, and to this end, [14] use a variant of Reed Solomon codes. Unfortunately, this ends up making $g(x)$ too long, and no longer yields shrinking functions when $t > 4$. Interestingly, [11] circumvents this issue by producing a *distributional* CRH (dCRH, for short). To break the security of a dCRH, an adversary must produce random collisions, which allows [11] to use the adversary's randomness in place of $g(x)$ in a clever way, allowing the reduction to succeed for higher initial values of t. However, the resulting dCRH primitive is not as strong as a CRH.

Our approach. While we adopt the same general blueprint as [14], we take a different angle which leads to a much *simpler* and more *powerful* building block. Concretely, we now start with an io-(t^2, k)-MCRH family, which we are going to use to build an io-$(t, k/2)$-MCRH family.[3] We first define a new function family which samples functions h_s described by h from our original family, along with a string $s \in \{0,1\}^{kn/2}$, such that

$$f_{h,s}(x) = h(s \parallel x) \ .$$

Critically, h_s has half the shrinkage of h. The argument now is rather simple (in fact, simpler than in [14]). If this function is already a $(t, k/2)$-MCRH, then we

[3] For technical reasons, our resulting MCRH has slightly smaller shrinkage; we ignore this in the high level discussion.

are finished. If not, assume that we have an adversary $\mathcal{A}$ which finds a t-multi-collision for $f_{h,s}$, and assume it is perfect and deterministic, i.e., $\mathcal{A}(h, s)$ returns a multi-collision $x_1, \ldots, x_t$. Then, we build a new function

$$f_{\mathcal{A},h}(s) = h(s \parallel x_1) \ ,$$

where x_1 is a canonical element chosen from the multi-collision output by $\mathcal{A}$ on input (h, s). The function $f_{\mathcal{A},s}$ *also* has half the shrinkage of h.

Then, if an adversary finds a t-multi-collision for $f_{\mathcal{A},h}$, it means we have obtained *distinct* $s_1, \ldots, s_t$ for which the $h(s_i \parallel x_{i,j})$ collide, where $(x_{i,1}, \ldots, x_{i,t}) = \mathcal{A}(h, s)$ is the t-multi-collision $\mathcal{A}$ finds for $f_{h,s}$. This means that we have found a t^2-multi-collision for h, since the $s_i \parallel x_{i,j}$'s are all distinct.

Once again, this requires further technical arguments due to the fact that $\mathcal{A}$ only succeeds on a non-negligible fraction of the inputs h, s. To achieve this, we generalize the arguments from [14] to show that we can transform the MCRH defined from $\mathcal{A}$ to one that has a non-negligible fraction of h which work for all inputs s. From there, the transformation to a full MCRH follows from the lemma from [14]. This second transformation causes a small, subconstant loss in shrinkage, which is handled in the formal proof.

Improving the shrinkage. The above construction transforms an io-(t^2, k)-MCRH into an io-$(t, k/2)$-MCRH. This means that we can iterate this a constant number of times, getting a CRH from any io-$(2^{2^\ell}, 2^\ell(1+\epsilon))$-MCRH. However, what if we have a function with less shrinkage to start with? We resolve this by revisiting the question of domain extension for MCRHs.

More specifically, the main stage of our domain extension transformation is a construction of a much more shrinking (io)-MCRH starting from one mapping $2n$ bits to n bits, where we prove a strong result about the worsening of the collision resistance properties. Let h be a function from an io-$(t, 2)$-MCRH family. We build a function h' mapping λn bits to n bits from h. The new function h' works by first encoding its input into ln bits for $l > \lambda$, and then using h in a hash tree construction to hash down to an n bit output.

Using a hash tree alone, the resulting function could have collisions of size growing exponentially in l, because each application of h may have $t - 1$ size collisions. However, any large collisions in the hash tree take on only a few distinct values on any particular block of the input. This is because all the distinct values in a length $2n$ block of the input must be mapped to the same value by $O(\log(l))$ applications of h, so the size of any colliding set cannot be larger than $(t-1)^{O(\log(l))}$. This intuition is formalized in Proposition 3.

Now, for h to be suitably multi-collision resistant it suffices for the encoding used in h' to avoid having a large intersection with any sets taking only few values in each $2n$-bit block (we refer to this property as *rectangle-freeness*). This combinatorial property, which is tightly connected with list-recoverability in codes, is satisfied by a purely random function from λn bits to ln bits for $l = 4\lambda$ with high probability. Since our construction cannot use a purely random function due to its large description, we prove our construction in the setting of limited independence. Specifically, we will show that the same holds for a function chosen randomly from a set of merely K-wise independent hash functions, for a suitable constant K. We will also discuss how folded Reed Solomon codes can be used in the place of a K-wise independent function family.

Our construction is similar to [2] in the general idea of using a code followed by a hash tree. However, the analysis in [2] results in too much of an increase in t for our purposes. In particular, even starting from a t-MCRH with constant t, the resulting construction is only provably a t'-MCRH with t' growing with n, and thus cannot be combined with our main transformation. Additionally, we believe our construction and analysis is more natural, showing that the requirements on the code, which in [2] were presented as a highly specialized definition for the specific construction, are achieved by a K-wise independent function family. Finally, the construction in [2] is interwoven in the construction of commitment schemes, the main focus of that work, whereas our construction extracts the core of the domain extension argument.

Proving Theorem 1. Finally, there are two minor technical difficulties to overcome in completing the proof of Theorem 1. First, while Theorem 1 assumes only the existence of a MCRH with arbitrarily small linear shrinkage, Theorem 3 begins with a MCRH that shrinks by a factor of two. We solve this by first applying a simple version of the Merkle-Damgård construction [12,3] to improve the shrinkage factor to 2, while keeping the collision resistance parameter t constant. We can then improve the shrinkage to an arbitrary constant while controlling the deterioration in t via Theorem 3.

The second difficulty is that Theorem 2 requires a MCRH with collision resistance parameter $t = 2^{c/2}$ for c a power of two. To circumvent this, note that for any t, there is a $T > t$ which is a power of 2, and that any t-MCRH is by definition also a T-MCRH. The trade-off between collision resistance and shrinkage in Theorem 3 is sufficiently good to allow us to use the t-MCRH it produces as a T-MCRH for T as above, allowing us to close the seeming gap between the consequences of Theorem 3 and the assumptions of Theorem 2.

1.3 Organization

The rest of this paper is organized as follows. The next section gives various definitions necessary for the formal discussion. Section 3 gives the main construction improving collision resistance, and the core of the proof of Theorem 2. Section 4 gives the proof of Theorem 3 and our domain extension results. Section 5 combines the transformations improving collision resistance and shrinkage together, completing the proof of Theorem 1. Section 6 is dedicated to transforming a weakly partial domain MCRH (see Definition 3) to a full MCRH. This is a technical result needed to complete the proof of Theorem 2; it is given separately because it may be of independent interest in similar proofs.

2 Definitions

In this section we present the relevant definitions of hash functions, as well as more technical notions and facts that will serve as intermediate steps in our proofs.

General notation. Throughout this paper, we let $\|$ denote string concatenation. We use $[n]$ to denote $\{1, 2, ..., n\}$. All logarithms are in base 2. For a set $C \subset X$ and a function $f : X \to Y$, we use $f(C)$ to denote the corresponding image $\{y \in Y : \exists x \in C \colon f(x) = y\}$. For a randomized algorithm A with input y we write $x \leftarrow A(y)$ to denote a random sample from A on input y. We use $\mathrm{negl}(n)$ to generically refer to a positive function $f : \mathbb{N} \to \mathbb{R}$ which is negligible, that is, eventually smaller than any inverse polynomial function $1/p(n)$.

Function families. We describe a function family via a probabilistic family of polynomial size circuits Gen, referred to as the *function family generator*. On input the security parameter 1^n, Gen outputs the description of a function h, typically as a circuit. (We will use h to refer both to the function and the circuit implementing it.) We will let Gen be both non-uniform or uniform, depending on the context. (We elaborate on this a bit further below.)

Multi-collision resistance. For a an integer t, a function h, and a subset C of the domain of h, we define the predicate $\mathsf{MCOLL}_{h,t}(C)$ which is true if and only if $|C| = t$ and for all $x, x' \in C$, we have $h(x) = h(x')$, i.e., C describes a *t-multi-collision* under the function h.

Definition 1 (MCRH [10,14]). *For functions $t(n)$ and $k(n)$, a (t, k)-multi-collision resistant hash function (for short, (t, k)-MCRH) is described by a generator* Gen *which, on input 1^n, outputs the description of a function $h : \{0, 1\}^{k(n)\cdot n} \to \{0, 1\}^n$ such that for any family of polynomially sized circuits $(A_n)_{n\in\mathbb{N}}$, every polynomial $p(n)$, and for all $n \in \mathbb{N}$ sufficiently large,*

$$\Pr_{h\leftarrow\mathsf{Gen}(1^n)}\left[\mathsf{MCOLL}_{h,t}(C) \;\middle|\; C \leftarrow A_n(h)\right] \leq \frac{1}{p(n)} \;.$$

An io-(t, k)-MCRH only requires the above to hold for an infinite sequence of security parameters $n \in \mathbb{N}$.

For an MCRH to be nontrivial, it is necessary for it to be sufficiently shrinking so that the existence of size t-collisions is guaranteed. In particular, if the MCRH's output is l bits shorter than its input, t must be smaller than 2^l for the notion to be nontrivial. We are primarily interested in the case where the MCRH shrinks its input to a fraction of its original length (corresponding to constant $k > 1$), in which case exponential size collisions are guaranteed to exist.

When k is not specified and a t-MCRH is mentioned, it is assumed to be a sufficiently large constant. A CRH is any shrinking 2-MCRH. There are well-known transformations of a (io-)CRH shrinking one bit to one with arbitrary polynomial shrinkage, so we do not need to specify the shrinkage when discussing (io-)CRHs.

Uniformity. Our results hold in both the non-uniform and uniform computational settings. In the uniform setting, MCRHs are given by PPT (Probabilistic Polynomial Time) *generator* algorithms Gen, and we model the adversaries as uniform (PPT) algorithms. At a high level, because we use adversaries codes' to define MCRHs as part of our proofs, we do not consider the setting of uniform MCRHs secure against non-uniform adversaries. In the non-uniform setting, MCRHs are given by probabilistic polynomial sized circuit families as formalized above, and the adversaries are also modeled by such circuit families. Throughout the paper we formalize our results in the non-uniform setting.

Weaker forms of MCRHs. The following definitions are weaker forms of the MCRH definition, which we will use to construct fully-fledged MCRHs. The first one is from [14], whereas the latter is new.

Definition 2 (Partial MCRH ([14] Definition 6)). *For functions $t(n)$ and $k(n)$, a partial (t,k)-MCRH is described by a generator* Gen *which, on input 1^n, outputs the description of a function $h : \{0,1\}^{k(n)\cdot n} \to (\{0,1\}^n \cup \{\bot\})$ such that:*

1. *For any family of polynomially sized circuits $(A_n)_{n\in\mathbb{N}}$, every polynomial $p(n)$, for all sufficiently large $n \in \mathbb{N}$,*

$$\Pr_{h\leftarrow \mathsf{Gen}(1^n)}[\mathsf{MCOLL}_{h,t}(C) \wedge \bot \notin h(C) \mid C \leftarrow A_n(h)] \le \frac{1}{p(n)}$$

2. *There is a polynomial q such that with all but negligible probability over the outputs of* $\mathsf{Gen}(1^n)$*,*

$$|\{x \in \{0,1\}^n : h(x) \neq \bot\}| \ge \frac{2^n}{q(n)}.$$

In a partial io-(t,k)-MCRH the first condition above needs to hold only for an infinite sequence of security parameters $n \in \mathbb{N}$; the second is still required to hold for all security parameters.

We will use the following lemma rephrased from [14] that shows how to transform a partial domain MCRH to a full MCRH.

Lemma 1 (Partial to Full MCRH ([14] Lemma 7 Restated)). *If there exists a partial (t,k)-MCRH then there exists a $(t, k - O(\frac{\log(n)}{n}))$-MCRH. The same holds in the infinitely often case and/or if the construction is uniform.*

We will also need the following weakening of Definition 2.

Definition 3 (Weakly Partial MCRH). *For functions $t(n)$ and $k(n)$, a weakly partial (t,k)-MCRH is a described by a generator* Gen *which, on input 1^n, outputs the description of a function $h : \{0,1\}^{k(n)\cdot n} \to (\{0,1\}^n \cup \{\bot\})$ such that:*

1. *For any family of polynomially sized circuits $(A_n)_{n\in\mathbb{N}}$ and every polynomial $p(n)$, for all sufficiently large $n \in \mathbb{N}$,*

$$\Pr_{h\leftarrow \mathsf{Gen}(1^n)}[\mathsf{MCOLL}_{h,t}(C) \wedge \bot \notin h(C) \mid C \leftarrow A_n(h)] \le \frac{1}{p(n)}\ .$$

2. *There are polynomials $q_1(n), q_2(n)$ such that*

$$\Pr_{h \leftarrow \mathsf{Gen}(1^n)}\left[|\{x \in \{0,1\}^n : h(x) \neq \bot\}| \geq \frac{2^n}{q_1(n)}\right] \geq \frac{1}{q_2(n)} .$$

A weakly partial io-(t,k)-MCRH is the same, except that the first condition above needs to hold only for an infinite sequence of security parameters $n \in \mathbb{N}$; the second is still required to hold for all security parameters..

In Section 6 we show how to transform any weakly partial (t,k)-MCRH into a partial (t,k)-MCRH, which can be further transformed into a MCRH using Lemma 1.

K-WISE INDEPENDENT HASH FUNCTIONS. We will also make use of the standard notion of K-wise independent hash functions, which exist unconditionally for K polynomial in n (e.g., by considering random polynomials of degree $K-1$).

Definition 4 (K-wise Independent Hash Functions). *For a function $K(n)$, a family of functions described by* Gen *outputting, on input 1^n, a function $h : \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$ is* K-wise independent *if for any n, for any sequence of $K = K(n)$ distinct inputs $x_1, \ldots, x_K \in \{0,1\}^{\ell(n)}$ and any sequence of K (not necessarily distinct) outputs $y_1, \ldots, y_K \in \{0,1\}^{m(n)}$, we have*

$$\Pr_{h \leftarrow \mathsf{Gen}(1^n)}[\forall i \in [K] \colon h(x_i) = y_i] = \frac{1}{2^{K(n) \cdot m(n)}} .$$

3 Improving Collision Resistance

This section gives a proof of the following theorem which establishes the existence of an io-CRH from that of a suitable io-MCRH, and is our first main result. Later on, we combine this with domain extension techniques to complete the picture and show that io-CRHs are implied by any io-(t,k)-MCRH where $t > 2$ and $k > 1$ are constants.

Theorem 2. *For any constant c, equal to a power of two, there exists a constant c_2 such that if an io-$\left(2^{c/2}, c + c_2 \frac{\log(n)}{n}\right)$-MCRH exists, then an io-CRH exists.*

As in the case of the main construction of [14], the proof of this theorem yields a non-black-box and non-uniform construction. We also note that, while we do not do so explicitly, the resulting construction can be made uniform if we make all security assumptions hold against uniform adversaries. The core of the proof of Theorem 2 is the following lemma, which gives a way of trading off the shrinkage and collision-resistance parameters of (io)-MCRHs.

Lemma 2. *Let $t = t(n)$ and $k = k(n)$ be functions bounded above by polynomials. If there exists an io-(t^2, k)-MCRH, then there exists an io-$(t, \frac{k}{2} - O(\frac{\log(n)}{n}))$-MCRH.*

We first assume this lemma, and use it to prove Theorem 2.

Proof (of Theorem 2). We simply apply the lemma above $\log(c) - 1$ times. After i steps, we obtain an io-$(2^{c/2^{1+i}}, c/2^i + (c_2 - c_3(i)) \log(n)/n)$-MCRH, where the constant $c_3(i)$ is obtained by summing the constants from the big-O in Lemma 2. If we set $i = \log(c) - 1$, then this gives us an io-$(2, 2 + O(\log(n)/n))$-MCRH, assuming c_2 was chosen to be sufficiently large. Therefore, we get an io-CRH, as we wanted to show. □

Next we give a proof of Lemma 2. At the end of this section, we also state some results that follow from applying our techniques to t-MCRHs for super-constant $t = \omega(1)$.

3.1 Proof of Lemma 2

Let Gen be the sampler for the io-(t^2, k)-MCRH, which samples functions from kn bits to n bits. We assume without loss of generality that k is even. We set $m = kn/2$.

THE Gen' FAMILY. For $s \in \{0,1\}^m$, define $f_{h,s} \colon \{0,1\}^m \to \{0,1\}^n$ by

$$f_{h,s}(x) = h(x \parallel s) \ .$$

Let Gen' be the sampler that, on input 1^n samples $h \leftarrow \mathsf{Gen}(1^n)$ as well as $s \leftarrow \{0,1\}^m$ and returns $f_{h,s}$.

We now have two cases. Either Gen' is an io-$(t, k/2)$-MCRH, and then we are done. Or it is not. In this case, there exist a polynomial $q(n)$, an integer $n_0 = n_0(A)$, and a polynomial-sized family of adversaries $A = (A_n)_{n\in\mathbb{N}}$ such that

$$\Pr_{f_{h,s} \leftarrow \mathsf{Gen}'(1^n)} \left[\mathsf{MCOLL}_{h,t}(C) \;\middle|\; C \leftarrow A_n(f_{h,s}) \right] > \frac{1}{q(n)}$$

for all $n \geq n_0$. Without loss of generality, we patch A_n so that it outputs either an ordered t-tuple that forms a collision, or a special symbol $\perp$ when it fails. (Note that A_n can easily be modified to check if the collision it has found is valid, and output $\perp$ if not.)

THE Gen_A FAMILY. Let $A_n(f_{h,s})$ be the output of A_n on input the hash function $f_{h,s}$, and let $A_n(f_{h,s})[1]$ be the first element of the t-tuple that $A_n(f_{h,s})$ outputs, or $\perp$ if $A_n(f_{h,s})$ outputs $\perp$. We then define the function

$$g_{h,A_n} \colon \{0,1\}^m \to \{0,1\}^n$$

such that

$$g_{h,A_n}(s) = f_{h,s}(A_n(f_{h,s})[1]) \ .$$

Let Gen_A be a sampler that on input 1^n samples $h \leftarrow \mathsf{Gen}(1^n)$ and then returns g_{h,A_n}. We note that this sampler is not uniform, as the adversary A_n is assumed not to be uniform. Furthermore, the function is only well-behaved (in the sense of our analysis below) for $n \geq n_0(A)$. Consequently, for $n < n_0(A)$, we define g_{h,A_n} to be some arbitrary function which is defined on all of its domain.

Gen_A IS A WEAKLY PARTIAL MCRH. We now establish that Gen_A is a weakly partial io-$(t, k/2)$-MCRH. We prove that it meets the two parts of the definition in the propositions below.

Proposition 1. *For every polynomial-sized adversaries $B = (B_n)_{n\in\mathbb{N}}$, every polynomial $p(n)$, and infinitely many values of n, we have*

$$\Pr_{g_{h,A_n} \leftarrow \mathsf{Gen}_A(1^n)} \left[\mathsf{MCOLL}_{g_{h,A_n},t}(C) \wedge \bot \notin g_{h,A_n}(C) \mid C \leftarrow B_n(g_{h,A_n})\right] \leq \frac{1}{p(n)} \ .$$

Proof (Of Proposition 1). Assume that the the proposition is not true, i.e., that there actually exists a polynomial-size adversary $B = (B_n)_{n\in\mathbb{N}}$ and a polynomial $p(n)$ such that

$$\Pr_{g_{h,A_n} \leftarrow \mathsf{Gen}_A(1^n)} \left[\mathsf{MCOLL}_{g_{h,A_n},t}(C) \wedge \bot \notin g_{h,A_n}(C) \mid C \leftarrow B_n(g_{h,A_n})\right] > \frac{1}{p(n)}$$

for all n's larger than some fixed value $n_0 = n_0(B)$. We now build a polynomial-sized adversary $\tilde{B} = (\tilde{B}_n)_{n\in\mathbb{N}}$ against Gen. The adversary is described as follows, on input a circuit h in the range of $\mathsf{Gen}(1^n)$:

Adversary $\tilde{B}_n(h)$:

1. Run $B_n(g_{h,A_n})$ to get an output C, which we assume without loss of generality to consist of t distinct inputs $s_1, \dots, s_t$, which may or may not be a multi-collision.
2. For each $i \in [t]$, run $A_n(f_{h,s_i})$. If the output is not $\bot$, let $(x_{i,1}, ..., x_{i,t})$ for $i \in [t]$ be the output of A_n.
3. Output the set $C' = \{s_i \parallel x_{i,j} \mid i, j \in [t]\}$.

It is not hard to see that $\tilde{B}_n$ can be implemented in polynomial size as well, as long as t is polynomial.

Now, assume $n \geq \max\{n_0(A), n_0(B)\}$. We observe the following:

- Whenever the output C of $B_n(g_{h,A_n})$ is such that $\bot \notin g_{h,A_n}(C)$, then, by definition, we also have that $A_n(f_{h,s_i}) = (x_{i,1}, \dots, x_{i,t}) \neq \bot$ for all $i \in [t]$. Further, for each $i \in [t]$, we have $h(s_i \parallel x_{i,j}) = g_{h,A_n}(s_i)$ for all $j \in [t]$.
- If additionally C is a multi-collision, i.e., $g_{h,A_n}(s_1) = \cdots = g_{h,A_n}(s_t)$, this means that $h(s_i \parallel x_{i,j})$ is a muticollision for h.
- We have additionally that $s_i \parallel x_{i,j} \neq s_{i'} \parallel x_{i',j'}$ whenever $(i,j) \neq (i',j')$. This is because either $i \neq i'$, in which case $s_i \neq s_{i'}$, or $i = i'$, in which case $x_{i,j} \neq x_{i,j'}$. Therefore, C' is a multi-collision among t^2 inputs for h.

Using the fact that given $h \leftarrow \mathsf{Gen}(1^n)$, directly computing g_{h,A_n} to feed it into B_n within $\tilde{B}_n$ gives us the same distribution as sampling the latter from Gen_A, we also have

$$\Pr_{h \leftarrow \mathsf{Gen}(1^n)} \left[\mathsf{MCOLL}_{h,t}(C') \wedge \bot \notin h(C') \mid C' \leftarrow \tilde{B}_n(h)\right] > \frac{1}{p(n)}$$

for all n's larger than $n \geq \max\{n_0(A), n_0(B)\}$. This contradicts the io-MCRH assumption on Gen, and hence concludes the proof. □

Proposition 2. *With $q_1(n) = q_2(n) = 2q(n)$, we have*

$$\Pr_{g_{h,A_n} \leftarrow \mathsf{Gen}_A(1^n)} \left[|\{s \in \{0,1\}^m : g_{h,A_n}(s) \neq \bot\}| \geq \frac{2^m}{q_1(n)} \right] \geq \frac{1}{q_2(n)}$$

for all n.

Proof (Of Proposition 2). If $n < n_0(A)$, the claim is vacuously true, as we defined the function on the whole domain. So, let us assume that $n \geq n_0(A)$. We call an h good if it has the property that

$$\Pr_{s \leftarrow \{0,1\}^m} \left[\mathsf{MCOLL}_{h,t}(C) \;\middle|\; C \leftarrow A_n(f_{h,s}) \right] \geq \frac{1}{2q(n)}$$

Note that the fact that h is good is equivalent to saying that g_{h,A_n} is defined on at least $2^m/2q(n)$ of the inputs s. The probability that an h is good is therefore at least $\frac{1}{2q(n)}$ by a Markov-like argument, as if this were not true, the probability that A_n succeeds against $f_{h,s}$ would be smaller than $1/q(n)$, a contradiction to our assumption on A_n's success probability being at least $1/q(n)$. □

Wrapping this up. It remains to turn Gen_A into a full MCRH. This will require some technical work, which is packed in the proof of Lemma 3, which we defer to Section 6 below. However, the proof of Lemma 2 follows directly from this claim applied to Gen_A.

Lemma 3 (Weakly partial to full MCRH). *If a weakly partial (t, k)-MCRH exists, then so does a $(t, k - O(\frac{\log(n)}{n}))$-MCRH. The same holds in the infinitely often case and/or if the construction is uniform.*

3.2 t-MCRHs for superconstant t

The proof of Theorem 2 cannot be directly extended to show that any io-t-MCRH, for a growing function $t(n)$ (such as a polynomial), implies an io-CRH. This is because to prove such a claim, we would want to apply the transformation $O(\log(t(n)))$ times, which is an unbounded number of times. One could try to give a direct proof, but it is not clear how to do so, as we require building a new adversary at every step.

However, Lemma 2 does show that the existence of t-MCRH for any polynomial $t(n)$ is equivalent to the existence of a n^ϵ-MCRH for any $\epsilon > 0$ fixed (with sufficiently large shrinkage). This is because starting from the former, if $t(n)$ is asymptotically bounded by n^c, then applying our transformation $\log(\frac{c}{\epsilon})$ times as in the proof above results in a n^ϵ-MCRH. Since $\log(\frac{c}{\epsilon})$ is a constant, this iterative construction does not run in to the problems described above. This gives us the following corollary of Lemma 2.

Corollary 1. *If, for some polynomial $t(n)$, there exists a sufficiently shrinking io-t-MCRH, then there exists an io-n^ϵ-MCRH for any constant $\epsilon > 0$.*

4 Domain Extension

In this section, we complement our main transformation with one that improves the shrinkage of a MCRH at the cost of worsening the collision-resistance property. The trade-off between the improved shrinkage and worsened collision resistance will however be favorable, in that combining this transformation with Theorem 2 will allow us to significantly weaken the starting shrinkage of the t-MCRH required to prove existence of a CRH. (We will indeed do so in Section 5 below.) In summary, we are going to prove the following theorem.

Theorem 3 (Domain Extension for MCRHs). *Given a $(t,2)$-MCRH for constant t, for any constant λ there exists a (t',λ)-MCRH for $t' = 2t^{\log(\lambda)+2}$. The same holds when both the starting and ending MCRH are merely infinitely often secure.*

Overview of our proof. Our proof of Theorem 3 follows the high-level approach originally used in [2] to create commitment schemes from a MCRH.

However, as mentioned in the introduction, there are significant differences. The primary technical difference is that the domain extension theorems in [2] produce t'-MCRHs with t' growing in n, even when starting with a t-MCRH with constant t. This prevents us from combining such transformations with the techniques from Section 3.

More concretely, the proof is obtained by combining the results from Sections 4.1 and 4.2 below. We follow the natural approach of using a hash tree to build a function with larger domain from a given MCRH. To this end, let $h\colon \{0,1\}^{2n} \to \{0,1\}^n$ be sampled from our $(t,2)$-MCRH. Our hash-tree construction breaks the input into blocks of length $2n$, hashes these with h, concatenates the outputs and repeats until the result has only n bits. By a standard argument, if it were hard to find collisions in h, it would be hard to find collisions at each level, and the overall construction would be a CRH. However, because h is sampled from a t-MCRH, it may be easy to find $(t-1)$ size collisions in h, and the hash tree constructed may itself have exponential size collisions (relative to the multiplicative domain extension factor). Still, we will observe that large collisions have a particular structure. Specifically, for each block at a leaf of the hash tree, the colliding set only takes on a small number of different values.

The key idea is to take advantage of this structure by restricting the hash tree to a set of inputs that avoid these large local collisions. Concretely, we define the new MCRH to compute an encoding from length λn bit strings to length ln bit strings for λ smaller than but on the same order as l (e.g. $\lambda = l/4$), and then apply the hash tree. If the encoding is sufficiently good at avoiding local collisions, the resulting function should be a t'-MCRH with a much larger domain and t' not too much larger than t.

4.1 Local MCRHs from Hash Trees

We start with the standard definition of a hash tree.

Definition 5 (Hash Tree). *Let $h\colon \{0,1\}^{2n} \to \{0,1\}^n$ be a function. For any depth parameter $d \geq 0$, we define*

$$\mathrm{ht}_{d,h}\colon \{0,1\}^{2^d n} \to \{0,1\}^n$$

such that $\mathrm{ht}_{0,h}(x) = x$ for all $x \in \{0,1\}^n$, and for all $x = x_L \parallel x_R$, where $x_L, x_R \in \{0,1\}^{2^{d-1}n}$

$$\mathrm{ht}_{d,h}(x) = h(\mathrm{ht}_{d-1,h}(x_L) \parallel \mathrm{ht}_{d-1,h}(x_R)) \ .$$

Now, for a constant t, let Gen be a $(t,2)$-MCRH. Let d be a constant depth parameter. We define a new generator $\mathsf{Gen}_{\mathcal{HT}} = \mathsf{Gen}_{\mathcal{HT}}[\mathsf{Gen}, d]$ which, on input 1^n, runs $h \leftarrow \mathsf{Gen}(1^n)$, and then outputs a circuit implementing $\mathrm{ht}_{d,h}$ as in Definition 5. Here, and below, we use the following notation: Let $C \subseteq \{0,1\}^*$ be a set of strings. Let $i \in \{1,2,\ldots\}$ be a non-zero natural number. Then, assuming all strings in C have length at least $2ni$, we define

$$C_i = \{x\,[2n(i-1),\ldots,2ni-1] \ : \ x \in C\}$$

where $x[a,\ldots,b]$ denotes the substring of x including all characters from the a-th one to the b-th one; the indexing is inclusive of both endpoints, and starts from 0. In other words, C_i is the projection of the set C on the i-th $2n$-bit block of each string.

The next proposition says that $\mathsf{Gen}_{\mathcal{HT}}$ is *locally* a good MCRH, meaning it is hard to find a multi-collision C for which the projection C_i on the i-th $2n$-bit block contains many distinct values.

Proposition 3 (Local Collisions on Hash Trees). *Let d, Gen, and $\mathsf{Gen}_{\mathcal{HT}}$ be as above. Then, for any polynomial-sized ciruits $(A_n)_{n\in\mathbb{N}}$ and all polynomials $p(n)$*

$$\Pr_{\mathrm{ht}\leftarrow\mathsf{Gen}_{\mathcal{HT}}(1^n)}\left[\begin{array}{l}\mathsf{MCOLL}_{\mathrm{ht},|C|}(C) \text{ and} \\ \exists i \in [2^{d-1}] \text{ s.t. } |C_i| > (t-1)^d\end{array} \,\middle|\, C \leftarrow A_n(\mathrm{ht})\right] \leq \frac{1}{p(n)}$$

for all sufficiently large n.

We note that Proposition 3 can be be re-stated for the case where Gen is only infinitely-often secure, in which case the statement is only true for infinitely many n's. The proof remains identical.

Proof. Suppose $A = (A_n)_{n\in\mathbb{N}}$ is such that

$$\Pr_{\mathrm{ht}\leftarrow\mathsf{Gen}_{\mathcal{HT}}(1^n)}\left[\begin{array}{l}\mathsf{MCOLL}_{\mathrm{ht},|C|}(C) \text{ and} \\ \exists i \in [2^{d-1}] \text{ s.t. } |C_i| > (t-1)^d\end{array} \,\middle|\, C \leftarrow A_n(\mathrm{ht})\right] > \frac{1}{p(n)}$$

for infinitely many n's.

We are now going to build an adversary $B = (B_n)_{n\in\mathbb{N}}$ which breaks the t-MCRH security of Gen. In particular, B_n, on input h in the support of $\mathsf{Gen}(1^n)$, will run $A_n(\mathrm{ht}_{d,h})$, and obtain C. If $|\mathrm{ht}_{d,h}(C)| = 1$, i.e., we have t-multi-collision, and there exists $i \in [2^{d-1}]$ such that $|C_i| > (t-1)^d$, then the adversary B_n runs the following recursive procedure $\mathsf{Extract}(h, d, C)$, and returns its output:

Procedure $\mathsf{Extract}(h, d, C)$:

```
1. Let $i \in [2^{d-1}]$ be smallest such that $|C_i| > (t-1)^d$
2. Let $Y_i = \{h(x) \,:\, x \in C_i\}$
3. If $|Y_i| \leq (t-1)^{d-1}$ then
   (a) Find $y \in Y_i$ and $C_y \subseteq C_i$ such that $h(x) = y$ for all
       $x \in C_y$ and $|C_y| = t$
   (b) Output $C_y$.
4. If $|Y_i| > (t-1)^{d-1}$ then
   (a) Let
       $$C' = \{h(x_1) \,\|\, \cdots \,\|\, h(x_{2^{d-1}}) \,:\, x_1 \,\|\, \cdots \,\|\, x_{2^{d-1}} \in C\} \ .$$
   (b) Output $\mathsf{Extract}(h, d-1, C')$.
```

It is not too hard to see that the recursive procedure always terminates with a t-multi-collision for h whenever A_n wins. (And thus the success probability of B_n equals that of A_n.) Indeed, at every iteration there are two cases:

Case 1: $|Y_i| \leq (t-1)^{d-1}$. Then, by the pigeonhole principle, there exists $y \in Y_i$ such that $h(x) = y$ for at least t distinct values in C_i. Then, we output these t values as a multi-collision, and are done.

Case 2: $|Y_i| > (t-1)^{d-1}$. Here, we clearly have $|\mathrm{ht}_{d-1,h}(C')| = 1$, and further, we must have that $|C'_{\lceil i/2 \rceil}| \geq |Y_i| > (t-1)^{d-1}$. Therefore, we can proceed recursively.

The execution terminates because if we ever call $\mathsf{Extract}(h, 1, d)$, we necessarily have $i = 1$, and $|Y_1| = 1$. Therefore, $\mathsf{Extract}$ also returns a t-multi-collision in this case, and thus B_n succeeds whenever A_n does. We also note that $\mathsf{Extract}$ can be implemented in polynomial time (and by polynomial size circuits) as long as $(t-1)^d$ is polynomial. This concludes the proof. □

4.2 Rectangle-free Function Families

In this section, we formalize the requirements on functions, which, when combined with a hash tree, yield domain extension for MCRHs, and also construct such functions. The required combinatorial property is similar to that used by [2] in the context of building commitment schemes from MCRHs, and essentially can be thought of as a special case of a list-recoverable code [5,7] with no efficiency requirements for list recovery. In the following, for two sets of strings $S_1 \subseteq \{0,1\}^m$ and $S_2 \subseteq \{0,1\}^n$, it is convenient to define $S_1 \times S_2 \subseteq \{0,1\}^{m+n}$ as the set of all strings $x \,\|\, y$ where $x \in S_1$ and $y \in S_2$.

Definition 6 (Rectangle). *For an even $l \geq 1$, we call a set $S \subseteq \{0,1\}^{ln}$ a* size-r rectangle *if there are sets $S_1, \ldots, S_{l/2} \subset \{0,1\}^{2n}$ of size r such that $S = S_1 \times S_2 \times \cdots \times S_{l/2}$.*

Definition 7 (Rectangle-Free Family). *Let* Gen *describe a function family consisting of functions* $f\colon \{0,1\}^{\lambda n} \to \{0,1\}^{ln}$*. We say that* Gen *is* (λ, l, r, R)*-*rectangle free *if*

$$\Pr_{f \leftarrow \mathsf{Gen}(1^n)}[\exists r\text{−size rectangle } S : |\mathrm{Im}(f) \cap S| \geq R]$$

and

$$\Pr_{f \leftarrow \mathsf{Gen}(1^n)}[f \text{ is not injective}]$$

are both negligible, where $\mathrm{Im}(f)$ *denotes the image of* f*.*

Rectangle-free families give domain extension. First we prove that, when combined with hash trees, rectangle-free function families give a way of extending the domain of any MCRH. Concretely, for constant t, t' and k, we assume we are given:

1. A $(t, 2)$-MCRH $\mathsf{Gen}_{\mathcal{H}}$
2. A $(\lambda, l, (t-1)^d, t')$-rectangle-free function family $\mathsf{Gen}_{\mathcal{F}}$ for $l = 2^d$

We define a family $\mathsf{Gen}_{\mathcal{G}}$, which on input 1^n, operates as follows:

- It samples $f \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^n)$
- It samples $\mathrm{ht}_{d,h} \leftarrow \mathsf{Gen}_{\mathcal{HT}}[\mathsf{Gen}_{\mathcal{H}}, d](1^n)$
- It returns the function g such that

$$g(x) = \mathrm{ht}_{d,h}(f(x)) \ .$$

We now prove the following proposition.

Proposition 4. $\mathsf{Gen}_{\mathcal{G}}$ *as defined above is a* (t', λ)*-MCRH.*

Proposition 4 also directly extends to the case when $\mathsf{Gen}_{\mathcal{H}}$ is an io-$(t, 2)$-MCRH, in which case $\mathsf{Gen}_{\mathcal{G}}$ is an io-(t', λ)-MCRH.

Proof. Suppose not, and let $A = (A_n)_{n \in \mathbb{N}}$ be an adversary against $\mathsf{Gen}_{\mathcal{G}}$, i.e., there exists a polynomial p such that we have

$$\Pr_{g \leftarrow \mathsf{Gen}_{\mathcal{G}}(1^n)}\left[\mathsf{MCOLL}_{g,t'}(C) \;\middle|\; C \leftarrow A_n(g)\right] > \frac{1}{p(n)} \ .$$

for infinitely many values n. Let's fix one such value of n, and denote by f the function sampled from $\mathsf{Gen}_{\mathcal{F}}(1^n)$ as part of g. Because $\mathsf{Gen}_{\mathcal{F}}$ is $(\lambda, \ell, (t-1)^d, t')$-rectangle free, assume further that n is sufficiently large so that

$$\Pr_{f \leftarrow \mathsf{Gen}(1^n)}\left[\exists (t-1)^d\text{−size rectangle } S \text{ s.t. } |\mathrm{Im}(f) \cap S| \geq t'\right] \leq \frac{1}{4p(n)} \ .$$

and

$$\Pr_{f \leftarrow \mathsf{Gen}(1^n)}[f \text{ is not injective}] \leq \frac{1}{4p(n)} \ .$$

In particular, let GOOD_f be the event that f is injective and for all $(t-1)^d$-sized rectangles S we have $|\mathrm{Im}(f) \cap S| < t'$. Then, for every such n,

$$\Pr_{g \leftarrow \mathsf{Gen}_{\mathcal{G}}(1^n)}\left[\mathsf{MCOLL}_{g,t'}(C)\ \wedge \mathsf{GOOD}_f \;\middle|\; C \leftarrow A_n(g)\right] > \frac{1}{p(n)} - 2 \cdot \frac{1}{4p(n)} = \frac{1}{2p(n)}\ .$$

Now, assume that $\mathsf{MCOLL}_{g,t'}(C)\ \wedge \mathsf{GOOD}_f$ indeed occurs. Define a random variable $C' = f(C) \subseteq \{0,1\}^{2^d n}$, and define C'_i for $i \in [2^{d-1}]$ as the projection of C' on the i-th $2n$-bit block of each string in x. Further, define the rectangle $S = C'_1 \times \cdots \times C'_{2^{d-1}}$. Note that $C' \subseteq S$ and $|C'| = t'$. (The latter is true because f is injective.) Then

$$|\mathrm{Im}(f) \cap S| \geq |C' \cap S| = |C'| = t'\ .$$

Now, because Good_f holds, it therefore cannot be that all C'_i's are at most size $(t-1)^d$, since then S would be a size $(t-1)^d$ rectangle with large intersection with $\mathrm{Im}(f)$. Thus, we have overall established that

$$\Pr_{g \leftarrow \mathsf{Gen}_{\mathcal{G}}(1^n)}\left[\mathsf{MCOLL}_{g,t'}(C)\ \wedge \exists i \in [2^{d-1}] : |C'_i| > (t-1)^d \;\middle|\; C \leftarrow A_n(g)\right] > \frac{1}{2p(n)}$$

for infinitely many n's. In turn, this allows us to give a simple adversary $B = (B_n)_{n \in \mathbb{N}}$ against $\mathsf{Gen}_{\mathcal{HT}} = \mathsf{Gen}_{\mathcal{HT}}[\mathsf{Gen}_{\mathcal{H}}, d]$ which, given $\mathrm{ht}_{d,h}$, internally samples f, builds g out of f and $\mathrm{ht}_{d,h}$ to simulate an execution of A_n, and then returns $C' = f(C)$. As the success probabilities of A_n and B_n are the same, we clearly have

$$\Pr_{\mathrm{ht} \leftarrow \mathsf{Gen}_{\mathcal{HT}}(1^n)}\left[\begin{matrix}\mathsf{MCOLL}_{\mathrm{ht},t'}(C') \wedge \\ \exists i \in [2^{d-1}] : |C'_i| > (t-1)^d\end{matrix} \;\middle|\; C' \leftarrow B_n(\mathrm{ht})\right] > \frac{1}{2p(n)}$$

for infinitely many n's. The existence of B however contradicts Proposition 3. This concludes the proof. □

Constructions of rectangle-free families. Now we give two constructions of rectangle-free families that result in domain extension when used in conjunction with Proposition 4 above. Our first construction is simply obtained from any t'-wise independent hash function family. This construction has the benefit of being very simple, and the proof essentially follows from the fact that a random function is rectangle-free for the parameters which we need, along with the realization that t'-wise independence is enough to carry out the proof. This rectangle-free family immediately yields meaningful domain extension, when combined with Proposition 4. In particular, Theorem 3 follows as its corollary. Below, we discuss some alternative de-randomized coding-theoretic instantiations of the family; the details of these are delegated to Appendix A.

Proposition 5 (Independent Hash Functions are Rectangle Free). *For any constant t, let Gen be a family of t'-wise independent hash functions from λn bits to ln bits for $l = 2^d$, $\lambda = l/4$, and $t' = 2t^{\log(\lambda)+2}$. Then Gen is a $(\lambda, l, (t-1)^d, t')$-rectangle-free family.*

Proof. Let S be an arbitrary size $(t-1)^d$ rectangle, and $f \leftarrow \mathsf{Gen}(1^n)$. Then for $i \in \{0, 1, ..., 2^{\lambda n} - 1\}$ let $X_{S,i}$ be the indicator random variable for the event that $f(\mathrm{Enc}(i))$ is in S, where Enc is the encoding of i as a binary string. Let $X_S = \sum_{i=0}^{2^{\lambda n}-1} X_{S,i}$, so that $|\mathrm{Im}(f) \cap S| = X_S$.

Then, since $f(\mathrm{Enc}(i))$ is, individually, uniformly random in $\{0,1\}^{ln}$ (by the independence of the hash function family) and $|S| = (t-1)^{dl/2}$, for all i,

$$\mathsf{E}\left[X_{S,i}\right] = \frac{1}{2^{ln}}(t-1)^{dl/2},$$

where the expectation is over the random choice of f. Further, t'-wise independence of Gen means that the X_i are t'-wise independent. Thus for any subset T of $\{0, 1, ..., 2^{\lambda n} - 1\}$ of size t',

$$\Pr\left[\forall i \in T\colon X_{S,i} = 1\right] = \mathsf{E}\left[X_{S,i}\right]^{t'} = \frac{1}{2^{lnt'}}(t-1)^{dlt'/2}$$

We now take a union bound over all such sets T, of which there are $\binom{2^{\lambda n}}{t'} \leq 2^{\lambda n t'}$ to get

$$\Pr\left[X_S \geq t'\right] \leq \Pr\left[\exists T\colon \forall i \in T\colon X_{S,i} = 1\right] \leq \frac{2^{\lambda n t'}}{2^{lnt'}}(t-1)^{dlt'/2} \ .$$

We also note that the number of $(t-1)^d$-rectangles is $\binom{2^{2n}}{(t-1)^d}^{l/2}$, since the binomial is the number of ways to choose each S_i, and any combination of $l/2$ of them determine S. Since

$$\binom{2^{2n}}{(t-1)^d}^{l/2} \leq 2^{nl(t-1)^d} \ ,$$

we get

$$\Pr\left[\exists S\colon X_S \geq t'\right] \leq 2^{nl(t-1)^d} \cdot \frac{2^{\lambda n t'}}{2^{lnt'}}(t-1)^{dlt'/2} \ .$$

We now show that the exponential factors in this bound go to zero. With $t' = 2t^{\log(\lambda)+2} + 1$, since $l = 4\lambda$ and $d = \log(l)$,

$$t' = 2t^{\log(4\lambda)} = 2t^{\log(l)} = 2t^d > 2(t-1)^d.$$

Thus, using that $l = 4\lambda$, we have

$$\begin{aligned} nl(t-1)^d + \lambda n t' - lnt' &= n\big(l(t-1)^d - (l-\lambda)t'\big) \\ &\leq n\big(l(t-1)^d - (l-\lambda)2(t-1)^d\big) \leq -\epsilon n \ , \end{aligned}$$

for some $\epsilon > 0$. Then asymptotically,

$$\Pr\left[\exists S\colon |\mathrm{Im}(f) \cap S| \geq t'\right] = \Pr\left[\exists S\colon X_S \geq t\right] \leq c2^{-n\epsilon}$$

for some constant c, showing that the first probability in the definition of a rectangle free code is indeed negligible, as required. Finally, since our hash function family is also pairwise independent, for all $x_1, x_2 \in \{0,1\}^\lambda$ we also have

$$\Pr\left[f(x_1) = f(x_2)\right] \leq 2^{-ln} ,$$

and another union bound over x_1 and x_2 gives

$$\Pr\left[\exists x_1, x_2 \colon\ f(x_1) = f(x_2)\right] \leq 2^{(-l+2\lambda)n} = 2^{-2\lambda n} ,$$

which is obviously negligible in n. But this last probability is just the probability that f is not injective, completing the proof. □

Derandomizing rectangle-freeness. We complement the above result by showing in Appendix A that one can build an explicit derandomized construction of a rectangle-free family based on codes, i.e., one where $\mathsf{Gen}_{\mathcal{F}}$ outputs a *fixed* function. Note that this is not necessary to prove Theorem 3, as the sampler for the MCRH family already uses randomness to output a particular hash function, and thus using a t'-wise independent family does not lead to any qualitative degradation of the result.

Nonetheless, we believe it is natural to ask whether randomness is necessary for a rectangle-free family, given its inherent coding-theoretic flavor. Surprisingly, achieving sufficiently good parameters seems to require fairly recent coding-theoretic machinery, which is a testament to the simplicity of the above result for t'-wise independent functions. Although the construction we give in Appendix A does not quite achieve the parameters of Theorem 3, it is strong enough to be used in the proof of Theorem 1. Note that the above result for t'-wise independence already implicitly shows, by the probabilistic method, that one can fix a single function meeting the requirements. However, constructing such a function explicitly is nontrivial.

5 Putting Pieces Together (Proof of Theorem 1)

This section is dedicated to the proof of Theorem 1, the main result, which we restate here.

Theorem 1. *For any constants t and $\epsilon > 0$, if an io-$(t, 1+\epsilon)$-MCRH exists, then an io-CRH exists.*

We will use the two results from Sections 3 and 4 in turn to prove the theorem; we restate them here for convenience.

Theorem 2. *For any constant c, equal to a power of two, there exists a constant c_2 such that if an io-$\left(2^{c/2}, c + c_2\frac{\log(n)}{n}\right)$-MCRH exists, then an io-CRH exists.*

Theorem 3 (Domain Extension for MCRHs). *Given a $(t, 2)$-MCRH for constant t, for any constant λ there exists a (t', λ)-MCRH for $t' = 2t^{\log(\lambda)+2}$. The same holds when both the starting and ending MCRH are merely infinitely often secure.*

Proof (of Theorem 1). We break up the proof of Theorem 1 into two steps. In the first part, we show that the Merkle-Damgård construction [12,3] allows us to build an io-$(t', 2)$-MCRH from any io-$(t, 1+\epsilon)$-MCRH such that if ϵ and t are constant, t' is as well. In the second step, we show that starting with an io-MCRH that maps $2n$ bits to n bits, we can arbitrarily improve the collision resistance parameter t by alternating our hash-tree based domain extension technique with our main transformation.

Step 1. Let Gen be a sampler for an io-$(t, 1+\epsilon)$-MCRH for constants t and $\epsilon > 0$. We define a new MCRH with sampler Gen' by the Merkle-Damgård construction [12,3] as follows. For h in the support of Gen, we define $h'_h : \{0,1\}^{2n} \to \{0,1\}^n$ by the following procedure:

```
Procedure $h'_h(x)$:

1. Let $c = \lceil \frac{1}{\epsilon} \rceil$.
2. Let $x' = x \parallel 0$ such that $|x'| = (c\epsilon + 1)n$.
3. Let $y \parallel y_1 \parallel ... \parallel y_c = x'$ such that $|y| = n$ and $|y_i| = n\epsilon$ for
   all $i \in [c]$.
4. Let $z_0 = y$
5. For $i \in [c]$, let $z_i = h(z_{i-1} \parallel y_i)$.
6. Output $z_c$.
```

Then Gen' simply samples from Gen and outputs the corresponding function h'_h:

```
Procedure $\mathsf{Gen}'(1^n)$:

1. $h \leftarrow \mathsf{Gen}(1^n)$
2. Output $h'_h$.
```

We now have the following lemma, which completes Step 1. Its proof is a simplified analog of that of Proposition 3.

Lemma 4. Gen' *as defined above is an io-$(t^c, 2)$-MCRH.*

Proof. Suppose not, and let $A = (A_n)_{n \in \mathbb{N}}$ be an efficient adversary against the t^c multi-collision resistance of Gen'. Then, let us define the adversary $B = (B_n)_{n \in \mathbb{N}}$ against t multi-collision resistance of Gen as follows.

```
Procedure $B_n(h)$:

1. Let $X = A_n(h'_h)$.
2. If $X = \bot$ output $\bot$. Else let $\{x_1, ..., x_{t^c}\} = X$.
3. Let $c$ be as defined in $h'_h$.
4. For each $i \in [c]$, let $Z_i$ be the set of values $z_i$ obtained in the
   computation of $h'_h(x_j)$ for $x_j \in X$.
5. For $i \in [c]$ with $i \geq 1$, for each $z \in Z_i$, let $W_z$ be the set of
   values $w \in Z_{i-1}$ such that for some $x_w \in X$, the computation
   of $h'_h(x_w)$ has $z_{i-1} = w$ and $z_i = z$. For each such $w$, let $u_w$
   be the value $y_i$ from the computation of $h'_h(x_w)$. If any $W_z$
   satisfies $|W_z| \geq t$, output $t$ elements of $\{(w, u_w) : w \in W_z\}$.
6. Output $\bot$.
```

Since A is a polynomial-size circuit family, it is not hard to see that B can be implemented in polynomial size because its circuits run A_n and do some additional efficient computations. We argue that whenever A_n successfully finds a t^c size collision on h'_h, B_n finds a t-size collision on h. Then, it follows that A cannot succeed with non-negligible probability.

Suppose A_n succeeds and returns the set X in Step 1. Then, for all $x_i, x_j \in X$, $h'_h(x_i) = h'_h(x_j)$ by definition of A_n finding a multi-collision. Also, $|Z_n| = 1$, since by definition $h'_h(x)$ is the z_n defined in the computation of h'_h on that x. By definition, $|Z_0| = |X| = t^c$. Therefore, since there are $c+1$ sets Z_i, there must exist an $i \in [c]$ such that $|Z_{i-1}| \geq t|Z_i|$ by the pigeonhole principle. For this choice of i, by averaging there must be a $z \in Z_i$ such that $|W_z| \geq t$. Thus if A_n succeeds then B_n will not reach Step 6.

By definition, the set that B_n outputs will have size t, since its elements are distinct because each corresponds to a distinct $w \in W_z$. Furthermore, they all collide under h, because by definition of h'_h they all satisfy $h(w, u_w) = z$. □

STEP 2. Now we have an io-$(t', 2)$-MCRH for $t' = t^c$. Note that since $c = \lceil \frac{1}{\epsilon} \rceil$ and ϵ is constant, c is also constant, and t' is constant. Theorem 3 gives that for any $\lambda = 2^{d-2} + 1$ for integer d, there exists an io-$(2(t')^d, \lambda)$-MCRH. Note that the exponent of the collision resistance parameter is growing as $O(d) = O(\log(\lambda))$, which is a sufficiently good trade-off between collision resistance and shrinkage to apply Theorem 2. Since Theorem 2 applies for MCRHs with collision resistance parameter $t = 2^{c/2}$ for c a power of 2, it remains to argue that we can choose t of this form.

To this end, choose λ a large enough constant such that $d \log(3t') < (\lambda-1)/2$ and such that $(\lambda-1)/2$ is an integer. Then since

$$2(t')^d \leq 2^{d \log(3t')} < 2^{(\lambda-1)/2} ,$$

our io-$(2(t')^d, \lambda)$-MCRH is also a io-$(2^{(\lambda-1)/2}, \lambda)$-MCRH, since this is a weaker collision resistance requirement. Since for any constant c_2, $\frac{\log(n)}{n}$ is $o(1)$, this implies there is an io-$(2^{(\lambda-1)/2}, \lambda - 1 + c_2\frac{\log(n)}{n})$-MCRH, since the latter MCRH simply has smaller shrinkage.

Applying Theorem 2 with $c = \lambda - 1$ proves the existence of an io-CRH. □

6 Weakly Partial to Full MCRH (Proof of Lemma 3)

This section is dedicated to the proof of Lemma 3, restated below for convenience.

Lemma 3 (Weakly partial to full MCRH). *If a weakly partial (t, k)-MCRH exists, then so does a $(t, k - O(\frac{\log(n)}{n}))$-MCRH. The same holds in the infinitely often case and/or if the construction is uniform.*

We prove Lemma 3 by breaking it down into two parts:

Lemma 6 (Weakly partial to partial MCRH). *If a weakly partial (t, k)-MCRH exists, then so does a partial (t, k)-MCRH. The same holds in the infinitely often case and/or if the construction is uniform.*

Lemma 1 (Partial to Full MCRH ([14] Lemma 7 Restated)). *If there exists a partial (t, k)-MCRH then there exists a $(t, k - O(\frac{\log(n)}{n}))$-MCRH. The same holds in the infinitely often case and/or if the construction is uniform.*

Assuming the above lemmas, Lemma 3 follows immediately by composition: if a weakly partial (t, k)-MCRH exists, so does a partial (t, k) by Lemma 6, and then a $(t, k - O(\frac{\log(n)}{n}))$-MCRH exists by Lemma 1.

Lemma 1 is proved in [14]. We give a proof of Lemma 6 below. The proof resembles that Lemma 8 in [14], which however only applied to the special case of a particular weakly-partial MCRH[4] used there. We generalize the proof by extracting the important properties into Definition 3 and proving Lemma 6.

Proof (of Lemma 6). Let Gen be a weakly partial (t, k)-MCRH, and $q_1(n)$ and $q_2(n)$ the polynomials associated with Gen as in the definition of a weakly partial MCRH. We will call an h sampled as $h \leftarrow \mathsf{Gen}(1^n)$ *good* whenever it is defined on a large fraction of its domain, and create a new sampler $\widetilde{\mathsf{Gen}}$ that samples such good h with all but negligible probability. Formally, for any h in the range of $\mathsf{Gen}(1^n)$, we define

$$\delta_h = \frac{|\{x \in \{0,1\}^n \ : \ h(x) \neq \perp\}|}{2^n}$$

and call h *good* if $\delta_h \geq \frac{1}{3q_1(n)}$.

We start by showing that we can efficiently generate a good h, with only negligible probability of failure. To this end, we define a MCRH sampler Gen_0 as follows:

```
Procedure $\mathsf{Gen}_0(1^n)$:
1. Sample $h \leftarrow \mathsf{Gen}(1^n)$.
2. For $m = nq_1(n)^2$, sample $x_1, \ldots, x_m \leftarrow \{0,1\}^{kn}$
3. Let $\hat{\delta}_h := |\{i \mid h(x_i) \neq \perp\}|/m$
4. If $\hat{\delta}_h \geq \frac{1}{2q_1(n)}$ output $h$, otherwise output $\perp$.
```

We note that Gen_0 outputs $\perp$ as a sign for aborting. This makes it strictly speaking not a valid generator (as it needs to output a function), but we can think of $\perp$ as being some canonical function. We observe that Gen_0 runs in polynomial time, because running Gen and m evaluations for h each take polynomial time. We have $h \neq \perp$ with non-negligible probability, as established by the following proposition.

Proposition 6 **(Gen_0 succeeds sufficiently often).**

$$\Pr_{h' \leftarrow \mathsf{Gen}_0(1^n)} [h' \neq \perp] \geq \frac{1}{2q_2(n)} \, .$$

[4] There it was not labelled as such, as the definition of a weakly partial MCRH is new in our work. However, the construction in [14] meets this definition.

Proof. Intuitively, Gen_0 is very likely to succeed, i.e., $h' \neq \perp$, as long as the hash function it samples in Step 1, which we denote by h, has high δ_h. Note if $h' \neq \perp$, then $h' = h$. Formally,

$$\Pr_{h' \leftarrow \mathsf{Gen}_0(1^n)}[h' \neq \perp] \geq \Pr_{h' \leftarrow \mathsf{Gen}_0(1^n)}\left[h' \neq \perp \;\middle|\; \delta_h \geq \frac{1}{q_1(n)}\right] \\ \times \Pr_{h \leftarrow \mathsf{Gen}(1^n)}\left[\delta_h \geq \frac{1}{q_1(n)}\right] .$$

By definition, since Gen is a weakly partial MCRH, the second probability is at least $\frac{1}{q_2(n)}$. To bound the first probability, by complements we have

$$\Pr_{h' \leftarrow \mathsf{Gen}_0(1^n)}\left[h' \neq \perp \;\middle|\; \delta_h \geq \frac{1}{q_1(n)}\right] \\ = 1 - \Pr_{h \leftarrow \mathsf{Gen}_0(1^n)}\left[\hat{\delta}_h < \frac{1}{2q_1(n)} \;\middle|\; \delta_h \geq \frac{1}{q_1(n)}\right] .$$

We clearly have $\hat{\delta}_h = \frac{1}{m}\sum_{i=1}^{m} 1_{\{h(x_i) \neq \perp\}}$, and for each $i \in [m]$, $1_{\{h(x_i) \neq \perp\}}$ is an unbiased estimator of δ_h. Then for any $h \leftarrow \mathsf{Gen}(1^n)$ with $\delta_h \geq \frac{1}{q_1(n)}$, over the randomness of step 2 of Gen_0, by Hoeffding's inequality

$$\Pr_{\text{Step 2 of } \mathsf{Gen}_0(1^n)}\left[\hat{\delta}_h < \frac{1}{2q_1(n)}\right] \leq \Pr_{\text{Step 2 of } \mathsf{Gen}_0(1^n)}\left[|\hat{\delta}_h - \delta_h| > \frac{1}{2q_1(n)}\right] \\ \leq 2e^{-2\frac{m}{(2q_1(n))^2}} = 2e^{-\frac{n}{2}} \leq \frac{1}{2} .$$

Thus $\Pr_{h' \leftarrow \mathsf{Gen}_0(1^n)}\left[h' \neq \perp \;\middle|\; \delta_h \geq \frac{1}{q_1(n)}\right] \geq \frac{1}{2}$, completing the proof of the proposition. □

We also show that conditioned on $h \neq \perp$, the h produced by Gen_0 are good with all but negligible probability.

Proposition 7 (Testing For Good h).

$$\Pr_{h \leftarrow \mathsf{Gen}_0(1^n)}\left[\delta_h \geq \frac{1}{3q_1(n)} \;\middle|\; h \neq \perp\right] \geq 1 - \mathrm{negl}(n) .$$

Proof. Taking complements, by definition of conditional probability and Proposition 6, we have

$$\Pr_{h \leftarrow \mathsf{Gen}_0(1^n)}\left[\delta_h < \frac{1}{3q_1(n)} \;\middle|\; h \neq \perp\right] = \frac{\Pr_{h \leftarrow \mathsf{Gen}(1^n)}\left[\delta_h < \frac{1}{3q_1(n)} \text{ and } h \neq \perp\right]}{\Pr_{h \leftarrow \mathsf{Gen}_0(1^n)}[h \neq \perp]} \\ \leq 2q_2(n) \Pr_{h \leftarrow \mathsf{Gen}(1^n)}\left[\delta_h < \frac{1}{3q_1(n)} \text{ and } h \neq \perp\right] .$$

As in the previous proposition, we apply Hoeffding's inequality to $\hat{\delta}_h$, which gives

$$\Pr_{h \leftarrow \mathsf{Gen}(1^n)}\left[\delta_h < \frac{1}{3q_1(n)} \text{ and } \hat{\delta}_h \geq \frac{1}{2q_1(n)}\right] \leq \Pr_{h \leftarrow \mathsf{Gen}(1^n)}\left[|\delta_h - \hat{\delta}_h| \geq \frac{1}{6q_1(n)}\right]$$
$$\leq 2e^{-2\frac{m}{(6q_1(n))^2}} = 2e^{-\frac{n}{18}} \ .$$

Thus overall

$$\Pr_{h \leftarrow \mathsf{Gen}_0(1^n)}\left[\delta_h < \frac{1}{3q_1(n)} \;\middle|\; h \neq \bot\right] \leq 2q_2(n) \cdot 2e^{-\frac{n}{18}} \leq \mathrm{negl}(n) \ .$$

The proposition follows by taking complements. □

Now, we define a new sampler $\widehat{\mathsf{Gen}}_0$ that repeatedly runs Gen_0 and ends the first time that it succeeds. This will boost the probability of success from the $\frac{1}{2q_2(n)}$ proven for Gen_0 above to $1 - \mathrm{negl}(n)$, while maintaining the property that the h's sampled are good.

```
Procedure $\widehat{\mathsf{Gen}}_0(1^n)$:
1. For $r = q_2(n) \times n$, sample $h_1, \ldots, h_r \leftarrow \mathsf{Gen}_0(1^n)$
2. If there exists $i \in [r]$ with $h_i \neq \bot$, output $h_i$
3. Output $h \leftarrow \mathsf{Gen}(1^n)$.
```

$\widehat{\mathsf{Gen}}_0$ is clearly polynomial time, and it always outputs a valid h. We first show that it satisfies part 1 of the definition of a partial MCRH.

Proposition 8 ($\widehat{\mathsf{Gen}}_0$ outputs good h).

$$\Pr_{h \leftarrow \widehat{\mathsf{Gen}}_0(1^n)}[h \text{ is good}] \geq 1 - \mathrm{negl}(n) \ .$$

Note that if the above proposition holds, then by definition of good, $\widehat{\mathsf{Gen}}_0$ satisfies the first part of the partial MCRH definition with $q(n) = 3q_1(n)$.

Proof. For $i \in [r]$, let BAD_i be the event that $h_i = \bot$ or $h_i \neq \bot$ but h_i it not good. Then,

$$\Pr_{h \leftarrow \widehat{\mathsf{Gen}}_0(1^n)}[h \text{ is not good}] \leq \Pr\left[\bigwedge_{i=1}^{r} \mathsf{BAD}_i\right] = \prod_{i=1}^{r} \Pr\left[\mathsf{BAD}_i\right] \ ,$$

since the h_i's are sampled independently. Further, note that for $h_i \leftarrow \mathsf{Gen}_0(1^n)$,

$$\Pr\left[\mathsf{BAD}_i\right] \leq \Pr\left[h_i = \bot\right] + \Pr\left[\delta_{h_i} < \frac{1}{3q_1(n)} \;\middle|\; h_i \neq \bot\right]$$
$$\leq 1 - \frac{1}{2q_2(n)} + \mathrm{negl}(n)$$

by Propositions 6 and 7. Plugging this into the above,

$$\begin{aligned}\Pr_{h\leftarrow\widehat{\mathsf{Gen}}_0(1^n)}[h \text{ is not good}] &\le \left(1-\frac{1}{2q_2(n)}+\mathrm{negl}(n)\right)^r \\ &\le \left(1-\frac{1}{2q_2(n)}\right)^m (1+\mathrm{negl(n)})^r ,\end{aligned}$$

since dividing a negligible function by $1-\frac{1}{2q_2(n)}$ keeps it negligible. Further, $(1+\mathrm{negl}(n))^r$ can be replaced by $1+\mathrm{negl}(n)$ by using the binomial theorem and the fact that r is polynomial. Also,

$$\left(1-\frac{1}{2q_2(n)}\right)^r \le e^{-n/2}$$

is negligible as well. This implies that $\Pr_{h\leftarrow\widehat{\mathsf{Gen}}_0(1^n)}[h \text{ is not good}]$ is negligible, too, concluding the proof. □

Finally, we show that the partial MCRH defined by $\widehat{\mathsf{Gen}}_0$ retains the security property from the underlying weakly partial MCRH. To this end, suppose that there exists a polynomial-size adversary $A = (A_n)_{n\in\mathbb{N}}$ and a polynomial $p(n)$ such that

$$\Pr_{h\leftarrow\widehat{\mathsf{Gen}}_0(1^n)}[\mathsf{MCOLL}_{h,t}(C) \ \wedge \ \bot \notin h(C) \mid C \leftarrow A_n(h)] \ge \frac{1}{p(n)}$$

for infinitely many n's. (Or for all sufficiently large n's in the infinitely often case.) Then, for the same n's, we have that

$$\Pr_{h\leftarrow\mathsf{Gen}(1^n)}[\mathsf{MCOLL}_{h,t}(C) \ \wedge \ \bot \notin h(C) \mid C \leftarrow A_n(h)] \ge \frac{1}{2p(n)q_2(n)} - \mathrm{negl}(n) .$$

To see this, assume without loss of generality that A_n is deterministic, and let GOOD the set of functions h in the support of $\widehat{\mathsf{Gen}}_0(1^n)$ for which A_n succeeds. Then, the above can be written as

$$\Pr_{h\leftarrow\widehat{\mathsf{Gen}}_0(1^n)}[h \in \mathsf{GOOD}] \ge \frac{1}{p(n)} .$$

Whenever $\widehat{\mathsf{Gen}}_0$ does not reach Step 3, the distribution of its outputs is identical to that of Gen_0 conditioned on not outputting $\bot$. We have shown indirectly above that the probability of ever reaching Step 3 is negligible. Therefore, with FAIL being the event that Step 3 is reached, then in particular

$$\begin{aligned}\Pr_{h\leftarrow\widehat{\mathsf{Gen}}_0(1^n)}[h \in \mathsf{GOOD}] &\le \Pr_{h\leftarrow\widehat{\mathsf{Gen}}_0(1^n)}[h \in \mathsf{GOOD} \mid \neg\mathsf{FAIL}] + \Pr[\mathsf{FAIL}] \\ &= \Pr_{h\leftarrow\mathsf{Gen}_0(1^n)}[h \in \mathsf{GOOD} \mid h \ne \bot] + \mathrm{negl}(n) .\end{aligned}$$

or, equivalently,

$$\Pr_{h \leftarrow \mathsf{Gen}_0(1^n)} [h \in \mathsf{GOOD} \mid h \neq \bot] \geq \frac{1}{p(n)} - \mathrm{negl}(n) \ .$$

Now, for any fixed function h^*, we have

$$\begin{aligned} \Pr_{h \leftarrow \mathsf{Gen}_0(1^n)} [h = h^* \mid h \neq \bot] &= \frac{\Pr_{h \leftarrow \mathsf{Gen}_0(1^n)} [h = h^* \ \wedge \ h \neq \bot]}{\Pr_{h \leftarrow \mathsf{Gen}_0(1^n)} [h \neq \bot]} \\ &\leq 2q_2(n) \cdot \Pr_{h \leftarrow \mathsf{Gen}_0(1^n)} [h = h^* \ \wedge \ h \neq \bot] \\ &\leq 2q_2(n) \cdot \Pr_{h \leftarrow \mathsf{Gen}(1^n)} [h = h^*] \ , \end{aligned}$$

where the first inequality follows from Proposition 6, whereas the second follows from the fact that a function h^* is output by Gen_0 with probability not larger than the probability that the same function is output by Gen itself. Therefore, the probability that A_n succeeds when h is output by $\mathsf{Gen}(1^n)$ is

$$\begin{aligned} \Pr_{h \leftarrow \mathsf{Gen}(1^n)} [h \in \mathsf{GOOD}] &\geq \frac{1}{2q_2(n)} \cdot \Pr_{h \leftarrow \mathsf{Gen}_0(1^n)} [h \in \mathsf{GOOD} \mid h \neq \bot] \\ &\geq \frac{1}{2q_2(n)p(n)} - \mathrm{negl}(n) \ , \end{aligned}$$

as we wanted to show. This completes the proof of Lemma 6. □

Acknowledgments

This research was partially supported by NSF grants CNS-2026774, CNS-2154174, a JP Morgan Faculty Award, a CISCO Faculty Award, and a gift from Microsoft.

References

1. Berman, I., Degwekar, A., Rothblum, R.D., Vasudevan, P.N.: Multi-collision resistant hash functions and their applications. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 133–161. Springer, Heidelberg (Apr / May 2018). https://doi.org/10.1007/978-3-319-78375-8_5
2. Bitansky, N., Kalai, Y.T., Paneth, O.: Multi-collision resistance: a paradigm for keyless hash functions. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) 50th ACM STOC. pp. 671–684. ACM Press (Jun 2018). https://doi.org/10.1145/3188745.3188870
3. Damgård, I.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (Aug 1990). https://doi.org/10.1007/0-387-34805-0_39
4. Goldreich, O., Goldwasser, S., Micali, S.: On the cryptographic applications of random functions. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO'84. LNCS, vol. 196, pp. 276–288. Springer, Heidelberg (Aug 1984)

5. Guruswami, V., Indyk, P.: Expander-based constructions of efficiently decodable codes. In: 42nd FOCS. pp. 658–667. IEEE Computer Society Press (Oct 2001). https://doi.org/10.1109/SFCS.2001.959942
6. Guruswami, V., Rudra, A.: Explicit capacity-achieving list-decodable codes. In: Kleinberg, J.M. (ed.) 38th ACM STOC. pp. 1–10. ACM Press (May 2006). https://doi.org/10.1145/1132516.1132518
7. Guruswami, V., Sudan, M.: Improved decoding of Reed-Solomon and algebraic-geometric codes. In: 39th FOCS. pp. 28–39. IEEE Computer Society Press (Nov 1998). https://doi.org/10.1109/SFCS.1998.743426
8. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28**(4), 1364–1396 (1999)
9. Joux, A.: Multicollisions in iterated hash functions. Application to cascaded constructions. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 306–316. Springer, Heidelberg (Aug 2004). https://doi.org/10.1007/978-3-540-28628-8_19
10. Komargodski, I., Naor, M., Yogev, E.: Collision resistant hashing for paranoids: Dealing with multiple collisions. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 162–194. Springer, Heidelberg (Apr / May 2018). https://doi.org/10.1007/978-3-319-78375-8_6
11. Komargodski, I., Yogev, E.: On distributional collision resistant hashing. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 303–327. Springer, Heidelberg (Aug 2018). https://doi.org/10.1007/978-3-319-96881-0_11
12. Merkle, R.C.: Fast software encryption functions. In: Menezes, A.J., Vanstone, S.A. (eds.) CRYPTO'90. LNCS, vol. 537, pp. 476–501. Springer, Heidelberg (Aug 1991). https://doi.org/10.1007/3-540-38424-3_34
13. Personal communication with the authors of [KNY18]
14. Rothblum, R.D., Vasudevan, P.N.: Collision-resistance from multi-collision-resistance. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part III. LNCS, vol. 13509, pp. 503–529. Springer, Heidelberg (Aug 2022). https://doi.org/10.1007/978-3-031-15982-4_17
15. Rudra, A.: List Decoding and Property Testing of Error Correcting Codes. Phd thesis, University of Washington, Seattle, WA (2007), available at https://cse.buffalo.edu/faculty/atri/papers/coding/thesis.html
16. Simon, D.R.: Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In: Nyberg, K. (ed.) EUROCRYPT'98. LNCS, vol. 1403, pp. 334–345. Springer, Heidelberg (May / Jun 1998). https://doi.org/10.1007/BFb0054137

A Derandomizing Rectangle-Freeness

In this section, we prove the existence of a single, explicit function which can be used as the encoding function in our domain extension construction for MCRHs. Recall that the goal was to construct a (t', λ)-MCRH using a $(t, 2)$-MCRH, for λ much larger than 2 and t' not too much larger than t. The idea behind the construction was to first encode the input into ln bits with an encoding function f, and then apply a hash tree built from the MCRH. As shown in Proposition 4, the notion of rectangle-freeness of Definition 7 captures the requirements on

f for this construction to be secure. Or course, since f is a single function, the requirements of the definition will hold with probability 1.

We view the function $f\colon \{0,1\}^{\lambda n} \to \{0,1\}^{ln}$ in Definition 7 as a code with block size $2n$ with rate $\frac{\lambda}{l}$. This allows us to relate rectangle-freeness to the following definition, which we restate from [15].

Definition 8 ([15] Definition 2.4, restated and specialized). *Let C be a q-ary code of block length l. Let $r, R \geq 1$ be integers and $0 \leq \rho \leq 1$ be real. Then C is (ρ, r, R)-List-Recoverable if the following is true. For every sequence of sets $S_1, ..., S_l$ where $S_i \subset [q]$ and $|S_i| \leq r$ for all $1 \leq i \leq l$, there are at most R codewords $\mathbf{c} = \langle c_1, ..., c_l \rangle \in C$ such that $c_i \in S_i$ for at least $(1-\rho)l$ positions i.*

For a family containing just a single function, our Definition 7 corresponds to the above with $\rho = 0$. Since for $\rho' > \rho$, a (ρ', r, R)-list-recoverable code is also (ρ, r, R)-list-recoverable, $\rho = 0$ is the weakest possible setting of the parameter. Now we argue that folded Reed Solomon codes give the desired f. We use the following corollary from [15]. (See also [6] for work leading up to this result.)

Corollary 2 ([15] Corollary 3.7, restated). *For every integer $r \geq 1$, for all K, K' with $0 \leq K \leq K' < 1$, for all constants $\epsilon \in (0, K]$, and for every prime p, there is an explicit family of folded Reed-Solomon codes, over fields of characteristic p that have rate at least K and which can be $(1 - R - \epsilon, r, R(l)$-list recovered in polynomial time, where for codes of block length l, $R(l) = (l/\epsilon^2)^{O(\epsilon^{-1}\log(r/K))}$ and the code is defined over alphabet of size $(l/\epsilon^2)^{O(\epsilon^{-2}\log(r/(1-K')))}$.*

We now show that the above can be used to obtain non-trivial domain extension by plugging it into the construction above Proposition 4. We are interested in codes where the rate K is a constant, and set $K = \frac{1}{2}$. Note that this implicitly sets $l = 2\lambda$, i.e. the code expands its inputs to twice their length. We set $\epsilon = \frac{1}{10}$, in particular constant. Then the above corollary gives a $(\rho, r, R(l))$-list-recoverable code for $\rho > 0$, which is also a $(0, r, R(l))$-list-recoverable code. We set $r = (t-1)^d$ for $d = \log(l)$. Then we have $R(l) = O(l^{cd\log(t-1)})$, where c is a fixed constant. By the big O notation of [15], this c is fixed as l varies.

By setting K' sufficiently close to 1, we can guarantee that the alphabet of the code is $q = 2^{2n}$; i.e. that the blocks of the code are $2n$ bits long as in our setting. Thus we have a fixed c such that for $t_{code} = O(l^{cd\log(t-1)})$ we can construct $(\lambda, l, (t-1)^d, t_{code})$-rectangle-free codes for arbitrary l and t.

We briefly argue that the parameters of this code are sufficient to be used in place of the randomized one. The parameters are used only in Step 2 of the proof of Theorem 1. There, we require that we can choose a sufficiently large (constant) λ for which $t_{code} \leq 2^{(\lambda-1)/2}$ so that we can apply Theorem 2 with $c = \lambda - 1$. This is possible with the derandomized code described because it has t_{code} growing (in λ) like $O(\lambda^{O(\log(\lambda)})$, which is of order $2^{\log^2(\lambda)}$ and eventually smaller than $2^{(\lambda-1)/2}$.

Therefore, such a code can be used in place of the randomized one from Proposition 5 in Proposition 4 to prove Theorem 1; the proof is analogous to the one carried out in section 5 with the modification to Step 2 as mentioned above.