A preliminary version of this paper appears in *Proceedings of the Cryptographers' Track of the RSA Conference (CT-RSA '12)*, Springer, 2012. This is the full version.

A New Pseudorandom Generator from Collision-Resistant Hash Functions

Alexandra Boldyreva Virendra Kumar
School of Computer Science, Georgia Institute of Technology
266 Ferst Drive, Atlanta, GA 30332-0765 USA
{sasha,virendra}@gatech.edu

Abstract

We present a new hash-function-based pseudorandom generator (PRG). Our PRG is reminiscent of the classical constructions iterating a function on a random seed and extracting Goldreich-Levin hardcore bits at each iteration step. The latest PRG of this type that relies on reasonable assumptions (regularity and one-wayness) is due to Haitner et al. In addition to a regular one-way function, each iteration in their "randomized iterate" scheme uses a new pairwise-independent function, whose descriptions are part of the seed of the PRG. Our construction does not use pairwise-independent functions and is thus more efficient, requiring less computation and a significantly shorter seed. Our scheme's security relies on the standard notions of collision-resistance and regularity of the underlying hash function, where the collision-resistance is required to be *exponential*. In particular, any polynomial-time adversary should have less than $2^{-n/2}$ probability of finding collisions, where n is the output size of the hash function. We later show how to relax the regularity assumption by introducing a new notion that we call *worst-case regularity*, which lower bounds the size of primages of different elements from the range (while the common regularity assumption requires all such sets to be of equal size). Unlike previous results, we provide a concrete security statement.

Keywords: Pseudorandom generator, hash function, collision-resistance, provable security.

1 Introduction

1.1 Motivation

A pseudorandom generator (PRG) is an important cryptographic primitive that was introduced by Blum and Micali [BM82], and later formalized into its current form by Yao [Yao82]. PRGs are used to generate *pseudo*random bits from a short random seed, which can then be used in place of truly random bits that most cryptographic schemes rely on. On the foundational side, PRGs can be used as a building block for more complex cryptographic objects like pseudorandom function (PRF) [GGM86], bit commitment [Nao91], etc.

In their seminal work, Håstad et al. [HILL99] building on the previous works [ILL89, Has90] show how to construct a PRG, henceforth called the HILL-PRG, from *any* one-way function. While the construction is of great theoretical value, it is extremely (orders of magnitude) inefficient compared to the Blum-Micali-Yao (BMY) PRG that builds on a one-way *permutation*. BMY-PRG is the most efficient known construction, whose security relies on a reasonable assumption. Practical standardized PRGs based on block-ciphers and hash functions (a hash function is a function whose range is smaller than the domain, also referred to as a compression function) [FIPS94], though much more efficient, rely on a rather strong and not well-studied assumption (in the theoretical cryptography community) that the underlying function is a PRF [DHY02], and thus are not a focus of this work. In this paper, we investigate a question of finding an *efficient* hash-function-based PRG, whose security relies on *collision-resistance*, a very well-studied and widely-used property of a hash function. A collision-resistant hash function (CRHF) is of course one-way but certainly not a permutation, as it compresses the input, and hence the BMY-PRG is not suitable for our problem.

1.2 Related Work

The seed length (as a function of the input length m of the underlying function) is an important measure of the efficiency and the security of a PRG. The best known bound for the HILL-PRG of $\mathcal{O}(m^8)$ was shown by Holenstein [Hol06]. This was later improved (for an alternative construction) to $\mathcal{O}(m^7)$ and $\mathcal{O}(m^4)$ by Haitner et al. in [HHR06a] and [HRV10], respectively. While the efficiency is obvious from the seed length, we present an example to truly appreciate the effect of seed length on the security of a PRG. Say, we have a one-way function that is secure, according to current standards, only for inputs of size at least 128 bits, then Holenstein's proof shows that the HILL-PRG is secure only for seeds of size (ignoring constants) at least 2^{56} bits! Several works have tried to bridge this huge gap from the BMY-PRG's seed length of $\mathcal{O}(m)$, by making stronger assumptions on the underlying function. Following are the two main types of strengthening in the assumption:

- **Regularity.** Goldreich et al. [GKL88] gave a construction of PRG with seed length $\mathcal{O}(m^3)$, whose security requires that the underlying function is one-way and *regular.* This was later improved by Haitner et al. [HHR06a], where they first present a tighter security proof for a construction similar to that of Goldreich et al., thus improving the seed length to $\mathcal{O}(m^2)$ (cf. Section 3.3 in [HHR06a]). In the following section of the same work, Haitner et al. show how the seed length can be further reduced to $\mathcal{O}(m \log m)$ by the use of bounded-space generators of Nisan [Nis92] (or, Impagliazzo et al. [INW94]).

- **Exponential hardness.** Holenstein [Hol06] gave a construction of PRG with seed length $\mathcal{O}(m^5)$, whose security relies on the underlying function being an *exponentially hard* one-way function. This was later improved by Haitner et al. to seed length $\mathcal{O}(m^2)$ in [HHR06b] and [HRV10], where the latter (unlike prior works) doesn't require adaptive calls to the one-way function.

1.3 Our Result

We construct a new hash-function-based PRG with seed length less than $2m$, i.e. as efficient as the BMY-PRG, thus improving the efficiency over all prior works which do not rely on permutations (i.e., function-based PRGs). Our scheme is reminiscent of the classical constructions [BM82, Yao82] iterating a function on a random seed and extracting Goldreich-Levin hardcore bits [GL89] at each iteration step. One notable difference from the BMY-PRG is that instead of a permutation, we use a hash function. Let h be a hash function mapping strings of size m bits to strings of size n bits, for $m > n$. Assume we have a random seed $x \| r$, where both x and r are n bits long, and we want to generate $l(> 2n)$ pseudorandom bits. The first bit of the output is the inner product of x and r, $\langle x, r\rangle$. To generate the second bit, compute $h_n^1(x) \leftarrow h(x \| 0^{m-n})$, and output $\langle h_n^1(x), r\rangle$. For the third bit, compute $h_n^2(x) \leftarrow h(h_n^1(x) \| 0^{m-n})$, and output $\langle h_n^2(x), r\rangle$. Repeat this process until $(l - n)$ bits are output, and also output r.

The latest PRG of this type that relies on reasonable assumptions (regularity and one-wayness) is due to Haitner et al. [HHR06a]. In addition to a regular one-way function, each iteration in their scheme uses a new pairwise-independent function (which is basically the only main difference from our construction), whose descriptions are part of the seed of the PRG. Our construction presented above does not use pairwise-independent functions and is thus more efficient, requiring less computation and a significantly shorter seed. Our scheme's security relies on the standard notions of collision-resistance and regularity of the underlying hash function, where the collision-resistance is required to be *exponential* (such a function is also referred in the literature as an "exponentially hard CRHF"). In particular, any polynomial-time adversary should have less than $2^{-n/2}$ probability of finding collisions, where n is the output size of the hash function. This should not be confused with the famous birthday bound, which roughly says that with $2^{n/2}$ number of random trials one can find collisions (with noticeable probability) in any hash function of output size n. Here, we are talking about the probability of collision and not the number of trials.

To the best of our knowledge, this is the first attempt to combine the above two strengthenings (i.e., regularity and exponential hardness) for improving the efficiency of a function-based PRG. While our assumption of exponential collision-resistance is quite strong, unlike the pseudorandomness of hash functions (which not only do not use secret keys, but are usually keyless) ours is still a very well accepted assumption in the community. Also, given the search for a new hash standard SHA-3 by the NIST [SHA3], it is plausible that some (if not all) of the candidate submissions to the competition provide exponential collision-resistance. We later show how to relax the regularity assumption by introducing a new notion that we call *worst-case regularity.* The notion of worst-case regularity lower bounds the size of the smallest set of preimages of different elements in the range, while the common regularity assumption requires all such sets to be of equal size. It was shown by Bellare and Kohno [BK04] that collision-resistance degrades exponentially (in the range of the function) when a function deviates from regularity, so a CRHF must be very "close" to regular, and experiments on practical hashes like SHA-1 support this claim (cf. Section 11 in [BK04]). So, the worst-case regularity assumption on a practical CRHF seems to be reasonable. We note that a notion similar to ours, called "weakly regular" was introduced in [GKL88]. This notion doesn't seem to be useful for our proof, because at a high level it captures the average of the sizes of different preimage sets of a function, while we need a lower bound on these sizes.

Levin [Lev87] observed that the BMY-type constructions are secure for functions that are one-way even when applied on their own outputs, a property called *one-way on iterates* (OWI), which one-way permutations trivially satisfy. However, it would be a stretch to assume that practical hashes have this property. We also note that collision-resistance alone may not be sufficient to prove that a function has the OWI property. Consider a CRHF h that acts as a permutation after one application, i.e. for any x in the domain of h, $h(h(x))$ is a permutation on $h(x)$ (some padding can be used to make $h(x)$ of input size, we omit this padding here for simplicity). For such a

CRHF, a security reduction from OWI to collision-resistance is not possible. The reason is that the output of an adversary that can break the OWI security $\big(y \in h^{-1}(h(h(x)))\big)$ cannot be used to find collisions in h, because the set $h^{-1}(h(h(x)))$ has just one element due to h being a permutation after one application. Someone familiar with the proofs of BMY and related PRG constructions may also be skeptical about the other direction, i.e. proving the security of our scheme assuming only the regularity and collision-resistance of h, without employing the "re-randomizing" pairwise-independent functions. The reason is that the security requires h to remain one-way on every iteration, but while h is believed to be collision-resistant and thus one-way (i.e., it is hard to invert $h(x)$ for a random point x in the domain), it is not necessarily hard to invert $h(h(x))$, because $h(x)$ (for a random x) is not necessarily a random point in the domain. In other words, the sets of points to which h is applied may shrink with each iteration, diminishing the one-wayness property of h, and thus violating the security of the PRG. Somewhat surprisingly, we show that these sets in our construction do not shrink significantly, if it is exponentially hard to find collisions in h. Unlike previous results on the security of PRGs, our theorem provides a concrete security statement, so that it is possible to see exactly how the security of our PRG degrades with the degradation in the collision-resistance of the underlying hash function, and thus allows a more accurate comparison with other schemes.

Our construction is very efficient (though still not comparable to practical standardized PRGs [FIPS94]) and simple, as at each iteration it uses a hash function and an inner-product computation, both of which are relatively fast. In Section 7, we show how using a classical method of [GKL88, Gol01] the efficiency of our scheme can be further improved by extracting up to a constant fraction of n hardcore bits at each iteration, as the underlying CRHF is assumed to be exponentially hard. We recall that our scheme is similar to the basic construction (that doesn't use bounded-space generators and has a seed length of $\mathcal{O}(m^2)$) of [HHR06a], but we do not use pairwise-independent functions, which permits significant efficiency improvements, allowing our scheme to have a very short seed. To put the comparison in perspective, the basic scheme of [HHR06a] (whose efficiency is comparable to ours) implemented with the compression function of SHA-256 (as the regular one-way function) would require around half a million random bits (as seed) to generate one extra pseudorandom bit, while our construction would just require 512 bits. Our security reduction is very tight, comparable to that of [HHR06a], even though the latter does not provide all the details for the concrete security of their PRG. While our construction is mainly of theoretical interest, we believe our approach and treatment has moved theoretically sound PRGs much further towards practical use. The novel worst-case regularity definition may be of independent interest.

2 Preliminaries

2.1 Notation

If f is a function, then $\mathsf{Im}(f)$ denotes the image set of f, and for any $y \in \mathsf{Im}(f)$, $\mathsf{Preim}(f, y)$ denotes the set of preimages of y under f. Let $a, b \in \mathbb{N}$, for simplicity and correctness, we define $\binom{a}{b}$ to be 1 if $a < b$. An adversary is an algorithm. By convention, the running-time of an adversary includes that of its overlying experiment. All algorithms are assumed to be randomized and efficient, and all functions are assumed to be efficiently computable, unless noted otherwise.

2.2 Hash Functions and their Security

Hash Function. Because of the known difficulties of defining collision-resistance (cf. Section 6.1 in [BR]), we follow the standard approach and define hash function families. A *hash function family* H is a collection of functions, where each $h \in H$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$, such that $m > n$. An instance $h \in H$ may be described by a key which is publicly known.

Collision-Resistance and Target Collision-Resistance. Let H be a hash function family, where each $h \in H$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$. The *collision-resistance* advantage of an adversary $\mathcal{C}$ attacking H, $\mathbf{Adv}_H^{\mathrm{cr}}(\mathcal{C})$ is defined as

$$\Pr\left[\, h \xleftarrow{\$} H,\ x, x' \xleftarrow{\$} \mathcal{C}(h):\ x \neq x' \in \{0,1\}^m \bigwedge h(x) = h(x') \,\right].$$

Also, the *target* collision-resistance advantage of an adversary $\mathcal{C}$ attacking H, $\mathbf{Adv}_H^{\mathrm{tcr}}(\mathcal{C})$ is defined as

$$\Pr\left[\, h \xleftarrow{\$} H, x \xleftarrow{\$} \{0,1\}^m, x' \xleftarrow{\$} \mathcal{C}(h,x):\ x' \in \{0,1\}^m \bigwedge x \neq x' \bigwedge h(x) = h(x') \,\right].$$

Birthday Attack. The birthday attack on a function $f : \{0,1\}^m \rightarrow \{0,1\}^n$ is defined in Figure 1. In this attack, $q \in \mathbb{N}$ points, $x_1, ..., x_q$ are picked independently at random from the domain. If any two of these points form a collision for f, then the attack is successful and those two points are returned. We denote the probability of success of the birthday attack on f by *collision probability*, $\mathbf{CP}(f,q)$. We will slightly abuse the notation sometimes, and use it for function families, where in $\mathbf{CP}(F,q)$ for a function family F, would mean the collision probability of a function picked at random from F.

```
For $i = 1, ..., q$
    $x_i \xleftarrow{\$} \{0,1\}^m$
    $y_i \leftarrow f(x_i)$
    If $(\exists j :\ j < i \ \bigwedge\ y_i = y_j \ \bigwedge\ x_i \neq x_j)$, return $(x_i, x_j)$.
```

Figure 1: Birthday attack (with q trials) on a function $f : \{0,1\}^m \rightarrow \{0,1\}^n$.

Regularity. A function $f : \{0,1\}^m \rightarrow \{0,1\}^n$ is said to be *regular*, if every point in the image set of f have equal number of preimages. Bellare and Kohno introduced the notion of a balance measure, denoted $\mu(f)$ (cf. Section 1 in [BK04]) to measure the regularity of a function: $\mu(f) = 1$ indicates that the function is fully regular and $\mu(f) = 0$ means fully irregular (an image point has the maximum number of preimages). The collision probability in the birthday attack for q trials, $\mathbf{CP}(f,q) = \binom{q}{2} \cdot 2^{-n\mu(f)}$ (up to constant factors), so the collision-resistance of any function degrades exponentially (in the range of the function) with the decline in its balance. A CRHF must therefore have a balance close to 1, and experiments on practical hashes like SHA-1 support this claim (cf. Equation 2, Section 11 in [BK04]). So, SHA-1 and other hash functions (SHA-256, SHA-512, etc.) can be assumed to be *close* to regular. We introduce a notion that we call *worst-case regularity* in Section 6 that also captures this closeness.

One-Wayness. Let F be a family of functions, where each $f \in F$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$. The *one-way* advantage of an adversary $\mathcal{I}$ attacking F, $\mathbf{Adv}_F^{\mathrm{ow}}(\mathcal{I})$ is defined as

$$\Pr\left[\, f \xleftarrow{\$} F, x \xleftarrow{\$} \{0,1\}^m, x' \xleftarrow{\$} \mathcal{I}(f, f(x)) :\ x' \in \{0,1\}^m \bigwedge f(x') = f(x) \,\right].$$

The one-way advantage of a function f (instead of a function family) can be defined similarly: the adversary is given $f(x)$ for a random x, and it has to return an element $x' \in \{0,1\}^m$ such that $f(x') = f(x)$.

Target Collision-Resistance and One-Wayness. The following relation between the notions is well-known.

Theorem 2.1. [[BR], Corollary 5.5] Let H be a hash function family, where each $h \in H$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$. Then for an adversary $\mathcal{I}$ with running time $t_{\mathcal{I}}$, there exists an adversary $\mathcal{C}$ with running time $t_{\mathcal{C}}$, so that

$$\mathbf{Adv}_H^{\mathrm{ow}}(\mathcal{I}) \leq 2 \cdot \mathbf{Adv}_H^{\mathrm{tcr}}(\mathcal{C}) + 2^{n-m}, \text{ and } t_{\mathcal{C}} \approx t_{\mathcal{I}}.$$

We now present a more general definition that also captures the one-wayness.

Hard to Compute. Let f and g be functions with the same domain $S_m \subseteq \{0,1\}^m$. The *hard-to-compute* advantage of an adversary $\mathcal{I}$ attacking (f, g), $\mathbf{Adv}_{f,g}^{\mathrm{htc}}(\mathcal{I})$ is defined as

$$\Pr\left[\, x \xleftarrow{\$} S_m :\ \mathcal{I}(f(x)) \in \mathsf{Preim}(g, f(x)) \,\right].$$

Note that for any adversary $\mathcal{I}$ and any function f, $\mathbf{Adv}_f^{\mathrm{ow}}(\mathcal{I}) = \mathbf{Adv}_{f,f}^{\mathrm{htc}}(\mathcal{I})$.

2.3 Hardcore Predicate

Hardcore Predicate. Informally, a hardcore predicate of a function is at least as hard to predict as inverting the function itself. Formally, let $g : \{0,1\}^m \to \{0,1\}^n$, $b : \{0,1\}^m \to \{0,1\}$ be two functions, and $a \xleftarrow{\$} \{0,1\}$ be a random bit. The *hardcore predicate* advantage of adversary $\mathcal{A}$, $\mathbf{Adv}_{g,b}^{\mathrm{hcp}}(\mathcal{A})$ is defined as

$$\Pr\left[\, x \xleftarrow{\$} \{0,1\}^m : \mathcal{A}(g(x), b(x)) = 1 \,\right] - \Pr\left[\, x \xleftarrow{\$} \{0,1\}^m : \mathcal{A}(g(x), a) = 1 \,\right].$$

Here $b(x)$ is called the *hardcore predicate (or bit)* of $g(x)$. In this paper, we use the general hardcore predicate construction of Goldreich and Levin [GL89], called the "GL-hardcore bit". For two bitstrings x $(= x_1\|\ldots\|x_m)$ and r $(= r_1\|\ldots\|r_m)$, define $b(x, r) = \langle x, r\rangle$, the inner product of x and r modulo 2, i.e. $\sum_{i=1}^m x_i \cdot r_i \pmod 2$. The following theorem is from [HHR06a], and states (using our notation) the security of the GL-hardcore bit.

Theorem 2.2. [Theorem 2.7, [HHR06a]] Let f and g be functions with the same domain $S_m \subseteq \{0,1\}^m$. For a random $x \in S_m$ and a random $r \in \{0,1\}^m$, define $\widehat{f}$ as $\widehat{f}(x, r) = (f(x), r)$, and its GL-hardcore bit b as $\langle z, r\rangle$, where $z \in \mathsf{Preim}(g, f(x))$ is one of the preimages of $f(x)$ under g. Then for an adversary $\mathcal{A}$ with running time $t_{\mathcal{A}}$, there exists an adversary $\mathcal{I}$ with running time $t_{\mathcal{I}}$, so that

$$\mathbf{Adv}_{\widehat{f},b}^{\mathrm{hcp}}(\mathcal{A}) \leq 4 \cdot \mathbf{Adv}_{f,g}^{\mathrm{htc}}(\mathcal{I}), \text{ and } t_{\mathcal{I}} = \mathcal{O}\left(m^3 \cdot t_{\mathcal{A}} \cdot \left(\mathbf{Adv}_{\widehat{f},b}^{\mathrm{hcp}}(\mathcal{A}) \right)^{-4} \right).$$

2.4 Pseudorandom Generator

Informally, a pseudorandom generator (PRG) is a function that expands a random seed into a longer pseudorandom bit sequence. PRGs were first proposed and constructed by Blum and Micali [BM82], and Yao [Yao82]. Let $G : \{0,1\}^m \to \{0,1\}^l$ be a function, so that $l > m$. The *prg* advantage of an adversary $\mathcal{P}$ attacking G, $\mathbf{Adv}_G^{\mathrm{prg}}(\mathcal{P})$ is defined as

$$\Pr\left[\, s \xleftarrow{\$} \{0,1\}^m : \mathcal{P}(G(s)) = 1 \,\right] - \Pr\left[\, y \xleftarrow{\$} \{0,1\}^l : \mathcal{P}(y) = 1 \,\right].$$

Here m is the seed length, and l is the number of pseudorandom bits generated.

3 PRG from Iterates

Most of the pseudorandom generators (PRGs) that we know today employ a general design technique: take a function that remains one-way on iterates, and iterate that function for a desired number of times, extracting hardcore bits at every iteration. Below we give a general theorem for the security of such PRGs. The theorem already exists in some form in the cryptographic literature (or, is implied from results in several papers, [Lev87, GKL88, HHR06a], to name a few), but we restate it and sketch its proof here for two main reasons. One is that the proof has evolved over time, starting from Levin's work [Lev87], followed by a proof sketch by Goldreich et al. (cf. Appendix B in [GKL88]), and the improved construction of hard-core predicate by Goldreich and Levin [GL89]. The second reason is that none of the prior works state the result in its entirety with a concrete security statement.

We will start with a more general definition that also captures the definition of pseudorandomness presented in Section 2.4. Let X and Y be random variables with equal output lengths. Let $\mathcal{D}$ be an adversary for distinguishing X from Y. The *indistinguishability* advantage of $\mathcal{D}$, $\mathbf{Adv}_{X,Y}^{\mathrm{ind}}(\mathcal{D})$ is defined as

$$\mathbf{Adv}_{X,Y}^{\mathrm{ind}}(\mathcal{D}) = \Pr\left[\, x \xleftarrow{\$} X : \mathcal{D}(x) = 1 \,\right] \; - \; \Pr\left[\, y \xleftarrow{\$} Y : \mathcal{D}(y) = 1 \,\right] \; .$$

For any adversary $\mathcal{P}$ and any pseudorandom generator G, $\mathbf{Adv}_{G,U_{|G|}}^{\mathrm{ind}}(\mathcal{P}) =$ $\mathbf{Adv}_G^{\mathrm{prg}}(\mathcal{P})$, where $U_{|G|}$ is a uniform distribution of size equal to the output size of G. The following theorem states the security of a PRG constructed from a function that is one-way on iterates, using the well known Goldreich-Levin hardcore bits (see Section 2.3 for details).

Theorem 3.1. Let $f : \{0,1\}^m \to \{0,1\}^n$ be any function, and for any $i \in \mathbb{N}$, let f^i denote its i^{th} iterate, defined arbitrarily but satisfying the following condition: given only $f^i(x)$ for any $x \in \{0,1\}^m$, $f^{i+1}(x)$ should be efficiently computable. For any $k \in \mathbb{N}$, if f^k is one-way on iterates[1], then for random $x, r \in \{0,1\}^m$, the random variables

$$X = \langle x, r\rangle \,\|\, \left\langle f^1(x), r\right\rangle \,\|\, \ldots \,\|\, \left\langle f^{k-1}(x),\ r\right\rangle \|r\| f^k(x) \quad \text{and} \quad Y = U_k \|r\| f^k(x)$$

are indistinguishable, where U_k is a uniform distribution of k bits. More formally, for an adversary $\mathcal{D}$ with running time $t_{\mathcal{D}}$, there exists an adversary $\mathcal{I}$ with running time $t_{\mathcal{I}}$, so that

$$\mathbf{Adv}_{X,Y}^{\mathrm{ind}}(\mathcal{D}) \leq 8k \cdot \max_{i=1}^{k} \left(\mathbf{Adv}_{f^i,f}^{\mathrm{htc}}(\mathcal{I})\right), \text{ and } t_{\mathcal{I}} = \mathcal{O}\left(m^3 \cdot t_{\mathcal{D}} \cdot \left(\mathbf{Adv}_{X,Y}^{\mathrm{ind}}(\mathcal{D})\right)^{-4}\right).$$

[1] f^k is one-way on iterates, if given $f^k(x)$ for a random $x \in \{0,1\}^m$, it is hard to compute $x' \in \{0,1\}^m$ such that $f(x') = f^k(x)$.

Informally, the above theorem states that $\langle x,r\rangle \,\|\, \left\langle f^1(x),r\right\rangle \|\ldots\| \left\langle f^{k-1}(x),\ r\right\rangle$ is pseudorandom, given r and $f^k(x)$.

Proof Sketch of Theorem 3.1. Any adversary trying to distinguish

$$X = \langle x,r\rangle \,\|\, \left\langle f^1(x),r\right\rangle \|\ldots\| \left\langle f^{k-1}(x),\ r\right\rangle \|r\| f^k(x) \quad \text{from} \quad Y = U_k\|r\|f^k(x)\ ,$$

can distinguish them only from the first k bits, because the remaining portion of X and Y are the same. The proof is presented in three parts. In the first part, we show that the indistinguishability of X and Y follows from the unpredictability of $X' = f^k(x)\|r\| \left\langle f^{k-1}(x),r\right\rangle \|\ldots\| \left\langle f^1(x),\ r\right\rangle \| \langle x,r\rangle$ (without loss of generality, the output of X is written in reverse order). Yao [Yao82] showed using hybrid argument that a sequence is indistinguishable from random, if and only if, it is hard to predict the next bit of the sequence, for every prefix of the sequence. Using this result, for an adversary $\mathcal{D}$ with running time $t_{\mathcal{D}}$, there exist an adversary $\mathcal{U}$ with running time $t_{\mathcal{U}}$, and $i \in [k-1]$, such that given $X'_i = f^k(x)\|r\| \left\langle f^{k-1}(x),r\right\rangle \|\ldots\| \left\langle f^i(x),\ r\right\rangle$, $\mathcal{U}$ can output the next bit $\left\langle f^{i-1}(x),\ r\right\rangle$, so that

$$\left(\Pr\left[\,\mathcal{U}\left(X'_i\right) = \left\langle f^{i-1}(x),\ r\right\rangle\,\right] - \frac{1}{2}\right) \geq \frac{\mathbf{Adv}^{\text{ind}}_{X,Y}(\mathcal{D})}{2k},\ \text{and } t_{\mathcal{U}} \approx t_{\mathcal{D}},$$

where $f^0(x) = x$.

In the second part, we show that given the adversary $\mathcal{U}$ with running time $t_{\mathcal{U}}$, there exists an adversary $\mathcal{A}$ with running time $t_{\mathcal{A}}$ that can distinguish the hardcore predicate $b\left(f^{i-1}(x),r\right) = \left\langle f^{i-1}(x),\ r\right\rangle$ from random, given $\widehat{f^i}(x,r) = \left(f^i(x),r\right)$, so that

$$\mathbf{Adv}^{\text{hcp}}_{\widehat{f^i},b}(\mathcal{A}) = \left(\Pr\left[\,\mathcal{U}\left(X'_i\right) = \left\langle f^{i-1}(x),\ r\right\rangle\,\right] - \frac{1}{2}\right),\ \text{and } t_{\mathcal{A}} \approx t_{\mathcal{U}}.$$

$\mathcal{A}$ is easy to construct. We know from the theorem that $f^{i+1}(x),\ldots,f^k(x)$ are efficiently computable from $f^i(x)$, so given $f^i(x)$ and r, $\mathcal{A}$ can compute X'_i, which it can use to run $\mathcal{U}$ to get back $\left\langle f^{i-1}(x),\ r\right\rangle$.

Finally, using Theorem 2.2, given the adversary $\mathcal{A}$ with running time $t_{\mathcal{A}}$, one can construct an adversary $\mathcal{I}$ with running time $t_{\mathcal{I}}$ that can compute $f^{i-1}(x)$, given $f^i(x)$, so that

$$\mathbf{Adv}^{\text{htc}}_{f^i,f}(\mathcal{I}) \geq \frac{\mathbf{Adv}^{\text{hcp}}_{\widehat{f^i},b}(\mathcal{A})}{4},\ \text{and } t_{\mathcal{I}} = O\left(m^3 \cdot t_{\mathcal{A}} \cdot \left(\mathbf{Adv}^{\text{hcp}}_{\widehat{f^i},b}(\mathcal{A})\right)^{-4}\right).$$

Putting things together, we have

$$\max_{i=1}^{k}\left(\mathbf{Adv}^{\text{htc}}_{f^i,f}(\mathcal{I})\right) \geq \frac{\mathbf{Adv}^{\text{ind}}_{X,Y}(\mathcal{D})}{8k}\ ,\ \text{and } t_{\mathcal{I}} = \mathcal{O}\left(m^3 \cdot t_{\mathcal{D}} \cdot \left(\mathbf{Adv}^{\text{ind}}_{X,Y}(\mathcal{D})\right)^{-4}\right)\ .$$

□

4 Our PRG Construction

We first define the *subset iterate*, a particular way to iterate a hash function on a subset of the actual domain. We use this in our PRG construction.

Subset Iterate. Let H be a hash function family, where each $h \in H$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$. For any $i \in \mathbb{N}$ and any $h \in H$, we define the i^{th} *subset iterate* of h, h_n^i, and denote the corresponding family by H_n^i. For $x \in \{0,1\}^n$, h_n^i is defined recursively as

$$\begin{aligned} h_n^1(x) &= h(x \| 0^{m-n}) \ , \\ h_n^i(x) &= h\left(h_n^{i-1}(x) \| 0^{m-n}\right) \quad \forall i > 1 \ . \end{aligned}$$

Any unambiguous padding (in place of zeroes, above) can be used to make the input to h of size m bits. For any $i \in \mathbb{N}$, we define the *one-way on iterates or owi* advantage of an adversary $\mathcal{I}$ attacking H_n^i, $\mathbf{Adv}^{\mathrm{owi}}_{H_n^i}(\mathcal{I})$ as

$$\Pr\left[\, h \xleftarrow{\$} H, x \xleftarrow{\$} \{0,1\}^n, x' \xleftarrow{\$} \mathcal{I}\left(h, h_n^i(x)\right) : \ h(x' \| 0^{m-n}) = h_n^i(x) \,\right] .$$

4.1 The Scheme

We now present our PRG construction. It requires a very short seed (twice the output size of the hash function).

Construction 4.1. Let H be a hash function family, where each $h \in H$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$. For any $l > 2n$, a random $h \in H$, which we assume becomes publicly known, and a random seed $s \in \{0,1\}^{2n}$, the pseudorandom generator G parses the input s as $x \| r$, such that both x and r are n-bit strings, and outputs

$$\langle x, r\rangle \, \| \, \left\langle h_n^1(x), r \right\rangle \| \ldots \| \left\langle h_n^{l-n-1}(x), r \right\rangle \| r \ ,$$

where for two bitstrings x $(= x_1 \| \ldots \| x_n)$ and r $(= r_1 \| \ldots \| r_n)$, $\langle x, r\rangle = \sum_{i=1}^n x_i \cdot r_i \pmod 2$ is their inner product modulo 2.

Note that the seed length of G is $2n$, and it is independent of the output length l. We now present the security analysis of the above construction.

4.2 Security

For simplicity, in the following theorem we assume that the underlying hash function family is regular. We will show how to relax this assumption to worst-case regularity in Section 6.

Theorem 4.2. Let H be a hash function family, where each $h \in H$ is a regular function from $\{0,1\}^m$ to $\{0,1\}^n$ and takes time t_H in computation. For any $l > 2n$, let G be the associated PRG, as defined by Construction 4.1. Then for an adversary $\mathcal{P}$ with running time $t_{\mathcal{P}}$, there exists an adversary $\mathcal{C}$ with running time $t_{\mathcal{C}}$, and $q = \lfloor t_{\mathcal{C}}/t_H \rfloor$, so that

$$\mathbf{Adv}^{\mathrm{prg}}_G(\mathcal{P}) \le 24 \cdot (l-n) \cdot \left[\binom{\lfloor q/(l-n) - 2 \rfloor}{2}^{-1} \cdot 2^n \cdot \left(\mathbf{Adv}^{\mathrm{cr}}_H(\mathcal{C})\right)^2 \right]^{\frac{1}{3}} ,$$

$$\text{and } t_{\mathcal{C}} = \max\left\{ \mathcal{O}\left(n^3 \cdot t_{\mathcal{P}} \cdot \left(\mathbf{Adv}^{\mathrm{prg}}_G(\mathcal{P})\right)^{-4}\right), 2(l-n)t_H \right\} .$$

Remark. The above advantage equation is meaningful only if $\mathbf{Adv}^{\mathrm{cr}}_{H}(\mathcal{C}) < 2^{-n/2}$. Also, as pointed out in the proof of Theorem 5.1, the above advantage expression can be made tighter (i.e., $(\mathbf{Adv}^{\mathrm{cr}}_{H}(\mathcal{C}))^2$ could be replaced with $\mathbf{Adv}^{\mathrm{tcr}}_{H}(\mathcal{C}_1) \cdot \mathbf{Adv}^{\mathrm{cr}}_{H}(\mathcal{C}_2)$ for $\mathcal{C}_1, \mathcal{C}_2$ attacking the target collision-resistance and collision-resistance of H, respectively), though the expression would become even more complicated. We present the proof in Section 5.

5 Proof of Theorem 4.2

We start with a short overview of the proof. The proof consists of two main parts: first we prove that the subset iterate used in the construction of our PRG is one-way on iterates (Theorem 5.1), and then we use the general result of Levin [Lev87] (Theorem 3.1) to show that our PRG is secure.

The subset iterate is constructed using a hash function. Now, suppose that we have an algorithm $\mathcal{I}$ that can invert the subset iterate, i.e. given $(h, h^i_n(x))$ for any $i \geq 2$, random h, and random x, it returns x' such that $h(x' \| 0^{m-n}) = h^i_n(x)$. Then, we can use $\mathcal{I}$ to break the target collision-resistance (TCR) of the underlying hash function. The challenge for the TCR attack (h, x) is used to compute $h(x)$, and then $(h, h(x))$ given to $\mathcal{I}$, and assuming that $h(x) \in \mathsf{Im}(h^i_n)$, with a very high probability the output of $\mathcal{I}$, x' (and x) is a collision instance for h. These steps are similar to those in the proof from [HHR06a].

Now, the main challenge is to show that with a non-negligible probability $h(x) \in \mathsf{Im}(h^i_n)$ (Lemma 5.4). The proof of the above is the crux and the main novelty of our analysis. We basically show that on iteration, the image set of the subset iterate shrinks by only a polynomial fraction, i.e. for any $i \geq 2$, $|\mathsf{Im}(h^i_n)|/|\mathsf{Im}(h^{i-1}_n)|$ is a polynomial fraction. For this purpose, we rely on Lemma 5.2, which says that the collision probability (in the birthday attack) of a subset iterate degrades only by a multiplicative factor of the square of the number of iterations. It may not be obvious, but the size of the image set and the collision probability of any function are closely related, which is precisely the reason why we are able to prove Lemma 5.2.

Before we provide the full security proof, we present some justification for our approach. One could argue that it is better to directly assume that the underlying CRHF is one-way on iterates (OWI), and be done with it. We dismiss this approach for the following reasons. First, the OWI property appears to be hard to test in practice. Unlike collision-resistance, we do not know of any experiment carried out by practitioners to measure the strength of a function against this kind of attack. Second, we do not know how does OWI security degrade with the number of iterations, which may be crucial in finding out exactly how many bits can be generated securely by any PRG.

In order to prove Theorem 4.2, we state the following theorem about the OWI security of the subset iterate used in the construction of our PRG. This theorem together with Theorem 3.1 (by substituting $(l-n)$ for k) will imply Theorem 4.2. (One might notice some inconsistencies between Theorem 5.1 and Theorem 3.1 in the sense that the underlying primitive in the former is a function family, while it is only a function in the latter. We note, however, that Theorem 3.1 is applicable without any change in the security reduction to our PRG construction from a hash function family.)

Theorem 5.1. Let H be a hash function family, where each $h \in H$ is a regular function from $\{0,1\}^m$ to $\{0,1\}^n$ and takes time t_H in computation. For any $i \in \mathbb{N}$, let H^i_n be the associated i^{th} subset iterate function family of H, as defined in Section 4. Then for an adversary $\mathcal{I}$ with running time $t_{\mathcal{I}}$, there exists an adversary $\mathcal{C}$ with running time $t_{\mathcal{C}}$, and $q = \lfloor t_{\mathcal{C}}/t_H \rfloor$, so that

$$\mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}) \leq 3 \cdot \left[\binom{\lfloor q/i - 2 \rfloor}{2}^{-1} \cdot 2^n \cdot \left(\mathbf{Adv}^{\mathrm{cr}}_H(\mathcal{C})\right)^2 \right]^{\frac{1}{3}}, \text{ and } t_{\mathcal{C}} = \max\left\{t_{\mathcal{I}}, 2it_H\right\}.$$

Proof. We construct an adversary $\mathcal{C}_1$ with running time $t_{\mathcal{C}_1} = t_{\mathcal{I}}$, for attacking the target collision-resistance of H. $\mathcal{C}_1$ is given a random $h \in H$ and a random $x \in \{0,1\}^m$. It runs the adversary $\mathcal{I}$ attacking one-wayness on iterates of H^i_n with input $(h, h(x))$. Let x' be the output of $\mathcal{I}$. If $x \neq x' \| 0^{m-n}$ and $h(x) = h(x' \| 0^{m-n})$, it returns $x' \| 0^{m-n}$.

We state the following three lemmas from which we will derive the inequality of Theorem 5.1. Lemma 5.2 gives an upper bound on the collison probability of birthday attack on the subset iterate of a hash function family. Lemma 5.3 which is similar to Claim 3.3 of [HHR06a], states that the set of inputs on which the adversary $\mathcal{I}$ succeeds reasonably well (better than one third of its advantage) is not small (at least two thirds of its advantage) in size. And, Lemma 5.4 which is similar to Lemma 3.4 of [HHR06a], states that the set of inputs that $\mathcal{I}$ should get in the actual experiment ($h^i_n(x)$ for a random $x \in \{0,1\}^n$) and the set of inputs that it actually gets in the above experiment simulated by $\mathcal{C}_1$ ($h(x)$ for a random $x \in \{0,1\}^m$), overlap for the most part.

Lemma 5.2. Let H be a hash function family, where each $h \in H$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$ and takes time t_H in computation. For any $i \in \mathbb{N}$, let H^i_n be the associated i^{th} subset iterate of H, as defined in Section 4. Then for any $q \geq 2i$, there exists an adversary $\mathcal{C}_2$ that runs in time (at most) $q \cdot t_H$, such that

$$\mathbf{CP}(H^i_n, 2) \leq \frac{\mathbf{Adv}^{\mathrm{cr}}_H(\mathcal{C}_2)}{\binom{\lfloor q/i - 2 \rfloor}{2}}.$$

Proof. We know that for any function f with output size n bits and balance measure $\mu(f)$, (upto constant factors) the collision probability for any $t \in \mathbb{N}$ trials, $\mathbf{CP}(f,t) = \binom{t}{2} \cdot 2^{-n\mu(f)}$, see [BK04] for details. Let $q' = \lfloor q/i - 2 \rfloor$, then

$$\mathbf{CP}(H^i_n, 2) = \frac{\mathbf{CP}(H^i_n, q')}{\binom{q'}{2}}.$$

Also, it is immediate that there exists an adversary $\mathcal{C}'$ running in time equivalent to q' computations of $h^i_n \in H^i_n$, such that

$$\mathbf{Adv}^{\mathrm{cr}}_{H^i_n}(\mathcal{C}') \geq \mathbf{CP}(H^i_n, q').$$

(In the worst case, $\mathcal{C}'$ could simply run the birthday attack with q' trials.)

Now, given $\mathcal{C}'$ we will construct the adversary $\mathcal{C}_2$ (from the lemma) that runs in time at most $q \cdot t_H$, so that

$$\mathbf{Adv}^{\mathrm{cr}}_H(\mathcal{C}_2) = \mathbf{Adv}^{\mathrm{cr}}_{H^i_n}(\mathcal{C}').$$

Note that for any $h^i_n \in H^i_n$, and any $x \neq x' \in \{0,1\}^n$, if $h^i_n(x) = h^i_n(x')$, then there exists $j < i$, such that $h^j_n(x) \neq h^j_n(x')$ and $h^{j+1}_n(x) = h^{j+1}_n(x')$. When $\mathcal{C}'$ returns (x, x'), $\mathcal{C}_2$ computes $y \leftarrow h^j_n(x)$, $y' \leftarrow h^j_n(x')$, and returns $(y \| 0^{m-n}, y' \| 0^{m-n})$. Recall that $y \neq y'$ and $h(y \| 0^{m-n}) = h(y' \| 0^{m-n})$, so the advantage of $\mathcal{C}_2$ is the same as that of $\mathcal{C}'$. Assuming that one computation of $h^i_n \in H^i_n$ requires the same time as i computations of $h \in H$, we have that the running time of $\mathcal{C}_2$ is at most $q \cdot t_H$ ($\geq (i \cdot q' + 2i) \cdot t_H$), because apart from running $\mathcal{C}'$ (which is equivalent to $i \cdot q'$ computations of

$h \in H$), $\mathcal{C}_2$ does $2j(< 2i)$ computations of $h \in H$ to compute its own output. Thus, $\mathbf{Adv}^{\mathrm{cr}}_{H}(\mathcal{C}_2)$ is equal to

$$\mathbf{Adv}^{\mathrm{cr}}_{H^i_n}(\mathcal{C}') \geq \mathbf{CP}(H^i_n, q') = \mathbf{CP}(H^i_n, 2) \cdot \binom{q'}{2} = \mathbf{CP}(H^i_n, 2) \cdot \binom{\lfloor q/i - 2 \rfloor}{2},$$

from which the lemma follows. □

Lemma 5.3. Let H be a hash function family, where each $h \in H$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$. For any $i \in \mathbb{N}$ and any $h \in H$, let h^i_n be the associated i^{th} subset iterate and H^i_n be the corresponding family, as defined in Section 4. For any adversary $\mathcal{I}$, consider the following probabilities in an experiment where a random $h \in H$ and a random $x \in \{0,1\}^n$ are picked, and a set $S \subseteq \mathsf{Im}(h^i_n)$ is defined as

$$S = \left\{ y \in \mathsf{Im}(h^i_n) : \Pr\left[\, h\left(\mathcal{I}\left(h, y\right)\right) = y \,\right] > \frac{1}{3} \cdot \mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}) \right\}.$$

Then,

$$\Pr\left[\, h^i_n(x) \in S \,\right] \geq \frac{2}{3} \cdot \mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}).$$

Proof. Assume (for contradiction) that in the above experiment, where a random $h \in H$ and a random $x \in \{0,1\}^n$ are picked, and a set $S \subseteq \mathsf{Im}(h^i_n)$ is defined as above, the following holds:

$$\Pr\left[\, h^i_n(x) \in S \,\right] < \frac{2}{3} \cdot \mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}).$$

Then we have

$$\begin{aligned} \mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}) &\leq \Pr\left[\, h^i_n(x) \in S \,\right] \cdot 1 + \Pr\left[\, h^i_n(x) \notin S \,\right] \cdot \frac{1}{3} \cdot \mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}) \\ &< \frac{2}{3} \cdot \mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}) + \frac{1}{3} \cdot \mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}), \end{aligned}$$

where the probabilities are over randomly picked $h \in H$ and $x \in \{0,1\}^n$.

S is the set of points where the adversary's advantage is greater than one-third of its actual (or, average) advantage. So, setting the adversary's advantage to be 1 for points inside S and one-third for points outside S, we get the first inequality. The second inequality follows directly from the above assumption.

Thus, $\mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}) < \mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I})$, which is a contradiction. □

Lemma 5.4. Let H be a hash function family, where each $h \in H$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$. For any $i \in \mathbb{N}$ and any $h \in H$, let h^i_n be the associated i^{th} subset iterate and H^i_n be the corresponding family, as defined in Section 4. Consider the following probabilities in an experiment where a random $h \in H$ and a random $x \in \{0,1\}^n$ are picked. If for any $T \subseteq \mathsf{Im}(h^i_n)$ and any $\delta \in [0,1]$,

$$\Pr\left[\, h^i_n(x) \in T \,\right] \geq \delta,$$

then

$$\Pr\left[\, h(x) \in T \,\right] \geq \frac{\delta^2}{2^{n+1} \cdot \mathbf{CP}(h^i_n, 2)}.$$

Proof. We will first compute a lower bound on the collision probability of h_n^i for two trials, $\mathbf{CP}(h_n^i, 2)$. Pick two elements x_1, x_2 uniformly at random from $\{0,1\}^n$, and then compute the probability that both $h_n^i(x_1), h_n^i(x_2)$ are equal and belong to the set T. This probability is clearly a lower bound on $\mathbf{CP}(h_n^i, 2)$, because T is a subset of $\mathsf{Im}(h_n^i)$. The probability that both $h_n^i(x_1), h_n^i(x_2) \in T$ is at least δ^2, and given that $h_n^i(x_1), h_n^i(x_2) \in T$, the probability that $h_n^i(x_1) = h_n^i(x_2)$ is at least $1/|T|$. The reason is that even though x_1, x_2 are uniformly random elements in $\{0,1\}^n$, $h_n^i(x_1), h_n^i(x_2)$ may not[2] be uniformly random elements in T. So, the probability that $h_n^i(x_1) = h_n^i(x_2)$ can be lower bounded by computing the probability of getting the same element, when two elements are picked (with replacement) uniformly at random from the set T. By simple probability theory, the probability of such an event is $1/|T|$. It may however be noted that in the above calculation, we are also counting trivial collisions, i.e. when $x_1 = x_2$. To compensate for this, we subtract 2^{-n} from the above probability. Hence,

$$\mathbf{CP}(h_n^i, 2) \geq \frac{\delta^2}{|T|} - \frac{1}{2^n}. \tag{1}$$

From Equation 1, we have

$$|T| \geq \frac{\delta^2}{\mathbf{CP}(h_n^i, 2) + 2^{-n}} \geq \frac{\delta^2}{2 \cdot \mathbf{CP}(h_n^i, 2)},$$

because $\mathbf{CP}(h_n^i, 2) \geq 2^{-n}$.

For any $h \in H$, $\mathsf{Im}(h_n^i) \subseteq \mathsf{Im}(h)$, and since $T \subseteq \mathsf{Im}(h_n^i)$, we have that $T \subseteq \mathsf{Im}(h)$. Also, since h is a regular function[3] and $\mathsf{Im}(h) \leq 2^n$, we have that

$$\Pr\left[\, h \xleftarrow{\$} H,\ x \xleftarrow{\$} \{0,1\}^m :\ h(x) \in T \,\right] = \frac{|T|}{|\mathsf{Im}(h)|} \geq \frac{|T|}{2^n}. \tag{2}$$

Thus, the statement of the lemma follows. □

Implication of Lemma 5.2, Lemma 5.3, and Lemma 5.4. Substituting S for T and $\frac{2}{3} \cdot \mathbf{Adv}^{\mathrm{owi}}_{H_n^i}(\mathcal{I})$ (from Lemma 5.3) for δ in Lemma 5.4, we get that for a random $h \in H$, adversary $\mathcal{I}$ and subset S as defined in Lemma 5.3

$$\begin{aligned} \Pr\left[\, h \xleftarrow{\$} H,\ x \xleftarrow{\$} \{0,1\}^m :\ h(x) \in S \,\right] \quad &\geq \quad \frac{\left(\frac{2}{3} \cdot \mathbf{Adv}^{\mathrm{owi}}_{H_n^i}(\mathcal{I})\right)^2}{2^{n+1} \cdot \mathbf{CP}(h_n^i, 2)} \\ &\geq \quad \frac{2^2}{3^2} \cdot \frac{\left(\mathbf{Adv}^{\mathrm{owi}}_{H_n^i}(\mathcal{I})\right)^2}{2^{n+1} \cdot \mathbf{CP}(h_n^i, 2)}. \end{aligned}$$

The above equation is a lower bound on the probability that for a random $h \in H$ and a random $x \in \{0,1\}^m$, $\mathcal{I}$'s challenge, $h(x)$ belongs to the subset S. From the description of $\mathcal{C}_1$, it is clear that $\mathbf{Adv}^{\mathrm{tcr}}_{H}(\mathcal{C}_1)$

$$\begin{aligned} &= \Pr\left[\, h \xleftarrow{\$} H, x \xleftarrow{\$} \{0,1\}^m, x' \xleftarrow{\$} \mathcal{I}(h, h(x)) : x \neq x' \| 0^{m-n} \bigwedge h(x' \| 0^{m-n}) = h(x) \,\right] \\ &= \Pr\left[\, h \xleftarrow{\$} H, x \xleftarrow{\$} \{0,1\}^m, x' \xleftarrow{\$} \mathcal{I}(h, h(x)) : x \neq x' \| 0^{m-n} \mid h(x' \| 0^{m-n}) = h(x) \,\right] \\ &\quad \times \Pr\left[\, h \xleftarrow{\$} H, x \xleftarrow{\$} \{0,1\}^m, x' \xleftarrow{\$} \mathcal{I}(h, h(x)) : h(x' \| 0^{m-n}) = h(x) \,\right]. \end{aligned}$$

[2] These elements are uniformly distributed, only if h_n^i is a regular function.

[3] We note that this is the only point in the proof that relies on the assumption that h is a regular function.

Let us denote the two probabilities in the last equation by P_1 and P_2, respectively. So, $\mathbf{Adv}_H^{\mathrm{tcr}}(\mathcal{C}_1) = P_1 \cdot P_2$. We know that

$$\begin{aligned} P_1 &\geq \Pr\left[\, z \xleftarrow{\$} \{0,1\}^{m-n} : \ z \neq 0^{m-n} \,\right] \\ &\geq \frac{2^{m-n}-1}{2^{m-n}} \geq \frac{1}{2}\,, \end{aligned}$$

because x is a uniformly random m-bit string, so the probability that the last $m-n$ bits of x are all 0's is at most 2^{n-m}. Also, from Lemma 5.3, we have that for a random $h \in H$, adversary $\mathcal{I}$ and subset S as defined in Lemma 5.3

$$\begin{aligned} P_2 &\geq \Pr\left[\, h \xleftarrow{\$} H,\ x \xleftarrow{\$} \{0,1\}^m : \ h(x) \in S \,\right] \cdot \frac{1}{3} \cdot \mathbf{Adv}_{H_n^i}^{\mathrm{owi}}(\mathcal{I}) \\ &\geq \frac{2^2}{3^2} \cdot \frac{\left(\mathbf{Adv}_{H_n^i}^{\mathrm{owi}}(\mathcal{I})\right)^2}{2^{n+1} \cdot \mathbf{CP}(h_n^i, 2)} \cdot \frac{1}{3} \cdot \mathbf{Adv}_{H_n^i}^{\mathrm{owi}}(\mathcal{I}) \\ &\geq \frac{2^2}{3^3} \cdot \frac{\left(\mathbf{Adv}_{H_n^i}^{\mathrm{owi}}(\mathcal{I})\right)^3}{2^{n+1} \cdot \mathbf{CP}(h_n^i, 2)}\,. \end{aligned}$$

The second inequality is from the lower bound on the probability that $\mathcal{I}$'s challenge $h(x)$ belongs to the subset S, as computed above. Thus,

$$\mathbf{Adv}_H^{\mathrm{tcr}}(\mathcal{C}_1) \geq \frac{\left(\mathbf{Adv}_{H_n^i}^{\mathrm{owi}}(\mathcal{I})\right)^3}{3^3 \cdot 2^n \cdot \mathbf{CP}(h_n^i, 2)}\,.$$

Combining the above inequality with Lemma 5.2, we have that for any $q \geq 2i$, there exists an adversary $\mathcal{C}_2$ that runs in time (at most) $q \cdot t_H$, such that

$$\mathbf{Adv}_H^{\mathrm{tcr}}(\mathcal{C}_1) \cdot \mathbf{Adv}_H^{\mathrm{cr}}(\mathcal{C}_2) \geq \frac{\binom{\lfloor q/i-2 \rfloor}{2}}{3^3 \cdot 2^n} \cdot \left(\mathbf{Adv}_{H_n^i}^{\mathrm{owi}}(\mathcal{I})\right)^3.$$

Recall that the running time of $\mathcal{C}_1$, $t_{\mathcal{C}_1} = t_{\mathcal{I}}$. Let $q = \max\{\lfloor t_{\mathcal{I}}/t_H \rfloor, 2i\}$, and let $\mathcal{C}$ denote the adversary (among $\mathcal{C}_1, \mathcal{C}_2$) with higher collision-resistance advantage, i.e. $\mathcal{C} = \mathcal{C}_1$ if $\mathbf{Adv}_H^{\mathrm{cr}}(\mathcal{C}_1) \geq \mathbf{Adv}_H^{\mathrm{cr}}(\mathcal{C}_2)$, otherwise $\mathcal{C} = \mathcal{C}_2$. (Note that we are getting rid of *target* collision-resistance advantage for a simpler theorem statement, albeit at a loss in the security guarantee) Then,

$$\left(\mathbf{Adv}_H^{\mathrm{cr}}(\mathcal{C})\right)^2 \geq \frac{\binom{\lfloor q/i-2 \rfloor}{2}}{3^3 \cdot 2^n} \cdot \left(\mathbf{Adv}_{H_n^i}^{\mathrm{owi}}(\mathcal{I})\right)^3.$$

The running time of $\mathcal{C}$, $t_{\mathcal{C}} = \max\{t_{\mathcal{I}}, 2it_H\}$, and hence, Theorem 5.1 follows. □

6 Relaxing the Regularity Assumption

We introduce a new notion that we call worst-case regularity. It captures the lower bound on the size of the smallest set of preimages of elements from the range of a function. The notion appears somewhat similar to the notions of "weakly regular" introduced by Goldreich et al. [GKL88] and "balance measure" introduced by Bellare and Kohno [BK04]. However, the reason for introducing a new notion (instead of working with the previous ones), is that it seems unlikely that one can find a tight relation between worst-case regularity and balance measure (or, weak regularity), and thus a tight bound for our theorem, for any general function (or, a CRHF in particular). The intuition behind this is that while worst-case regularity measures the lower bound on the size of preimages, the other two notions are related to the average of these sizes. We will first present the formal definition of worst-case regularity, and then adjust the statement of our main theorem for the case when the underlying CRHF is not necessarily regular.

Worst-case Regularity. Let F be a family of functions, where each $f \in F$ is a mapping from $\{0,1\}^m$ to $\{0,1\}^n$, and let $\alpha \in (0,1]$. We say that F is α-worst-case regular, if for all $f \in F$ and all $y \in \mathsf{Im}(f)$

$$|\mathsf{Preim}(f,y)| \geq \alpha \cdot 2^{m-n} \; .$$

For a completely regular function family, $\alpha = 1$.

As pointed out before, the only place where the regularity assumption is required for our proof is in Equation 2 of Lemma 5.4. So, we will first modify this equation and give justification for this modification, and then adjust our main theorem accordingly. For a not-necessarily regular function family Equation 2 changes as follows.

For any $h \in H$ and any $T \subseteq \mathsf{Im}(h)$, if H is α-worst-case regular, then

$$\Pr\left[\, h \xleftarrow{\$} H,\; x \xleftarrow{\$} \{0,1\}^m :\; h(x) \in T \,\right] \geq \frac{\alpha \cdot |T|}{2^n} \;, \tag{3}$$

where H is a hash function family as defined in Lemma 5.4. Since H is α-worst-case regular, the lower bound on the total size of the preimages of elements in T is $(\alpha \cdot 2^{m-n} \cdot |T|)$. So, when an element is picked uniformly at random from a set of size 2^m, the probability that it hits a subset of size $(\alpha \cdot 2^{m-n} \cdot |T|)$ is $\frac{\alpha \cdot |T|}{2^n}$.

Taking the above equation into account, we present the modified main theorem.

Theorem 6.1. [**Modified Theorem 4.2**] Let H be an α-worst-case regular hash function family, where each $h \in H$ is a function from $\{0,1\}^m$ to $\{0,1\}^n$ and takes time t_H in computation. For any $l > 2n$, let G be the associated pseudorandom generator, as defined by Construction 4.1. Then for an adversary $\mathcal{P}$ with running time $t_{\mathcal{P}}$, there exists an adversary $\mathcal{C}$ with running time $t_{\mathcal{C}}$, and $q = \lfloor t_{\mathcal{C}}/t_H \rfloor$, so that

$$\mathbf{Adv}^{\mathrm{prg}}_{G}(\mathcal{P}) \leq 24 \cdot (l-n) \cdot \left[\binom{\lfloor q/(l-n) - 2 \rfloor}{2}^{-1} \cdot \alpha^{-1} \cdot 2^n \cdot \left(\mathbf{Adv}^{\mathrm{cr}}_{H}(\mathcal{C})\right)^2 \right]^{\frac{1}{3}} ,$$

$$\text{and } t_{\mathcal{C}} = \max\left\{ \mathcal{O}\left(n^3 \cdot t_{\mathcal{P}} \cdot \left(\mathbf{Adv}^{\mathrm{prg}}_{G}(\mathcal{P})\right)^{-4} \right), 2(l-n)t_H \right\} .$$

7 Efficiency Improvement

Instead of extracting just one hardcore bit in an iteration, we can extract upto a constant factor of n hardcore bits, depending on the one-way on iterates security (and hence, the collision-resistance, see Theorem 5.1) of the underlying hash function. For the i^{th} iteration, let $\epsilon_i = \max_{\mathcal{I}} \left(\mathbf{Adv}^{\mathrm{owi}}_{H^i_n}(\mathcal{I}) \right)$ denote the one-way on iterates security of H^i_n, where the maximum is over all polynomial-time adversary $\mathcal{I}$. Then, one can extract $k_i = \mathcal{O}(\log \epsilon_i)$ hardcore bits in the i^{th} iteration without compromising the security of the PRG (cf. Theorem 2.5.6 in [Gol01]). The way to do it is to pick a random r (used with the iterated function's output in the inner product computation) of size $(n+k_i-1)$ bits, and return $\langle \cdot, r_1 \rangle \| \ldots \| \langle \cdot, r_k \rangle$, where "$\cdot$" is the output of the function in a particular iteration, and for $j \in [k]$, r_j is the first n bits of r starting from the j^{th} bit. Recall that the same r can be used in all the iterations, so a sufficiently large r ($< 2n$ bits) can be picked in the beginning and used throughout.

8 Conclusion

We propose a hash-function-based construction of a pseudorandom generator. Our scheme is similar to the "randomized iterate" construction of Haitner et al., but eliminates the need for the use of pairwise-independent functions on each iteration of the PRG. As a result, our PRG is significantly more efficient in terms of computation and the seed length. We first prove the security of our scheme assuming the underlying hash function is regular and collision-resistant, where the collision-resistance is required to be exponential. Then we show how to relax the regularity assumption on the hash function by introducing a new notion called worst-case regularity, which lower bounds the size of the smallest preimage set in a function. Unlike the previous similar schemes, our construction is accompanied by a concrete security statement.

Acknowledgements

We thank Zulfikar Ramzan for motivating us to work on the problem of constructing an efficient and theoretically sound PRG from hash functions, and for numerous useful discussions. We would also like to thank Mihir Bellare and anonymous reviewers for their valuable comments.

References

[BK04] M. Bellare and T. Kohno. Hash Function Balance and its Impact on Birthday Attacks. In *EUROCRYPT '04*, pages 401–418. Springer, 2004. Full version available at: http://eprint.iacr.org/2003/065. Cited on page 3, 5, 11, 14

[BR] M. Bellare and P. Rogaway. Chapter 5: Hash Functions. *Introduction to Modern Cryptography.* Available at: http://www-cse.ucsd.edu/users/mihir/cse207/w-hash.pdf. Cited on page 5, 6

[BM82] M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudo Random Bits. In *FOCS '82*, pages 112–117. IEEE, 1982. Cited on page 1, 2, 7

[DHY02] A. Desai, A. Hevia, and Y. L. Yin. A Practice-Oriented Treatment of Pseudorandom Number Generators. In *EUROCRYPT '02*, pages 368–383. Springer, 2002. Cited on page 2

[FIPS94] FIPS PUB 186-2, Digital Signature Standard, *National Institute of Standards and Technologies*, 1994. Cited on page 2, 4

[Gol01] O. Goldreich. Foundations of Cryptography - Volume 1. Cambridge University Press, 2001. Cited on page 4, 16

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4): 792–807, 1986. Cited on page 1

[GKL88] O. Goldreich, H. Krawczyk, and M. Luby. On the Existence of Pseudorandom Generators (Extended Abstract). In *FOCS '88*, pages 12–24. IEEE, 1988. Full version in *SIAM Journal of Computing*, 22(6): 1163–1175, 1993. Cited on page 2, 3, 4, 7, 14

[GL89] O. Goldreich and L. Levin. A Hard-Core Predicate for all One-Way Functions. In *STOC '89*, pages 25–32. ACM, 1989. Cited on page 3, 6, 7

[HHR06a] I. Haitner, D. Harnik, and O. Reingold. On the Power of the Randomized Iterate. In *CRYPTO '06*, pages 22–40. Springer, 2006. Full version available at: http://eccc.hpi-web.de/eccc-reports/2005/TR05-135. Cited on page 2, 3, 4, 6, 7, 10, 11

[HHR06b] I. Haitner, D. Harnik, and O. Reingold. Efficient Pseudorandom Generators from Exponentially Hard One-Way Functions. In *ICALP (2) '06*, pages 228–239. Springer, 2006. Cited on page 2

[HRV10] I. Haitner, O. Reingold, and S. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *STOC '10*, pages 437–446. ACM, 2010. Cited on page 2

[Has90] J. Håstad. Pseudo-Random Generators under Uniform Assumptions. In *STOC '90*, pages 395–404. ACM, 1990. Cited on page 2

[HILL99] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A Pseudorandom Generator from any One-way Function. In *SIAM Journal of Computing*, 28(4): 1364–1396, 1999. Cited on page 2

[Hol06] T. Holenstein. Pseudorandom Generators from One-Way Functions: A Simple Construction for Any Hardness. In *TCC '06*, pages 443–461. Springer, 2006. Cited on page 2

[ILL89] R. Impagliazzo, L. Levin, and M. Luby. Pseudo-random Generation from one-way functions (Extended Abstracts). In *STOC '89*, pages 12–24. ACM, 1989. Cited on page 2

[INW94] R. Impagliazzo, N. Nisan, and A. Wigderson. Pseudorandomness for network algorithms. In *STOC '94*, pages 356–364. ACM, 1994. Cited on page 2

[Lev87] L. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4): 357–363, 1987. Cited on page 3, 7, 10

[Nao91] M. Naor. Bit Commitment Using Pseudorandomness. *Journal of Cryptology*, 4(2): 151–158, 1991. Cite on page 1

[Nis92] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4): 449–461, 1992. Cited on page 2

[SHA3] SHA-3: Cryptographic Hash Algorithm Competition. *National Institute of Standards and Technology*, 2008. Available at: http://csrc.nist.gov/groups/ST/hash/sha-3/index.html. Cited on page 3

[Yao82] A. Yao. Theory and Applications of Trapdoor Functions (Extended Abstract). In *FOCS '82*, pages 80–91. IEEE, 1982. Cited on page 1, 2, 7, 8