Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 143 (2023)

# Counting Unpredictable Bits: A Simple PRG from One-way Functions

Noam Mazor * Rafael Pass †

September 22, 2023

**Abstract**

A central result in the theory of Cryptography, by Håstad, Imagliazzo, Luby and Levin [SICOMP'99], demonstrates that the existence one-way functions (OWF) implies the existence of pseudo-random generators (PRGs). Despite the fundamental importance of this result, and several elegant improvements/simplifications, analyses of constructions of PRGs from OWFs remain complex (both conceptually and technically).

Our goal is to provide a construction of a PRG from OWFs with a *simple proof of security*; we thus focus on the setting of *non-uniform* security (i.e., we start off with a OWF secure against non-uniform PPT, and we aim to get a PRG secure against non-uniform PPT).

Our main result is a construction of a PRG from OWFs with a self-contained, simple, proof of security, relying only on the Goldreich-Levin Theorem (and the Chernoff bound). Although our main goal is simplicity, the construction, and a variant there-of, also improves the efficiency—in terms of invocations and seed lengths—of the state-of-the-art constructions due to [Haitner-Reingold-Vadhan, STOC'10] and [Vadhan-Zheng, STOC'12], by a factor $O(\log^2 n)$.

The key novelty in our analysis is a generalization of the Blum-Micali [FOCS'82] notion of unpredictabilty—rather than requiring that every bit in the output of a function is unpredictable, we count how many unpredictable bits a function has, and we show that any OWF on $n$ input bits (after hashing the input and the output) has $n + O(\log n)$ unpredictable output bits. Such unpredictable bits can next be "extracted" into a pseudorandom string using standard techniques.

**Keywords:** Pseudorandom generator; one-way function; next-bit unpredictability

## 1 Introduction

Pseudorandom generators (PRGs) are one of the most fundamental cryptographic building blocks [BM82]. Roughly speaking, a PRG is a function taking a seed of length $n$ and expanding it into a

*Cornell Tech. E-mail: noammaz@gmail.com. Part of this work was done while at Tel Aviv University and while visiting the Simons Institute. Research partly supported by Israel Science Foundation grant 666/19, NSF CNS-2149305 and NSF CNS-2128519.

†Tel-Aviv University and Cornell Tech. E-mail: rafaelp@tau.ac.il. Part of this work was done while visiting the Simons Institute. Supported in part by NSF Award CNS 2149305, NSF Award SATC-1704788, NSF Award RI-1703846, AFOSR Award FA9550-18-1-0267, AFOSR Award FA9550-23-1-0387 and a JP Morgan Faculty Award. This material is based upon work supported by DARPA under Agreement No. HR00110C0086. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, DARPA or the AFOSR.



ISSN 1433-8092

longer string, of say, length $2n$, such that the output string is indistinguishable from random. While the existence of PRGs almost immediately implies the existence of one-way functions (OWF), it is significantly harder to show that OWFs imply the existence of PRGs. Indeed, the first construction of PRGs from OWFs was obtained in the seminal work by Håstad, Impagliazzo, Luby and Levin (HILL) [HILL99]. This beautiful work introduced a host of new notions and techniques and is a technical *tour-de-force*. To understand the importance of this result, let us remark that still today, known constructions of e.g., secure private-key encryption [GM84], commitment schemes [Nao91], zero-knowledge [GMW87], pseudorandom functions [GGM84] from the minimal assumption of OWFs, all pass through the notion of a PRG and the result of [HILL99].

Consequently, it would be desirable to come up with simpler constructions/proofs of the existence of PRGs from OWFs. Additionally, the PRGs construction of HILL, while asymptotically efficient, has a large polynomial running time. In particular, the PRG requires invoking the underlying OWFs $O(n^{11})$ times, where $n$ is the security parameter. Since then, several simplifications and improvements (in terms of the efficiency of the construction) were obtained by Holenstein [Hol06a], Haitner, Harnik and Reingold [HHR06], Haitner, Reingold and Vadhan [HRV13], Vadhan and Zheng [VZ12], leading up to constructions of PRGs from OWFs using only $\omega(n^3)$[1] *non-adaptive* invocations of the underlying OWF, and using a seed of length $\omega(n^4)$; additionally, Vadhan and Zheng [VZ12] show how to improve the seed length to $\omega(n^3 \log n)$, but at the price of using an adaptive construction. Finally, Haitner and Vadhan [HV17] obtained a construction with a simpler security proof (focusing only on the setting on non-uniform security), but which required $\omega(n^6 \log n)$ invocations of the OWF. But despite these beautiful works—and the intriguing new notions that they introduce—the security proofs involved remain quite complicated (even the simplest one with looser parameters in [HV17]).

**Our Results** The goal of this paper is to provide a simple, self-contained, proof of the existence of PRGs from any OWFs. Our proof relies only on standard results such as the Goldreich-Levin (GL) Theorem [GL89] and the Chernoff bound (and in case we want to optimize the seed-length using an adaptive construction, also the Leftover-hash Lemma (LHL) [HILL99]). The hope is that our proof will enable teaching the construction of a PRG from any OWF in graduate course in Cryptography.

Following Haitner and Vadhan [HV17], as our (main) goal is to present a security proof that is as easy as possible, we focus on the setting of *non-uniform* security (i.e., we start off with a OWF that is secure against non-uniform poly-time algorithms, and obtain a PRG secure against non-uniform polytime algorithms). (As we note in Appedix A, our proof of security also readily adapts to the uniform setting if we rely on Holenstein's Uniform Hard-core Lemma [Hol06b].)

Perhaps surprisingly, along the way, we manage to also improve the concrete efficiency of the PRG, obtaining a construction that only requires invoking the underlying OWF $\omega(n^3/\log^2 n)$ number of times, shaving a factor $\log^2 n$ from the best constructions [HRV13; VZ12], both in terms of number of invocations and seed length. (On a very high level, this improvement comes from the fact that we are relying on a simpler notion of "pseudo-entropy" and can next rely on a simpler 0-1 Chernoff bound instead of a "multi-valued" Chernoff bound as in [HRV13], which results in a tighter bound.)

[1] More formally, for any function $q(n) = \omega(n^3)$, there exists a construction of a PRG from OWFs that uses $q$ calls. HRV [HRV13] state their result with additional $\log n$ factor in both the seed length and the number of calls. However, the improved parameters can be easily deduced from their main theorem.

Our main result is a non-adaptive construction of a PRG from any OWFs, with a simple proof of security.

**Theorem 1.1** (Non-adaptive Construction of a PRG from OWFs)**.** *Assume the existence of a one-way function secure against non-uniform polynomial-time algorithms. Then there exists a PRG secure against non-uniform polynomial-time algorithms that non-adaptively invokes the underlying OWF $\omega(n^3/\log^2 n)$ times, and that has a seed of length $\omega(n^4/\log^2 n)$.*

As mentioned above, Vadhan and Zheng [VZ12] showed how to use adaptive calls to the underlying OWF to improve the seed length in the construction of [HRV13]; we note that the same method applies also to our construction enabling us again to shave $\log^2 n$ in the number of invocations of $f$ and the seed length; this time, however, we need to also rely also on the standard LHL of [HILL99].

**Theorem 1.2** (Adaptive PRG Construction from OWF with improved seed length)**.** *Assume the existence of a one-way function secure against non-uniform polynomial-time algorithms. Then there exist a PRG secure against non-uniform polynomial-time algorithms that adaptively invokes the underlying OWF $\omega(n^3/\log^2 n)$ times, and that has a seed of length $\omega(n^3/\log n)$.*

**On Concrete Efficiency (Exponentially-hard OWFs).** While shaving a $\log^2 n$ factor may not seem significant (when the running time is $O(n^3)$, this does make a significant difference in the regime of exponential security, or in the regime of concrete security. In particular, if we start off with an exponentially-secure OWF (i.e., a OWF secure against circuits of size $2^{\Omega(n)}$), then we can get a PRG that only invokes the OWF $\omega(\log n)$ times. This matches the bound of the best PRG from exponentially-secure OWFs from Haitner, Harnik and Reingold [HHR06], but only uses *non-adaptive* calls to the underlying OWF, whereas [HHR06] required adaptive calls, and may make the construction more feasible in practice. On the downside, our construction uses a seed of length $O(n^2)$, while [HHR06] uses seed of length $\omega(n \log n)$. We believe we can get a similar seed length using a better hash function, but we defer the details to a future version.

**Theorem 1.3.** *Assume the existence of a one-way function secure against circuits of size $2^{\Omega(n)}$. Then there exists a PRG secure against non-uniform polynomial-time algorithms that non-adaptively invokes the underlying OWF $\omega(\log n)$ times.*

We remark that the final PRG is also secure against exponential-size attackers, but only achieves negligible indisitinguishability gap.

**The Key Insight: Counting Unpredictable Bits** Starting with the work of HILL, the key method for constructing a PRG from OWFs is to start with a OWF and turning it into a generator of some "weak" form of pseudorandomness. Later these weak forms of pseudorandomness can be gradually amplified to achieve full pseudorandomness. Towards this, HILL introduced the notion of *pseudo-entropy*—roughly speaking, which requires a distribution to be indistinguishable from a distribution with some entropy. Haitner, Reingold and Vadhan [HRV13](HRV) improved and simplified the HILL construction by introducing and working with a relaxed notion of *next-block pseudo-entropy*, where following earlier notions of pseudorandomness by Shamir [Sha83] and Blum-Micali [BM82], we focus on the ability of a distinguisher to learn something about the next "block" in a sequence—and in more detail, this next block is required to have "high" pseudo-entropy in expectation over random blocks (see Appendix B for the formal definition).

In this work, we consider a strengthening of the HRV notion (which is incomparable to HILL notion): We start by going back to the "plain" notion of *unpredictability* from Blum-Micali [BM82]: Recall that we say that a function satisfies *unpredictability for the $i$-th bit*, if no non-uniform PPT attacker can guess the $i$-th bit of the output of the function on a random input given the first $i-1$ bits. We are interesting in *counting* how many unpredictable bits a function has. The simplest way to do this would be to say that a function has $k$ unpredictable bits if there exists a set $\mathcal{S}$ of indexes, with $|\mathcal{S}| \geq k$, such that for each $i \in \mathcal{S}$, the $i$-th bit is unpredictable for $f$.

Such a notion will be a bit too strong for our needs—we want to allow the indexes of the unpredictable bits to depend on the inputs. We do this by allowing the set $\mathcal{S}(x)$ of "unpredictable bits" to be a function of the input $x$, and we require that for each bit $i$ in the union of the support of $\mathcal{S}(U_n)$, we have that unpredictability of the $i$-th bit holds *conditioned* on sampling an input $x$ such that $i \in \mathcal{S}(x)$ (That is, unpredictability of bit $i$ holds whenever $i$ is in the set of "unpredictable bits"). To measure the number of such unpredictable bits, we simply consider the expected size of $\mathcal{S}(U_n)$: Roughly speaking, we say that a function has $k(\cdot)$ unpredictable bits if for every inverse polynomial $\epsilon$, there exists function $\mathcal{S}$ such that (1) the expected size of $\mathcal{S}(U_n)$ is at least $k(n)$, and (2) the bits specified by $\mathcal{S}$ are $\epsilon$-unpredictable. More formally,

**Definition 1.4.** *We say that a function $g : \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ has $k(\cdot)$-unpredictable bits if for every inverse polynomial $\epsilon(\cdot)$, there exists some $\mathcal{S}$ such that* **(1)** *for all $n \in \mathbb{N}$, $\mathrm{E}\left[\left|\mathcal{S}(U_{m(n)})\right|\right] \geq k(n)$, and* **(2)** *for all nonuniform* PPT $\mathsf{A}$*, every sufficiently large $n$, every $i \in \bigcup_{x \in \{0,1\}^{m(n)}} \mathrm{Supp}(\mathcal{S}(x))$,* $\mathsf{A}$ *distinguishes between*

- $\{x \leftarrow \{0,1\}^{m(n)} | i \in \mathcal{S}(x) \ : \ g(x)_{<i}, g(x)_i\}$
- $\{x \leftarrow \{0,1\}^{m(n)} | i \in \mathcal{S}(x) \ : \ g(x)_{<i}, U\}$

*with probability at most $\epsilon(n)$.*

For our purposes, we will need to generalize this definition to also apply to *families* of functions $\{g_h\}_{h \in \{0,1\}^*}$, where the above conditions hold for $g_h$ for a randomly sampled "key" $h$ (looking forward, for us, this key, will just be the description of a hash function based on inner-products mod 2).

## 2 Proof Overview

We present here our whole construction and provide a detailed proof overview—in essence, the below description provides *the whole proof* except that it omits standard hybrid arguments/reductions. (The formal proof in Sections 4 to 7 of course provides those details). We note that our construction closely follows the construction paradigm of HRV but due to the use of our notion of unpredictability, as opposed to next-bit pseudoentropy, we are able to simplify the analysis in the non-uniform setting (and improve its parameters).

Let $M$ be an $n \times n$ binary matrix, and we define the hash function $M(x) = Mx \mod 2$, where $x$ is interpreted as a binary vector. A simple form of the Leftover-hash Leamm (LHL) [HILL99] states that $\{M, M(X)_k)\}$ is $1/\mathrm{poly}(n)$-close to $\{M, U_k\}$, if $X$ has min-entropy $k + c \log n$ for a sufficiently large $c$, and when $M$ is sampled at random from the set of $n \times n$ binary matrices.[2]

[2] As an additional didactic contribution, we show that this simple form of the LHL follows as a direct corollary of the GL-theorem; while this observation may already be folklore, as far as we know, it has not been explicitly stated anywhere (more than for the case of extracting 1, or $O(\log n)$ bits).

- **Step 1: Unpredictability Generators from Regular OWF.** We start by showing how to turn any *regular* OWF—recall that for a $r(\cdot)$-regular OWF, each element in the support of the function on inputs of length $n$ has between $2^{r(n)-1}$ and $2^{r(n)}$ pre-images—into a function family that has $n+O(\log n)$ unpredictable bits; we refer to such function as an "unpredictability generator".

  For inputs of length $n$, the construction is defined as:

  $$g_M(x) = M(f(x))||M(x),$$

  where the "hash function" $M$ is described by an $n \times n$ binary matrix. In other words, we are applying $n$ GL-predicates to $f(x)$, and then the same $n$ GL predicates to $x$.[3]

  First, note that the since $f(\cdot)$ is $r(\cdot)$-regular, $f(U_n)$ has min-entropy $n - r(n) - 1$ and thus by the (simple) LHL the first $n - r(n) - O(\log n)$ bits of $M(f(x))$ are $1/\text{poly}(n)$-close to uniform, and thus unpredictable. Next, we want to argue that bits $n+1, \ldots, n+r(n)+c\log n$, for any $c$, also are unpredictable. Assume not; that is, there exists some efficient algorithm $P$ and some index $i$ such that

  $$P(f(x), M, M(x)_{<i}) = M(x)_i$$

  with inverse polynomial advantage. By the GL theorem, this means that there exists some PPT algorithm $E$ such that $E(f(x), M, M(x)_{<i}) = x$ with inverse polynomial probability, which in turn means that there exists some $E'$ that computes $x$ with probability $2^{-i}/\text{poly}(n) \geq 2^{-r(n)-O(\log n)}$ given just $f(x)$ (by guessing $M(x)_{<i}$). But since $f(x)$ has at least $2^{r(n)-1}$ pre-images, and all of which are equally likely, we have that the probability that $\Pr[E'(f(x)) = x] = \Pr\big[E'(f(x)) \in f^{-1}(f(x))\big]/2^{r(n)-1}$, and thus $E'$ inverts $f$ with inverse polynomial probability, which is a contradiction.

  Thus, we conclude that for every inverse polynomial $\epsilon$, there exists a set $\mathcal{S}$ of $\epsilon$-unpredictable indexes of size $[n - r(n) - O(\log n)] + [r(n) + c\log n] = n + (c - O(1))\log n$ (and which contains indexes $1, \ldots n - r(n) - O(\log n)$, as well as $n+1, \ldots n + r(n) + c\log n$).

  (Note that the set $\mathcal{S}$ depends on the unpredictability advantage $\epsilon$, but so far does not depend on the input $x$.)

- **Step 2: Unpredictability Generators from Any OWF.** We next show that the same construction actually works also for *any* (not necessarily regular) OWF. This directly follows from the observation that any OWF can be essentially split into regular OWFs on a partition of the input domain. In more detail, we can partition the input domain of the OWF into domains $D_1, D_2, \ldots$, such that (1) for each $r$, $f$ is $r$-regular when restricted to $D_r$—refer to this function as $f^r$, and (2) for each $r$ such that $D_r$ has inverse polynomial density in $\{0,1\}^n$,[4] we have that $f$ is one-way also on $D_j$. The set $D_j$ is simply the inputs $x \in \{0,1\}^n$ such that $f(x)$ has between

[3] We note that this step differs from the next-bit pseudo-entropy generator of HRV where $H$ is only applied to $x$ and not $f(x)$; this is the crucial difference that allows us to get unpredictability as opposed to next-bit pseudo-entropy. Additionally, we note that HRV has to work with a specially constructed hash function $H$ (based on concatenation of a Reed-Solomon Code and the Hadamard code); Haitner and Vadhan [HV17] showed how to just use the standard GL predicate, but this gave a final PRG construction with significantly worse parameters. Finally, Vadhan and Zheng [VZ12] show how to analyze also the construction without any hash function (achieving the same parameters as HRV), but this requires a much more complicated proof.

[4] Formally, $r = r(n)$ is a function of the input length $n$, and we here require the density condition to hold for all $n \in N$

$2^{j-1}$ and $2^j$ pre-images, and note that condition 2 follows directly from the assumption that $f$ is one-way.

Now, consider the set of "common" $r$'s such that $D_r$ has inverse polynomial density (and thus $f^r$ is one-way). By Step 1, we have shown that there exists some (appropriately large) set $\mathcal{S}_r$ of unpredictable indexes for every $f^r$ such that $r$ is "common", and for every such $x \in D_r$ we define $\mathcal{S}(x) = \mathcal{S}_r$. For the remaining $x$'s (that correspond to rare regularities), let $\mathcal{S}(x)$ simply be the empty set. By a union bound over the $n$ possible regularities, it follows that $\mathcal{S}(x)$ is set to the empty set only for a small fraction of inputs, and thus the expected size of $\mathcal{S}(U_n)$ is still $n + O(\log n)$.

To show that unpredictability holds, assume for contradiction that there exists some $i$ in the (union) of the support of $\mathcal{S}(U_n)$ such that bit $i$ can be predicted with inverse polynomial probability conditioned on $i \in \mathcal{S}(x)$ for infinitely many input lengths $n$. Then, note that $i \in \mathcal{S}(x)$ implies that $x \in D_r$ for some "common" regularity $r$, so we can always find some common $r$ (for each input length $n$) such that prediction also succeeds conditioned on $x \in D_r$ (for infinitely many input lengths), but this contradicts the unpredictability of bit $i$ for the function $f^r$.

- **Step 3: From Unpredictability to Random-Index Unpredictability.** In the next step, we consider a slightly stronger notion of unpredictability. Rather than bounding the expected size of the unpredictable set, the notion of $k(\cdot)$**-random bits unpredictability** requires that for each index $i$, we have that $\Pr[i \in \mathcal{S}(U_n)] \geq k(n)/\ell(n)$, where $\ell(\cdot)$ denotes the output length of the function. Note that by the linearity of expectation, this directly implies "plain" $k$-bits unpredictability (so this notion is a strengthening of "plain" unpredictability).

  To turn an unpredictability generator into a random-bit unpredictability generator, we rely on the same transformation as Haitner et al [HRV13] used in their "entropy equalization step" (and which was first used by [HRVW09]).

  Given a function $g : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ that has $k$-bit unpredictability, consider the "shifted" direct-product function $g'$:

$$g'(i, x^1, \ldots, x^r) = g(x^1)_{\geq i} || g(x^2) || \ldots || g(x^{r-1}) || g(x^r)_{<i}$$

  where $i \in [\ell(n)], x^j \in \{0,1\}^n$ (see Figure 1). That is, we apply the function $g$ on $r$ random inputs, output the concatenation (i.e., the direct product) and then simply truncate the $i-1$ bits from the beginning and the $\ell - (i-1)$ bits from the end, for a random $i$ (specified by the inputs).

  Note that each bit of $g'$ is part of the unpredictable set for $f$ with probability $k(n)/\ell(n)$. To see this, note that clearly a random index into $g$ is part of the unpredictable set for $g$ with probability $k(n)/\ell(n)$; but each bit of $g'$ has exactly the same distribution as a random bit of $g$. Thus, $g'$ has $(r-1)k(n)$ random unpredictable bits (while using a seed of length $n \cdot r + \log \ell(n)$).

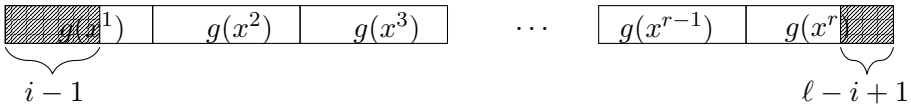


**Figure 1:** The construction of a function with random-bits-unpredictability from a function $g\colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ with bits-unpredictability. We take $r$ copies of $g$, and truncate the $i-1$ first bits and $\ell - i + 1$ last bits, such that the output, marked in white, is of length $(r-1)\ell$.

Finally, recall that the function obtained in Step 2 has a seed of length $n$, outputs $2n$ bits and has $(n + c \log n)$-bit unpredictability, for any $c$. If we plug in this function into $g$, we get a function with seed length $nr + O(\log n)$, output length $2(r-1)n$ and satisfying $(r-1)(n+c \log n)$-random bit unpredictability. To get "expansion" (i.e., more unpredictable bits than the seed length), we set $r = n/\log n$, which results in a function $g : \{0,1\}^{n^2/\log n + O(\log n)} \to \{0,1\}^{2n^2/\log n - 2n}$ that has $n^2/\log n + c \cdot n)$ random unpredictable bits, for any $c$.

- **Step 4: Pseudorandomness from Random-Bit Unpredictability.** In the final step, we show how to turn any generator of random-bit unpredictability into a standard PRG. The transformation is simple and goes back to HILL; it was also used by HRV to turn next-bit pseudoentropy into pseudorandomness, but for us, it will be even simpler (and due to this reason we can also improve the parameters from HRV).

  The transformation consists of doing a $t$-wise direct product of a function $g : \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ that has $k(n)$ random unpredictable bits, and then applying any (seeded) extractor *coordinate-wise* to the outputs of $g$. In more details, the $i$th block of the output will be $H(g(x^1)_i, g(x^2)_i, \ldots g(x^t)_i)$, where $H$ is an appropriate hash function, selected as part of the seed (and which also can be included in the output). (See Figure 2).

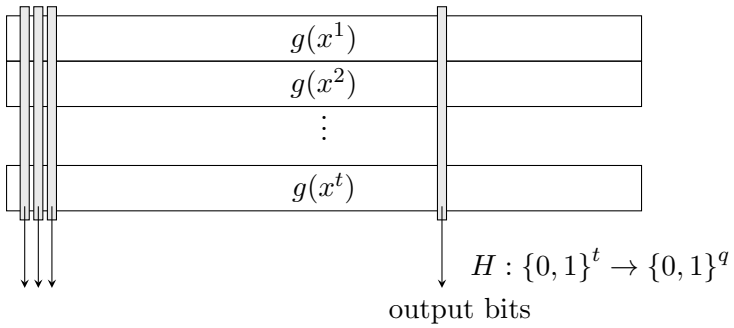


**Figure 2:** Extracting pseudoentropy from a function $g$ with random-bits unpredictability. We take $t$ copies of $g$ and apply a hash function (random matrix) on every column.

To analyze this construction, first note that by a standard hybrid argument, we simply need to show that each such output block $i$ is indistinguishable from uniform given the prefix up to block $i$. Next—and this is the key step—note that we can furthermore move to a hybrid where for each $j \in [t]$, we replace $g(x^j)_i$ with a random bit whenever $i$ is in the unpredictable set for $x^j$. Indistinguishability of the real experiment and this hybrid follows from the definition of unpredictability through an essentially standard hybrid argument, but there is an important subtlety: The set $S(x)$ is not efficiently computable, so in the hybrid argument it is not clear how to efficiently emulate the hybrids (and in particular, in Hybrid $j$, how to simulate all other "rows" $j' \neq j$). Since we are in the non-uniform setting, this issue, however, is easy to deal with: we can simply non-uniformly pick the best choices for those values.[5]

[5]In Appendix A, we explain how to also deal with this issue in the uniform setting, but this time we need to rely on Holenstein's uniform hard-core lemma[Hol06b]. In essence, using the hard-core lemma, we can argue that unpredictability also hold when getting access to an oracle for the set $S(x)$. Then the hybrids can be efficiently simulated given such an oracle.

Finally, by the Chernoff bound, we have that except with negligible probability, the number of "rows" $j$ such bit $i$ is unpredictable for $g$ is at least $t \cdot k(n)/\ell(n) - \sqrt{t\omega(\log n)}$, and thus all those bits will be uniform in the above hybrid. It follows that the min-entropy of the string on which we apply the extractor is $t \cdot k(n)/\ell(n) - \sqrt{t\omega(\log n)}$ and thus roughly this many bits may be extracted from each block; thus in total, we get $t \cdot k(n) - \ell\sqrt{t\omega(\log n)}$ pseudorandom bits.

The input is of length $t \cdot m(n)$, so we need to choose $t$ such that $t \cdot k(n) - \ell\sqrt{t\omega(\log n)} > t \cdot m(n)$, which yields $t \geq \omega(\log n)\ell^2/(k-m)^2$. Plugging in the construction from Step 3, we have that $k(n) = m(n) + O(n)$, $\ell(n) = O(n^2/\log n)$ which yields $t \geq \omega(\log n \ell^2/n^2) = \omega(n^2/\log n)$.

Note that the total seed length becomes $t \cdot m(n) + |H| = \omega(n^4/\log^2 n) + |H|$. If we rely on a random matrix as a hash function (and the above simplified LHL), its description length will be $t(n)^2 = n^4/\log^2 n$ (see Figure 3 for the complete construction).[6]

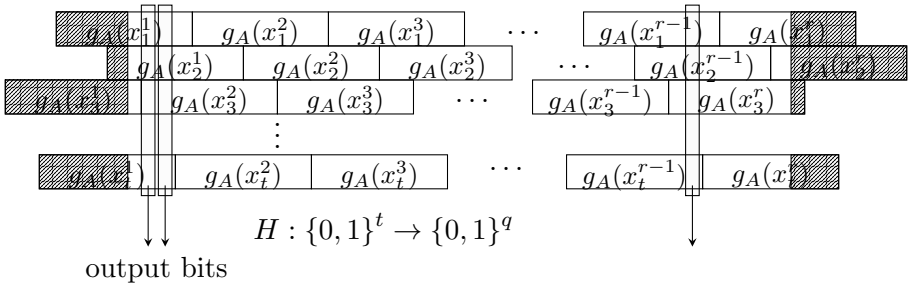


**Figure 3:** The non-adaptive PRG construction. There are $t \approx n^2/\log n$ rows, each row has $r \approx n/\log n$ i.i.d copies of $g_A(x) = (A(f(x)), A(x))$, shifted by a random offset. Every fully populated column, marked in white, is hashed by $H$.

**Further improving the seed length:** Vadhan and Zheng [VZ12] presented an elegant approach for shaving a factor $n/\log n$ in terms of the seed length in the construction of HRV. Their idea is to note that to compute "coordinate" $j$, we do not actually need to know the "seed" $x^{j'}$ to earlier coordinates $j' < j$, and thus we can take the input to coordinate $j-1$ from coordinate $j$ (while additionally outputting $O(\log n)$ bits). The same method can be applied to our construction and can be analyzed in a modular way. (We note that we do not claim any original contributions w.r.t. this step on top of [VZ12]; the only "novelty" here is the modular analysis of their construction.) Doing this yields an (adaptive) construction with seed length $\omega(n^3/\log n) + |H|$. So, to take advantage of this saving, we also need to have a hash function with a better description length. This is easily obtain by using a standard constructions of pair-wise independent hash functions (e.g., $h_{a,b}(x) = ax + b$ where the operations are over $\mathbb{F}_{2^n}$) and appealing to the standard LHL [HILL99] (instead of the above simplified form), which yields a description length of $O(t(n)) = O(n^2/\log^2 n)$, and thus a total seed length of $\omega(n^3/\log n)$

[6] In this step we save $\log^2 n$ factor over HRV. The reason is that we apply the Chernoff bound on a random variable that can only take zero-one values, while HRV consider the *sample entropy* of the next bit, which can take larger values.

# 3 Preliminaries

## 3.1 Notations

All logarithms are taken in base 2. We use calligraphic letters to denote sets and distributions, uppercase for random variables, and lowercase for values and functions. Let poly stand for the set of all polynomials. Let PPT stand for probabilistic poly-time, and n.u.-poly-time stand for non-uniform poly-time. An n.u.-poly-time algorithm $\mathsf{A}$ is equipped with a (fixed) poly-size advice string set $\{z_n\}_{n\in\mathbb{N}}$ (that we typically omit from the notation). Let neg stand for a negligible function. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. Given a vector $s \in \{0,1\}^n$, let $s_i$ denote its $i$-th entry, and $s_{1,\ldots,i}$ denote its first $i$ entries. For a function $f\colon \mathcal{D} \to \mathcal{R}$, and an image $y \in \mathcal{R}$, let $f^{-1}(y) = \{x \in \mathcal{D}\colon f(x) = y\}$.

The support of a distribution $\mathcal{P}$ over a finite set $\mathcal{S}$ is defined by $\mathrm{Supp}(\mathcal{P}) := \{x \in \mathcal{S} : \mathcal{P}(x) > 0\}$. Let $d \leftarrow \mathcal{P}$ denote that $d$ was sampled according to $\mathcal{P}$. Similarly, for a set $\mathcal{S}$, let $s \leftarrow \mathcal{S}$ denote that $s$ is drawn uniformly from $\mathcal{S}$. For $n \in \mathbb{N}$, we denote by $U_n$ the uniform distribution over $\{0,1\}^n$, and by $U$ the uniform distribution over $\{0,1\}$. The statistical distance (also known as, variation distance) of two distributions $\mathcal{P}$ and $\mathcal{Q}$ over a discrete domain $\mathcal{X}$ is defined by $\mathrm{SD}(\mathcal{P},\mathcal{Q}) := \max_{\mathcal{S}\subseteq\mathcal{X}} |\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S})| = \frac{1}{2}\sum_{x\in\mathcal{S}} |\mathcal{P}(x) - \mathcal{Q}(x)|$. For distribution ensembles $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}}$ and $\mathcal{Q} = \{\mathcal{Q}_n\}_{n\in\mathbb{N}}$ we write $\mathcal{P} \overset{c}{\approx}_\epsilon \mathcal{Q}$ if for every n.u.-poly-time $A$, for all but finitely many $n$'s, $|\Pr[A(\mathcal{P}_n) = 1] - \Pr[A(\mathcal{Q}_n) = 1]| \leq \epsilon(n)$. We write $\mathcal{P} \overset{c}{\approx} \mathcal{Q}$ if $|\Pr[A(\mathcal{P}_n) = 1] - \Pr[A(\mathcal{Q}_n) = 1]| = \mathrm{neg}(n)$ for every such $A$.

Lastly, we identify a matrix $M \in \{0,1\}^{n\times m}$ with a function $M\colon \{0,1\}^n \to \{0,1\}^m$ by $M(x) := x \cdot M \bmod 2$, thinking of $x \in \{0,1\}^n$ as a vector with dimension $n$.

## 3.2 One-Way Functions and Pseudorandom Generators

We now formally define basic cryptographic primitives. We start with the definition of one-way function.

**Definition 3.1** (One-way function)**.** *A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is called a* n.u-one-way function *if for every n.u.-poly-time algorithm $A$, there is a negligible function $\nu : \mathbb{N} \to [0,1]$ such that for every $n \in \mathbb{N}$*

$$\Pr_{x\leftarrow\{0,1\}^n}\left[A(f(x)) \in f^{-1}(f(x))\right] \leq \nu(n)$$

For simplicity, we assume that the one-way function $f$ is length-preserving. That is, $|f(x)| = |x|$ for every $x \in \{0,1\}^*$. This can be assumed without loss of generality, and is not crucial for our constructions.

In Section 4 we use one-way functions to construct PRGs. The latter is formally defined below.

**Definition 3.2** (Pseudorandom generator)**.** *Let $n$ be a security parameter. A polynomial-time computable function $G : \{0,1\}^n \to \{0,1\}^{m(n)}$ is called a* n.u-pseudorandom generator *if for every $n > 0$ it holds that $m(n) > n$ and, for every n.u.-poly-time algorithm $D$, there is a negligible function $\nu : \mathbb{N} \to [0,1]$ such that for every $n > 0$,*

$$\left|\Pr_{x\leftarrow\{0,1\}^n}[D(G(x)) = 1] - \Pr_{x\leftarrow\{0,1\}^{m(n)}}[D(x) = 1]\right| \leq \nu(n).$$

As in this paper we are focusing on the non-uniform setting, we will refer to n.u-one-way functions and n.u-PRGs simply by one-way functions and PRGs.

A key ingredient in the construction of PRG from one-way function is the Goldreich-Levin hardcore predicate. We will use the following version, which is a combination between Goldreich-Levin and Yao's distinguishing to prediction lemma ([Yao82]).

**Lemma 3.3** (Goldreich-Levin [GL89; Yao82])**.** *There exists an oracle-aided* PPT *$A$ such that the following holds. Let $n \in N$ be a number, and $\mathcal{Q}$ a distribution over $\{0,1\}^n \times \{0,1\}^*$, and let $D$ be an algorithm such that*

$$\Pr_{(x,z)\leftarrow\mathcal{Q},r\leftarrow\{0,1\}^n}[D(z,r,\mathrm{GL}(x,r))=1] - \Pr_{(x,z)\leftarrow\mathcal{Q},r\leftarrow\{0,1\}^n}[D(z,r,U)=1] \geq \alpha$$

*for some $\alpha$, where $\mathrm{GL}(x,r) := \langle x,r\rangle$ is the Goldreich-Levin predicate. Then*

$$\Pr_{(x,z)\leftarrow\mathcal{Q}}\Big[A^D(1^n,1^{\lceil 1/\alpha\rceil},z)=x\Big] \geq \alpha^3/8n.$$

### 3.3 Min-Entropy and Extraction

The min-entropy of a distribution $\mathcal{Q}$, denoted by $\mathrm{H}_\infty(\mathcal{Q})$ is defined by

$$\mathrm{H}_\infty(\mathcal{Q}) := -\log(\max_{q\in\mathrm{Supp}(\mathcal{Q})}\{\Pr[\mathcal{Q}=q]\}).$$

We will use the following simplified version of the leftover hash lemma, which shows that a random matrix is a strong extractor.

**Lemma 3.4** (Leftover hash lemma, simplified version)**.** *Let $n \in \mathbb{N}$, $\varepsilon \in [0,1]$, and let $X$ be a random variable over $\{0,1\}^n$. Let $M \leftarrow \{0,1\}^{n\times\ell}$ be a random matrix for $\ell \leq \mathrm{H}_\infty(X) - 3\log 1/\varepsilon - 4\log n - 4$. Then,*

$$SD((M,M(X)),(M,U_\ell)) \leq \varepsilon$$

*for $U_\ell$ being the uniform distribution over $\{0,1\}^\ell$.*

The above (simplified) version of the leftover hash lemma can be proven using GL. (The proof may be folklore, but we have not previously seen it in the literature.)

*Proof.* Let $\ell \leq \mathrm{H}_\infty(X) - 3\log(1/\epsilon) - 4\log n - 4 < n$, and let $M \leftarrow \{0,1\}^{n\times\ell}$ be a random matrix. Assume there exists an (inefficient) algorithm that distinguishes $M, M(X)$ from $M, U_\ell$ with advantage $\epsilon$. By a simple hybrid argument, there exists an (inefficent) distinguisher $D$ and an index $i \in [\ell]$, such that

$$\Pr[D(M,M(X)_{<i},M(X)_i)=1] - \Pr[D(M,M(X)_{<i},U)=1] \geq \epsilon/\ell \geq \epsilon/n.$$

Observe that $M(X)_i = \langle M_i, X\rangle$ is the GL hard-core predicate, and thus we get that there exists algorithm $A$ such that $\Pr[A(M,M(X)_{<i}) = X] \geq \epsilon^3/8n^4$. Consider the algorithm $A'$ that given $M$, guess $M(X)_{<i}$ and runs $A$. Clearly,

$$\Pr\big[A'(M)=X\big] \geq 2^{-i}\cdot\epsilon^3/8n^4 \geq 2^{-\ell}\cdot\epsilon^3/8n^4 > 2^{-\mathrm{H}_\infty(X)},$$

which is a contradiction, since $M$ is independent from $X$. □

We will also use the well-known Chernoff bound in our proof.

**Fact 3.5** (Chernoff bound)**.** *Let $A_1,...,A_n$ be independent random variables s.t. $A_i \in \{0,1\}$. Let $\widehat{A} = \Sigma_{i=1}^n A_i$ and $\mu = \mathrm{E}\Big[\widehat{A}\Big]$. For every $\epsilon \in [0,1]$ It holds that:*

$$\Pr\Big[\Big|\widehat{A}-\mu\Big| \geq \epsilon\cdot\mu\Big] \leq 2\cdot e^{-\epsilon^2\cdot\mu/3}.$$

# 4 Unpredictable Bits

In this section we define bits-unpredictability, which is the main building block in the construction. We will consider such a notion of unpredictability for families of functions.

**Definition 4.1** (Unpredictable bits)**.** *Let $m = m(n)$, $\ell = \ell(n)$, $\lambda = \lambda(n)$ and $k = k(n)$ be integer functions, and let $\epsilon = \epsilon(n) \in [0,1]$. We say that a function family $g = \left\{ g_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)} \right\}_{a \in \{0,1\}^{\lambda(n)}}$ has $(k,\epsilon)$-*bits-unpredictability *if for every $n \in \mathbb{N}$ and $x \in \{0,1\}^{m(n)}$, there exists a set $\mathcal{S}(x) \subseteq [\ell(n)]$, such that, for $X_n \leftarrow \{0,1\}^{m(n)}$ and $A \leftarrow \{0,1\}^{\lambda(n)}$:*

1. *For every $n$, $\mathrm{E}[|\mathcal{S}(X_n)|] \geq k(n)$, and,*
2. *for every sequence $\{i_n\}_{n\in\mathbb{N}}$ such that $i_n \in \bigcup_{x\in\{0,1\}^{m(n)}} \mathcal{S}(x)$,*

$$\left\{ (A, g_A(X_n)_{<i_n}, g_A(X_n)_{i_n})|_{i_n \in \mathcal{S}(X_n)} \right\}_{n\in\mathbb{N}} \overset{c}{\approx}_{\epsilon} \left\{ (A, g_A(X_n)_{<i_n}, U)|_{i_n \in \mathcal{S}(X_n)} \right\}_{n\in\mathbb{N}}.$$

*We say that $g$ has $k$-*bits-unpredictability *if it has $(k, n^{-c})$-bits-unpredictability for every $c \in \mathbb{N}$.*

We will also consider a stronger notion of unpredictability—called *$k$-random-bit unpredictability*, that requires each individual bit to be unpredictable with probability $k/\ell$ where $\ell$ is the output length.

**Definition 4.2** (Random bits unpredictability)**.** *Let $m = m(n)$, $\ell = \ell(n)$ and $k = k(n)$ be integer functions, and let $\epsilon = \epsilon(n) \in [0,1]$. We say that a function family $g = \left\{ g_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)} \right\}_{a \in \{0,1\}^{\lambda(n)}}$ has $(k,\epsilon)$-*random-bits-unpredictability *if it satisfies Definition 4.1 except that condition (1) is replaced by:*

1. *For every $i \in [\ell(n)]$, $\Pr[i \in \mathcal{S}(X_n)] \geq k(n)/\ell(n)$.*

*We say that $g$ has $k$-*random-bits-unpredictability *if it has $(k, n^{-c})$-bits-unpredictability for every $c \in \mathbb{N}$.*

# 5 OWFs ⇒ Unpredictable Bits

In this section, we prove the next theorem, which shows how to construct a function family with non-trivial bits-unpredictability from one-way functions.

**Theorem 5.1** (OWFs imply unpredictability)**.** *Let $f \colon \{0,1\}^n \to \{0,1\}^n$ be a one-way function and let $\mathcal{M}_n = \{0,1\}^{n\times n}$ be the family of all $n \times n$ matrices. Let $g = \left\{ g_M \colon \{0,1\}^n \to \{0,1\}^{2n} \right\}_{M \in \mathcal{M}_n}$ defined by*

$$g_M(x) = M(f(x)), M(x).$$

*Then $g$ has $(n + \log n)$-bits-unpredictability.*

We start with proving Theorem 5.1 for the case that $f$ is a regular one-way function on a partial domain. We later show how Theorem 5.1 follows from this case.

**Definition 5.2** (Regular one-way function, partial domain)**.** *For every $n \in \mathbb{N}$, let $\Omega_n \subseteq \{0,1\}^n$ be a set. An efficiently computable function $f\colon \Omega_n \to \{0,1\}^n$ is a* one-way function *if for every n.u.-poly-time algorithm $E$,*

$$\Pr\big[E(f(W_n)) \in f^{-1}(f(W_n))\big] = neg(n)$$

*for $W_n \leftarrow \Omega_n$. Such a function is $r = r(n)$* regular *if for every $n$ and $x \in \Omega_n$,*

$$2^r > \big|f^{-1}(f(x))\big| \geq 2^{r-1}.$$

**Lemma 5.3.** *Let $\epsilon = \epsilon(n) \in [0,1]$ and $r = r(n) \in \mathbb{N}$ be functions. Let $\Omega_n \subseteq \{0,1\}^n$ be a set such that $|\Omega_n| = \epsilon(n) \cdot 2^n$, and let $f\colon \Omega_n \to \{0,1\}^n$ be a $r$-regular one-way function. Let $M_n \leftarrow \mathcal{M}_n$ be a random matrix, and $Y_n = (M_n(f(W_n), M_n(W_n))$ for $W_n \leftarrow \Omega_n$. Then the following holds for every $c \in \mathbb{N}$:*

*For every $n \in \mathbb{N}$ there exists a set $\mathcal{S}_n \subseteq [2n]$ such that $|\mathcal{S}_n| = n + 4c\log n - \log(1/\epsilon)$, and for every sequence $\{i_n\}_{n\in\mathbb{N}}$ with $i_n \in \mathcal{S}_n$ it holds that*

$$\{(M_n, (Y_n)_{\leq i_n})\}_{n\in\mathbb{N}} \overset{c}{\approx}_{n^{-c}} \{(M_n, (Y_n)_{<i_n}, U)\}_{n\in\mathbb{N}}.$$

In the following, fix $c \in \mathbb{N}$, and let $r, \epsilon, \Omega_n, M_n, W_n$ and $Y_n$ be as defined in Lemma 5.3. For every $n \in \mathbb{N}$, let

$$\mathcal{S}_n = [n - r(n) - 8c\log n - \log(1/\epsilon(n))] \cup \{n < i \leq n + r(n) + 12c\log n\}. \tag{1}$$

Clearly, the size of $\mathcal{S}$ is $n + 4c\log n - \log(1/\epsilon)$, as stated in Lemma 5.3. To prove the lemma, we use the following two claims.

**Claim 5.4.** *For every $n \in \mathbb{N}$ and every $i \in [n - r(n) - 8c\log n - \log(1/\epsilon(n))]$ it holds that*

$$SD((M_n, M_n(f(W_n))_{\leq i}), (M_n, M_n(f(W_n))_{<i}, U)) \leq n^{-c},$$

*for $M \leftarrow \mathcal{M}_n$.*

*Proof.* To prove the claim we will show that $\mathrm{H}_\infty(f(W_n)) \geq n - r(n) - \log(1/\epsilon(n))$. The proof is then immediate from the leftover hash lemma (Lemma 3.4) and a simple hybrid argument, as by Lemma 3.4, $M_n(f(W_n))_{\leq i})$ is statistically close to $i$ uniform bits. To show the bound on the min-entropy of $f$, compute,

$$\begin{aligned}\mathrm{H}_\infty(f(X)) &= -\log(\max_y \Pr[f(X) = y]) \\ &\geq -\log(\max_y \frac{\big|f^{-1}(y)\big|}{|\Omega_n|}) > -\log(\frac{2^r}{\epsilon 2^n}) = n - r - \log(1/\epsilon)\end{aligned}$$

as stated. □

**Claim 5.5.** *For every sequence $\{i_n\}_{n\in\mathbb{N}}$, with $i_n \in [r(n) + 12c\log n]$ it holds that*

$$\{(M_n, f(W_n), M_n(W_n)_{\leq i_n})\}_{n\in\mathbb{N}} \overset{c}{\approx} \{(M_n, f(W_n), M_n(W_n)_{<i_n}, U)\}_{n\in\mathbb{N}}.$$

*Proof.* Assume towards a contradiction that the claim does not hold. That is, there exists some algorithm $E$ and a sequence $\{i_n\}_{n\in\mathbb{N}}$, such that

$$|\Pr[E(M_n, f(W_n), M_n(W_n)_{\leq i_n}) = 1] - \Pr[E(M_n, f(W_n), M_n(W_n)_{<i_n}, U) = 1]| \geq n^{-d}$$

for some constant $d \in \mathbb{N}$ and for infinitely many $n$'s. Fix such $n \in \mathbb{N}$, and omit it from the notation. Let $i^* = i_n$, assume without loss of generality that

$$\Pr[E(M, f(W), M(W)_{\leq i^*}) = 1] - \Pr[E(M, f(W), M(W)_{<i^*}, U) = 1] \geq n^{-d}.$$

By Lemma 3.3 (Goldreich-Levin), the existence of $E$ implies that there exists an algorithm $E'$ such that

$$\Pr\left[E'(1^{n^c}, M, f(W), M(W)_{<i^*}) = W\right] \geq n^{-2d}.$$

Let $\widehat{E}$ be the algorithm that on input $f(W)$, sample $M \leftarrow \{0,1\}^{n\times n}$, and guess $r \leftarrow \{0,1\}^{i^*-1}$. It then outputs $E'(1^{n^c}, M, f(W), r)$. Since $\Pr[M(W)_{<i^*} = r] = 2^{-i^*+1}$, it holds that

$$\Pr\left[\widehat{E}(f(W)) = W\right] \geq n^{-2d} \cdot 2^{-i^*+1}. \tag{2}$$

Since $f$ has at least $2^{r-1} \geq 2^{i^*-12c\log n}$ pre-images, it holds that

$$\begin{aligned}\Pr\left[\widehat{E}(f(W)) \in f^{-1}(f(W))\right] &\geq 2^{r-1} \cdot \Pr\left[\widehat{E}(f(W)) = W\right] \\ &\geq 2^{i^*-12c\log n-1} \cdot \Pr\left[\widehat{E}(f(W)) = W\right].\end{aligned} \tag{3}$$

Combining Equations (2) and (3), we get that

$$\Pr\left[\widehat{E}(f(W)) = W\right] \geq n^{-2d-12c-1}$$

which is a contradiction since $f$ is a one-way function. □

## 5.1 Proving Lemma 5.3.

We are now ready to prove Lemma 5.3.

*Proof of Lemma 5.3.* Assume towards a contradiction that the lemma does not hold. That is, there exists a constant $c$, a n.u.-poly-time algorithm $E$ and a sequence $\{i_n\}_{n\in\mathbb{N}}$ with $i_n \in \mathcal{S}_n$ such that,

$$|\Pr[E(M_n, (Y_n)_{\leq i_n}) = 1] - \Pr[E(M_n, (Y_n)_{<i_n}, U) = 1]| > n^{-c}$$

for infinitely many $n$'s, where $\mathcal{S}_n$ is the set defined in Equation (1) with respect to the constant $c$. We conclude the proof by the observation that, either for infinitely many such $n$'s it holds that $i_n \leq n$, or for infinitely many such $n$'s $i_n > n$. In the first case, $E$ contradicts Claim 5.4. In the second, $E$ contradicts Claim 5.5 by a simple data-processing argument. □

### 5.2 Proving Theorem 5.1.

*Proof of Theorem 5.1.* Fix $c \in \mathbb{N}$. The proof follows by the observation that every one-way function is a combination of regular one-way functions. Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a one-way function, and for every $x \in \{0,1\}^n$, let $D_f(x) = \lfloor \log |f^{-1}(f(x))| \rfloor$. For every $n \in N$ and $r \in [n]$, let $\Omega_n^r = \{x \in \{0,1\}^n \colon D_f(x) = r\}$. Let $\epsilon^r(n) = |\Omega_n^r|/2^n$ and let $\mathcal{S}_n^r$ be the set $\mathcal{S}_n$ promised by Lemma 5.3 with respect to $r$. Observe that for every function $r = r(n)$, $f^r\colon \Omega_n^r \to \{0,1\}^n$ is $r$-regular function. Moreover, for every such $r$ with $\epsilon^r(n) \geq n^{-2c}$ for every $n \in \mathbb{N}$, it holds that the function $f^r$ is one-way. Indeed, an algorithm $E$ that inverts $f^r$ with probability $\alpha(n)$ inverts $f$ with probability at least $\alpha(n) \cdot \Pr[D_f(X_n) = r(n)] \geq \alpha(n) \cdot n^{-2c}$.

For $x \in \{0,1\}^n$, let $\mathcal{S}(x) = \mathcal{S}_n^{D_f(x)}$ if $\epsilon^{D_f(x)}(n) \geq n^{-2c}$ or $\emptyset$ otherwise. In the following we show that

$$\Pr[|\mathcal{S}(X_n)| < n + 2c \log n] \leq n^{-c}. \tag{4}$$

It then follows that

$$\mathrm{E}[|\mathcal{S}(X_n)|] \geq (n + 2c \log n)(1 - n^{-c}) \geq n + c \log n$$

as stated. To see Equation (4), let $\mathcal{G}_n = \left\{r \in [n] \colon \epsilon^r(n) \geq n^{-2c}\right\}$. By definition of $\mathcal{S}$ and $\mathcal{G}_n$ we get that for every $r \in \mathcal{G}_n$ and $x$ with $D_f(x) = r$

$$\begin{aligned} |\mathcal{S}(x)| &\geq (n + 4c \log n - \log(1/n^{-2c})) \\ &= n + 2c \log n. \end{aligned}$$

Thus, $\Pr[|\mathcal{S}(X_n)| < n + 2c \log n] \leq \Pr[D_f(X_n) \notin \mathcal{G}_n]$, and it is enough to bound $\Pr[D_f(X_n) \notin \mathcal{G}_n]$. By union bound, as $D_f(x)$ can get at most $n$ values, and for every $r \notin \mathcal{G}_n$ it holds that $\Pr[D_f(X_n) = r] \leq n^{-2c}$, we get that $\Pr[D_f(X_n) \notin \mathcal{G}_n] \leq n^{-2c} \cdot n \leq n^{-c}$, as we wanted to show.

Next, assume toward a contradiction that $g$ has no $(n + \log n, n^{-c})$-bits-unpredictability with respect to the above sets $\mathcal{S}(x)$. Namely, there exists an algorithm $E$ such that

$$|\Pr[E(M_n, g_{M_n}(X_n)_{\leq i_n}) = 1 \mid i_n \in \mathcal{S}(X_n)]\Pr[E(M_n, g_{M_n}(X_n)_{<i_n}, U) = 1 \mid i_n \in \mathcal{S}(X_n)]| > n^{-c}$$

for some sequence $\{i_n\}_{n\in\mathbb{N}}$ and for infinite many $n$'s. Below we show how to construct a regular one-way function on partial domain $f^{r^*}$, such that $E$ contradicts Lemma 5.3 with respect to $f^{r^*}$. To do so, fix such $n$ and observe that, by an averaging argument, there exists some $r^* \in [n]$ such that $i_n \in \mathcal{S}_n^{r^*}$, and,

$$\begin{aligned} |&\Pr[E(M_n, g_{M_n}(X_n)_{\leq i_n}) = 1 \mid i_n \in \mathcal{S}(X_n), D_f(X_n) = r^*] \\ &\quad - \Pr[E(M_n, g_{M_n}(X_n)_{<i_n}, U) = 1 \mid i_n \in \mathcal{S}(X_n), D_f(X_n) = r^*]| > n^{-c}. \end{aligned}$$

Since $\mathcal{S}(X_n)$ is determined by $D_f(X_n)$, we get that,

$$|\Pr[E(M_n, g_{M_n}(X_n)_{\leq i_n}) = 1 \mid D_f(X_n) = r^*] - \Pr[E(M_n, g_{M_n}(X_n)_{<i_n}, U) = 1 \mid D_f(X_n) = r^*]| > n^{-c}.$$

Lastly, observe that the event $D_f(X_n) = r$ does not depend on $M_n$, and only depend on $f(X_n)$. Thus, we can write the above as

$$\left|\Pr_{x \leftarrow \Omega_n^{r^*}}[E(M_n, g_{M_n}(x)_{\leq i_n}) = 1] - \Pr_{x \leftarrow \Omega_n^{r^*}}[E(M_n, g_{M_n}(x)_{<i_n}, U) = 1]\right| \geq n^{-c}.$$

Moreover, since $i_n \in \mathcal{S}_n^{r^*}$, it holds that $\epsilon^{r^*}(n) \geq n^{-2c}$. For every $n$ let $r^*(n)$ be as described above (or, if no such $r^*$ exists, let $r^*(n)$ be arbitrary $r$ with $\epsilon^r(n) \geq n^{-2c}$).[7] The above is a contradiction to Lemma 5.3, as by construction $f^{r^*} \colon \Omega_n^{r^*} \to \{0,1\}^n$ is a regular one-way function (note that, while $r^*$ may not be an efficiently computable function, $f^{r^*}$ is). □

# 6 Bits Unpredictability ⇒ Random Bits Unpredictability

The next theorem, proven below, shows how to convert bits unpredictability to random bits unpredictability.

**Theorem 6.1** (Bits unpredictability to random bits unpredictability)**.** *Let $m = m(n)$, $\ell = \ell(n)$, $\lambda = \lambda(n)$ and $k = k(n)$ be integer functions and let $g = \left\{ g_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)} \right\}_{a \in \{0,1\}^{\lambda(n)}}$ be a function family with $k$-bits-unpredictability. Then, for every polynomial $r = r(n)$, the function family $g^r = \left\{ g_a^r \colon [\ell(n)] \times (\{0,1\}^{m(n)})^{r(n)} \to \{0,1\}^{(r(n)-1)\ell(n)} \right\}_{a \in \{0,1\}^{\lambda(n)}}$ defined by*

$$g_a^r(i, x^1, \ldots, x^r) = g_a(x^1)_{\geq i}, g_a(x^2), \ldots, g_a(x^{r-1}), g_a(x^r)_{<i}$$

*has $(r(n)-1)k(n)$-random-bits unpredictability.*

We get the following corollary, on construction of random-bits unpredictability from a one-way function.

**Corollary 6.2** (OWF to random-bits unpredictability)**.** *Let $f \colon \{0,1\}^n \to \{0,1\}^n$ be a one-way. Then there exists an efficiently computable function family $g' = \left\{ g'_a \colon \{0,1\}^{m'(n)} \to \{0,1\}^{\ell'(n)} \right\}_{a \in \{0,1\}^{\lambda(n)}}$ with $k'$-random-bits unpredictability, for $m'(n) = O(n^2/\log n)$, $\ell'(n) = O(n^2/\log n)$, $\lambda(n) = n^2$ and $k'(n) \geq m'(n) + n$.*

*Moreover, the construction uses $r(n)$ non-adaptive calls to $f$.*

*Proof.* Let $g$ be the function family defined in Theorem 5.1. Let $r(n) = \lceil 2n/\log n \rceil + 3$, and let $g' = g^r$, as defined in Theorem 6.1. It holds that $m'(n) = \lceil \log n \rceil + n \cdot r(n) = O(n^2/\log n)$, and $\ell'(n) = 2n \cdot (r(n)-1) = O(n^2/\log n)$. Moreover, by Theorem 6.1,

$$k'(n) = (r(n)-1)(n + \log n) = \log n + n \cdot r(n) + \log n \cdot (r(n)-2) - n \geq m'(n) + 2n - n.$$

□

## 6.1 Proving Theorem 6.1

*Proof of Theorem 6.1.* Let $\ell, m, \lambda, k$ and $g$ be as in Theorem 6.1, and fix a polynomial $r = r(n)$ and a constant $c$. In the following we prove that $g^r$ has $((r-1)k, n^{-c})$-random bits unpredictability. For every $n \in \mathbb{N}$ and $x \in \{0,1\}^{m(n)}$, let $\mathcal{S}^g(x)$ be the set promised by Definition 4.1 with respect to the $(k, n^{-c})$-bits-unpredictability of $g$.

[7] That is, for every $n$ for which $E$ distinguishes $g_{M_n}(X_n)_{\leq i_n}$ from $(g_{M_n}(X_n)_{<i_n}, U)$ given $i_n \in \mathcal{S}(X^n)$, we define $r^*(n)$ as described, and for all other $n$'s we define $r^*(n)$ arbitrarily such that $\Pr[D_f(X_n) = r^*(n)]$ is noticeable.

For $i \in [\ell(n)]$ and $x^1, \ldots, x^r \in (\{0,1\}^{m(n)})^{r(n)}$, define the set

$$\mathcal{S}(i, x^1, \ldots, x^r) = \Big(\bigcup_{j \in [r]} \big\{z + (j-1)n - (i-1) \colon z \in \mathcal{S}^g(x^j)\big\}\Big) \bigcap [\ell(n) \cdot (r(n)-1)].$$

let $X_n^1, \ldots, X_n^r \leftarrow \{0,1\}^{m(n)}$ and $I_n \leftarrow [\ell(n)]$. Clearly, for every $i \in [\ell(n) \cdot (r(n)-1)]$, it holds that

$$\begin{aligned}\Pr\big[i \in \mathcal{S}(I_n, X_n^1, \ldots, X_n^r)\big] &= \Pr[(i + I_n \bmod \ell(n)) \in \mathcal{S}^g(X_n)] \\ &= \frac{\mathrm{E}[|\mathcal{S}^g(X_n)|]}{\ell(n)} \geq \frac{k(n)}{\ell(n)} = \frac{(r(n)-1)k(n)}{(r(n)-1)\ell(n)}.\end{aligned}$$

Let $\mathcal{S}_n = \mathcal{S}(I_n, X_n^1, \ldots, X_n^r)$. Assume toward a contradiction that $g^r$ does not have $(r-1)k$-random-bits unpredictability with respect to the above set $\mathcal{S}$. That is, there exists an algorithm $E$ and an index $z = z(n) \in [\ell \cdot (r-1)]$, such that, for $A_n \leftarrow \{0,1\}^{\lambda(n)}$,

$$\begin{aligned}|&\Pr\big[E(A_n, g_{A_n}^r(I_n, X_n^1, \ldots, X_n^r)_{\leq z}) = 1 \mid z \in \mathcal{S}_n\big] \\ &\quad - \Pr\big[E(A_n, g_{A_n}^r(I_n, X_n^1, \ldots, X_n^r)_{<z}, U) = 1\big] \mid z \in \mathcal{S}_n| \geq n^{-c}.\end{aligned}$$

For infinitely many $n$'s. Fix such $n$ and omit $n$ from the notation. By an averaging argument, there exists an index $i^* \in [\ell(n)]$ such that

$$|\Pr\big[E(A, g_A^r(i^*, X^1, \ldots, X^r)_{\leq z}) = 1 \mid z \in \mathcal{S}\big] - \Pr\big[E(A, g_A^r(i^*, X^1, \ldots, X^r)_{<z}, U) = 1\big] \mid z \in \mathcal{S}| \geq n^{-c}.$$

Recall that $g_A^r$ is produced by $r$ blocks of the form $g_A(X^j)$ (with a random shift). Let $s = \lceil \frac{z+(i^*-1)}{\ell} \rceil$ be the index of the block in which the index $z$ belongs to, and $i$ be the index of $z$ inside the block. That is, $s$ and $i$ are such that $g_A^r(i^*, X^1, \ldots, X^r)_{\leq z} = g_A(X^1)_{\geq i^*}, g_A(X^2), \ldots, g_A(X^s)_{\leq i}$. Consider the algorithm $E'$ that, given $a, g_a(x)_{\leq i}$ and a bit $b$, sample $X^1, \ldots, X^{s-1}$ uniformly at random and executes $E(a, g_a(X^1)_{>i^*}, g_a(X^2), \ldots, g_a(x)_{<i}, b)$.

Observe that,

$$\begin{aligned}&|\Pr\big[E'(A, g_A(X)_{<i}, g_A(X)_i) = 1 \mid i \in \mathcal{S}^g(X)\big] - \Pr\big[E'(A, g_A(X)_{<i}, U) = 1 \mid i \in \mathcal{S}^g(X)\big]| \\ &= |\Pr\big[E(A, g_A(i^*, X^1, \ldots, X^r)_{\leq z}) = 1 \mid z \in \mathcal{S}\big] - \Pr\big[E(A, g_A(i^*, X^1, \ldots, X^r)_{<z}, U) = 1 \mid z \in \mathcal{S}\big]| \\ &\geq n^{-c}.\end{aligned}$$

The above is a contradiction to the $(k, n^{-c})$-bits unpredictability of $g$, since by assumption, it holds for infinitely many $n$'s □

# 7 Extracting Pseudorandomness and the Main Theorem

In this section we prove Theorem 7.1, which is the last step in our main construction. Theorem 7.1 shows how to extract pseudorandomness from random bits unpredictability.

**Theorem 7.1** (Extracting from random bits unpredictability)**.** *Let $s = \omega(1)$, $m = m(n)$, $\ell = \ell(n)$, $\lambda = \lambda(n)$ and $k = k(n)$ be integer functions, and let $g = \Big\{g_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}\Big\}_{a \in \{0,1\}^{\lambda(n)}}$ be a function family with $k(n)$-random-bits-unpredictability. Then the following holds for every*

*polynomial $t = t(n)$. Let $\alpha(n) = k(n)/\ell(n)$, and let $H_n \leftarrow \{0,1\}^{t(n)\times q(n)}$ be a random matrix, for $q = \lfloor \alpha t - \sqrt{\alpha t \cdot s \log n} - s \log n \rfloor$. Then for $X_n^1 \ldots, X_n^{t(n)} \leftarrow (\{0,1\}^{m(n)})^{t(n)}$ and $A_n \leftarrow \{0,1\}^{\lambda(n)}$, the distribution ensemble*

$$\left\{H_n, A_n, H_n(g_{A_n}(X_n^1)_1, \ldots, g_{A_n}(X_n^{t(n)})_1), \ldots, H_n(g_{A_n}(X_n^1)_{\ell(n)}, \ldots, g_{A_n}(X_n^{t(n)})_{\ell(n)})\right\}_{n\in\mathbb{N}}$$

*is pseudorandom.*

We prove Theorem 7.1 below, but first let us deduce our main theorem.

**Theorem 7.2** (PRG construction)**.** *For any function $s(n) = \omega(1)$, there exists a construction of a PRG from a one-way function, that uses $O(s(n) \cdot n^3/\log^2 n)$ non-adaptive calls to the one-way function and a seed of length $O(s^2(n) \cdot n^4/\log^2 n))$.*

*Proof of Theorem 7.2.* Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a one-way function, $g' = \left\{g'_a\colon \{0,1\}^{m'(n)} \to \{0,1\}^{\ell'(n)}\right\}_{a\in\{0,1\}^{\lambda(n)}}$ be the function family promised by Corollary 6.2, and let $\alpha = k'/\ell' \leq 1$.

Let $s$ be as in Theorem 7.2 (assume without loss of generality that $s(n) \leq \log n$), $t = 4\lceil \frac{\ell'^2 s \log n}{(k'-m')^2} \rceil = O(s\cdot n^2/\log n)$, $m = t\cdot m'$ and $\ell = (\lfloor \alpha t - \sqrt{\alpha t s \log n} - s\log n\rfloor)\ell'$. Let $\mathcal{H} = \{0,1\}^{t\times(\lfloor \alpha t - \sqrt{\alpha t s \log n} - s\log n\rfloor)}$ be the set of all matrices of size $t \times (\lfloor \alpha t - \sqrt{\alpha t s \log n} - s\log n\rfloor)$, and let $G\colon \mathcal{H}\times\{0,1\}^\lambda \times \{0,1\}^m \to \mathcal{H}\times\{0,1\}^\lambda\times\{0,1\}^\ell$ be the function defined by

$$G(H, A, W_1, \ldots, W_t) := H, A, H(g'^r_A(W_1)_1, \ldots g'^r_A(W_t)_1), \ldots, H(g'^r_A(W_1)_{\ell'}, \ldots, g'^r_A(W_t)_{\ell'}).$$

By Theorem 7.1, the output of $G$ is pseudorandom when $H \leftarrow \mathcal{H}$, and $W_1, \ldots, W_t \leftarrow (\{0,1\}^{m_2})^t$. We need to show that $G$ is expanding. To do so, it is enough to verify that $m < \ell$.

Indeed,

$$\begin{aligned}
\ell - m &= (\lfloor \alpha t - \sqrt{\alpha t s \log n} - s\log n\rfloor)\ell' - tm' \\
&> \alpha t \ell' - 2\ell'\sqrt{ts\log n} - tm' \\
&= tk' - 2\ell'\sqrt{ts\log n} - tm' \\
&= t(k'-m') - 2\ell'\sqrt{ts\log n} \\
&\geq 0,
\end{aligned}$$

where the last inequality holds since $m = m't$ and since $t(k'-m') \geq 2\ell'\sqrt{ts\log n}$ by our choice of $t$.

Moreover, $G$ uses $tr = O(s\cdot n^3/\log n)$ calls to $f$ and has seed length $\log|\mathcal{H}| + \lambda + t\cdot m_2 = \log|\mathcal{H}| + O(t^2 + n^2 + s\cdot n^4/\log^2 n) = O(s^2\cdot n^4/\log^2 n)$. □

## 7.1 Exponentially-hard OWFs

Before proving Theorem 7.1, we state and prove our results for exponentially-hard one-way functions. We start with a formal definition of the latter.

**Definition 7.3** (Exponentially hard one-way function)**.** *A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is called a $T = T(n)$-*hard one-way function *if for every n.u. algorithm $A$ of size at most $T(n)$, for all but finitely many $n \in \mathbb{N}$,*

$$\Pr_{x\leftarrow\{0,1\}^n}\left[A(f(x)) \in f^{-1}(f(x))\right] \leq 1/T(N).$$

*f is* n.u exponentially-hard one-way function *if it is* $2^{cn}$*-hard one-way function for some constant* $c > 0$.

We get the following theorem:

**Theorem 7.4** (PRG construction from exponentially-hard OWFs)**.** *For any function* $s(n) = \omega(1)$*, there exists a construction of a poly-time secure PRG from an exponentially-hard one-way function, that uses* $O(s(n) \cdot \log n)$ *non-adaptive calls to the one-way function.*

*Proof.* Let $f$ be an $2^{cn}$-hard one-way function. We use the well-known fact that we can extract $\delta n$ GL hard-core bits from the input of $f$ , for some constant $c > \delta > 0$. Thus, by the construction in Theorem 5.1, we get a function family $g$ with $(n + \epsilon n)$-bits-unpredictability, for some constant $\epsilon > 0$ (and $g$ only makes one call to $f$).

Next, by Theorem 6.1, and taking $r = \lceil 3/\epsilon \rceil + 1 = O(1)$, we get that the function family $g' = g^r$ has $k'(n) = (\lceil 3/\epsilon \rceil (1 + \epsilon)n)$-random-bits-unpredictability. Moreover, $g^r$ has input length $m'(n) = O(\log n) + n(\lceil 3/\epsilon \rceil + 1)$, output length $\ell'(n) = 2n(\lceil 3/\epsilon \rceil)$. We get that $k'(n) - m'(n) = \Omega(n) = \Omega(\ell'(n))$.

Let $\alpha = k'(n)/\ell'(n)$. Let $s$ be as in Theorem 7.4, $t = 4\lceil \frac{\ell'^2 s \log n}{(k'-m')^2} \rceil = O(s \log n)$, $m = t \cdot m'$ and $\ell = (\lfloor \alpha t - \sqrt{\alpha t s \log n} - s \log n \rfloor)\ell'$. Let $\mathcal{H}$ be the set of all matrices of size $t \times (\lfloor \alpha t - \sqrt{\alpha t s \log n} - s \log n \rfloor)$, and let $G \colon \mathcal{H} \times \{0,1\}^{\lambda} \times \{0,1\}^{m} \to \mathcal{H} \times \{0,1\}^{\lambda} \times \{0,1\}^{\ell}$ be the function defined by

$$G(H, A, W_1, \ldots, W_t) := H, A, H(g_A^r(W_1)_1, \ldots g_A^r(W_t)_1), \ldots, H(g_A^r(W_1)_{\ell'}, \ldots, g_A^r(W_t)_{\ell'}).$$

By Theorem 7.1, the output of $G$ is pseudorandom when $H \leftarrow \mathcal{H}$, and $W_1, \ldots, W_t \leftarrow (\{0,1\}^{m_2})^t$. By the same calculation as in the proof of Theorem 7.2, $G$ is expanding. Moreover, $G$ uses $tr = O(s \log n)$ calls to $f$ . □

## 7.2 Proving Theorem 7.1

By a simple hybrid argument, it is enough to prove the following claim.

**Claim 7.5.** *Let* $g, t, H_n, A_n$ *and* $X_n^1, \ldots, X_n^t$ *be as in Theorem 7.1. Then for every sequence* $\{i_n\}_{n \in \mathbb{N}}$*, and for every n.u.-poly-time algorithm E,*

$$\begin{aligned} &|\Pr\big[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, H_n(g_{A_n}(X_n^1)_{i_n}, \ldots, g_{A_n}(X_n^t)_{i_n})) = 1\big] \\ &\quad - \Pr\big[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, U_{q(n)}) = 1\big]| = neg(n). \end{aligned}$$

*Proof of Theorem 7.1.* Theorem 7.1 follows from Claim 7.5 by a simple hybrid argument. □

In the following we prove Claim 7.5. Fix $c \in \mathbb{N}$, a n.u.-poly-time $E$ and a constant $d$ such that $t(n) \leq n^d$ for large enough $n$. We want to show that

$$\begin{aligned} &|\Pr\big[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, H_n(g_{A_n}(X_n^1)_{i_n}, \ldots, g_{A_n}(X_n^t)_{i_n})) = 1\big] \\ &\quad - \Pr\big[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, U_{q(n)}) = 1\big]| < n^{-c}. \end{aligned} \tag{5}$$

for all but finitely many $n$'s. Let $c' = c + d + 2$. For every $n \in \mathbb{N}$ and $j \in [t(n)]$, let $\mathcal{S}_n^j = \mathcal{S}^g(X_n^j)$ be the set promised by the assumed $(k, n^{-c'})$-random-bits-unpredictability property of $g$. We define the random variables $Q^1, \ldots, Q^t$ as follows. For every $j \in t$, let $Q^j = g_A(X_n^j)_{i_n}$ if $i_n \notin \mathcal{S}_n^j$, or a

uniform bit otherwise. By the definition of bits-unpredictability, it holds that for every n.u.-poly-time algorithm $E'$,

$$\left|\Pr\left[E'(g_A(X_n^j)_{<i_n}, g_A(X_n^j)_{i_n}) = 1\right] - \Pr\left[E'(g_A(X_n^j)_{<i_n}, Q^j) = 1\right]\right| \leq n^{-c'}. \tag{6}$$

The proof of Claim 7.5 follows from the following two claims.

**Claim 7.6.** *For all but infinitely many $n$'s,*

$$\begin{aligned}&|\Pr\left[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, H_n(Q^1, \ldots, Q^t)) = 1\right]\\ &- \Pr\left[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, U_{q(n)}) = 1\right]| < n^{-c}/2\end{aligned}$$

**Claim 7.7.** *For all but infinitely many $n$'s,*

$$\begin{aligned}&|\Pr\left[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, H_n(g_{A_n}(X_n^1)_{i_n}, \ldots, g_{A_n}(X_n^t)_{i_n})) = 1\right]\\ &- \Pr\left[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, H_n(Q^1, \ldots, Q^t)) = 1\right]| < n^{-c}/2\end{aligned}$$

We will prove Claim 7.6 and Claim 7.7 below, but first let us prove Claim 7.5.

*Proof of Claim 7.5.* Equation (5) holds by Claim 7.6 and Claim 7.7 and the triangle inequality. The claim follows since Equation (5) holds for every $c \in \mathbb{N}$. □

### 7.3 Proving Claim 7.6

*Proof of Claim 7.6.* We will show that given $g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}$, the distribution of $(Q^1, \ldots, Q^t)$ is $n^{-c}/3$-close to a distribution with min-entropy at least $q(n) + \omega(\log n)$. The proof then follows by the leftover hash lemma.

To do so, we start by showing that with probability $1 - n^{-c}/3$, there are at least $q(n) + \omega(\log n)$ indexes $j$ such that $i_n \in \mathcal{S}^j$. To see the above, fix $n$ and omit it from the notation. Let $q' = q + s \log n$, and for every $j \in [t]$, let $\delta_j$ be an indicator for the event that $i \in \mathcal{S}_j$. By construction, $\delta_1, \ldots, \delta_t$ are independent random variables, and by the definition of $k$-random-bits-unpredictability, for each $j \in [t]$, it holds that $\Pr[\delta_j = 1] \geq k/\ell = \alpha$. Thus, by Chernoff inequality, for large enough $n$ it holds that

$$\Pr\left[\sum_{j=1}^{t} \delta_j < q'\right] = \Pr\left[\sum_{j=1}^{t} \delta_j < \alpha t - \sqrt{\alpha t s \log n}\right] \leq 2^{-s \log n/3} < n^{-c}/3,$$

as we wanted to show. Next, let $\mathcal{J} = \left\{j \colon i_n \in \mathcal{S}^j\right\}$ be the set of $j$'s for which $Q^j$ is uniform independent bit. By the above $\Pr[|\mathcal{J}| < q'] < n^{-c}/3$, and thus the distribution $(Q^1, \ldots, Q^t)$ is $n^{-c}/3$ close to the distribution $(Q^1, \ldots, Q^t)|_{|\mathcal{J}| \geq q'}$. To bound the min-entropy of the latter, we want to show that for every $q^1, \ldots, q^t$, it holds that $\Pr\left[Q^1, \ldots, Q^t = q^1, \ldots, q^t \mid |\mathcal{J}| \geq q'\right] \leq 2^{-q'}$, which concludes the proof. It holds that,

$$\begin{aligned}\Pr\left[Q^1, \ldots, Q^t = q^1, \ldots, q^t \mid |\mathcal{J}| \geq q'\right] &= \mathrm{E}_{J \leftarrow \mathcal{J}|_{|\mathcal{J}| > q'}}\left[\Pr\left[Q^1, \ldots, Q^t = q^1, \ldots, q^t \mid \mathcal{J} = J\right]\right]\\ &\leq \mathrm{E}_{J \leftarrow \mathcal{J}|_{|\mathcal{J}| > q'}}\left[2^{-|J|}\right]\\ &\leq 2^{-q'},\end{aligned}$$

where the first inequality holds since for every $j \in \mathcal{J}$, $Q^j$ is a uniform and independent random bit. □

### 7.4 Proving Claim 7.7

*Proof of Claim 7.7.* Assume towards a contradiction that the claim does not hold. That is,

$$\begin{aligned}&|\Pr\big[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, H_n(g_{A_n}(X_n^1)_{i_n}, \ldots, g_{A_n}(X_n^t)_{i_n})) = 1\big]\\ &- \Pr\big[E(H_n, A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, H_n(Q^1, \ldots, Q^t)) = 1\big]| \geq n^{-c}/2\end{aligned}$$

for some algorithm $E$ and for infinitely many $n$'s. By data-processing inequality, it holds that for some n.u.-poly-time $\widehat{E}$ and for infinitely many $n$'s,

$$\begin{aligned}&|\Pr\Big[\widehat{E}(A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, g_{A_n}(X_n^1)_{i_n}, \ldots, g_{A_n}(X_n^t)_{i_n}) = 1\Big]\\ &- \Pr\Big[\widehat{E}(A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, Q^1, \ldots, Q^t) = 1\Big]| \geq n^{-c}/2.\end{aligned}$$

Fix such $n$. By a simple hybrid argument, we get that there exists some $j^* \in [t]$, such that,

$$\begin{aligned}&|\Pr\Big[\widehat{E}(A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, g_{A_n}(X_n^1)_{i_n}, \ldots, g_{A_n}(X_n^{j^*})_{i_n}, Q^{j^*+1}, \ldots, Q^t) = 1\Big]\\ &- \Pr\Big[\widehat{E}(A_n, g_{A_n}(X_n^1)_{<i_n}, \ldots, g_{A_n}(X_n^t)_{<i_n}, g_{A_n}(X_n^1)_{i_n}, \ldots, g_{A_n}(X_n^{j^*-1})_{i_n}, Q^{j^*}, \ldots, Q^t) = 1\Big]| \geq n^{-c'}/2.\end{aligned}$$

By a simple averaging argument, there is a fixing $x^1, \ldots, x^{j^*-1}, x^{j+1}, \ldots, x^t$ for $X_n^1, \ldots, X_n^{j^*-1}, X_n^{j^*+1}, \ldots, X_n^t$, and $b^j$ for every $Q^j$ with $i_n \in \mathcal{S}^g(x^j)$, such that the following holds. Let $q^j(a) = g_a(x^j)$ if $i_n \notin \mathcal{S}^g(x^j)$, or $b^j$ otherwise. Then it holds that

$$\begin{aligned}&|\Pr[\widehat{E}(A_n, g_{A_n}(x^1)_{<i_n}, \ldots, g_{A_n}(X_n^{j^*}), \ldots, g_{A_n}(x^t)_{<i_n}, g_{A_n}(x^1)_{i_n}, \ldots, g_{A_n}(X_n^{j^*})_{i_n}, q^{j^*+1}(A_n), \ldots, q^t(A_n)) = 1]\\ &\quad - \Pr[\widehat{E}(A_n, g_{A_n}(x^1)_{<i_n}, \ldots, g_{A_n}(X_n^{j^*}), \ldots, g_{A_n}(x^t)_{<i_n}, g_{A_n}(x^1)_{i_n}, \ldots, Q^{j^*}, q^{j^*+1}(A_n), \ldots, q^t(A_n)) = 1]|\\ &\geq n^{-c'}/2.\end{aligned}$$

The above is a contradiction to the bit-unpredictability property of $g$. Indeed, Let

$$\begin{aligned}&E'(a, g_a(x)_{<i}, b)\\ &\quad = \widehat{E}(A_n, g_{A_n}(x^1)_{<i_n}, \ldots, g_a(x)_{<i}, \ldots, g_{A_n}(x^t)_{<i_n}, g_{A_n}(x^1)_{i_n}, \ldots, g_{A_n}(x^{j^*-1})_{i_n}, b, q^{j^*+1}(A_n), \ldots, q^t(A_n)).\end{aligned}$$

We get that

$$|\Pr\big[E'(A_n, g_{A_n}(X_n)_{<i_n}, g_{A_n}(X_n)_{i_n}) = 1\big] - \Pr\big[E'(A_n, g_{A_n}(X_n)_{<i_n}, Q) = 1\big]| \geq n^{-c'}/2.$$

where $Q$ is equal to $g_{A_n}(X_n)_{i_n}$) if $i_n \notin \mathcal{S}^g(X_n)$, or uniform bit otherwise. This is a contradiction to Equation (6). □

## 8 Saving Seed Length

In this section we show how to use the transformation from [VZ12] to get the following theorem.

**Theorem 8.1** (PRG construction)**.** *For any function $s = \omega(1)$, there exists a construction of a PRG from a one-way function, that uses $O(s(n) \cdot n^3/\log^2 n)$ calls to the one-way function and a seed of length $O(s(n) \cdot n^3/\log n)$.*

To get an improvement in the seed length, we will also need to use a hash function with a shorter description in the extraction step, described in Section 7. For this, we need to define 2-universal families.

**Definition 8.2** (2-universal family)**.** *A family of function* $\mathcal{F} = \left\{f\colon \{0,1\}^n \to \{0,1\}^\ell\right\}$ *is* 2-universal *if for every* $x \neq x' \in \{0,1\}^n$ *it holds that* $\Pr_{f\leftarrow\mathcal{F}}[f(x) = f(x')] = 2^{-\ell}$.

*A universal a family is* explicit *if given a description of a function* $f \in \mathcal{F}$ *and* $x \in \{0,1\}^n$*,* $f(x)$ *can be computed in polynomial time (in* $n, \ell$*).*

The family of all matrices of size $n \times m$ is an explicit 2-universal family, but it is well known that there are explicit 2-universal families with description size $O(n+m)$. An important property of 2-universal families is that they can be used to construct a strong extractor. This is stated in the leftover hash lemma:

**Lemma 8.3** (Leftover hash lemma, standard version, [ILL89])**.** *Let* $n \in \mathbb{N}$*,* $\varepsilon \in [0,1]$*, and let* $X$ *be a random variable over* $\{0,1\}^n$*. Let* $\mathcal{H} = \left\{h\colon \{0,1\}^n \to \{0,1\}^\ell\right\}$ *be a 2-universal hash family with* $\ell \leq \mathrm{H}_\infty(X) - 2\log 1/\varepsilon$*. Then,*

$$SD((H, H(X)), (H, U_\ell)) \leq \varepsilon$$

*for* $U_\ell$ *being the uniform distribution over* $\{0,1\}^\ell$ *and* $H$ *being the uniform distribution over* $\mathcal{H}$*.*

We are now ready to prove the main result of this section.

*Proof.* Observe that the significant parts of the seed of the PRG $G$ defined in the proof of Theorem 7.2 are the description of $\mathcal{H}$, and $t$ inputs to the function $g^r$.

**More efficient hash function.** We start with reducing the description length of $\mathcal{H}$ by using more efficient 2-universal family. Indeed, the proof of Claim 7.5 holds also when $H_n$ is a random function from a 2-universal family instead of a random matrix. We change the proof of Theorem 7.2, such that $\mathcal{H} = \left\{h\colon \{0,1\}^{t(n)} \to \{0,1\}^{\alpha t - \sqrt{\alpha t s \log n} - s\log n}\right\}$ is a 2-universal family of description size $\log|\mathcal{H}| = O(t)$.

**Using the transformation of [VZ12].** Next, we use the transformation of [VZ12] to avoid the need to get $t$ independent inputs for $g^r$ as input to the PRG. Let us first recall the construction given in Sections 5 to 7. The construction starts with a function family $g$ which has non-trivial bits-unpredictability. Then, for every $j \in [t]$ we compute

$$Y^j = g^r_A(I^j, X^1_j, \ldots, X^r_j) = g_A(X^{j,1})_{\geq I^j}, g_A(X^{j,2}), \ldots, g_A(X^{j,r})_{<I^j}.$$

Finally, we extract pseudorandom bits by applying an extractor on $Y^1_i, \ldots Y^t_i$ for every $i \in [(r-1)\ell]$. We prove that $H(Y^1_i, \ldots Y^t_i)$ is indistinguishable from uniform, given $A, Y^1_{i<}, \ldots Y^t_{<i}$. Moreover, by inspecting the reductions in the proofs of Theorems 6.1 and 7.1, it is not hard to see that $H(Y^1_i, \ldots Y^t_i)$ is indistinguishable from uniform, even given $I^1, \ldots, I^t$ (in addition to $A, Y^1_{i<}, \ldots Y^t_{<i}$).

Vadhan and Zheng [VZ12] observed that for computing $Y^j_i$, we only need to know the value of $A, I^j$ and exactly one (specific) of the values of $X^{j,1}, \ldots, X^{j,r}$. In particular, we don't need to

know the value of $X^{j,1},\ldots,X^{j,\alpha-1}$, where $\alpha$ is such that $i=\alpha\cdot\ell+\beta$ for $\beta\in[\ell]$, to compute $Y_i^j$. Thus, we can sample each input to $g$ only when it is used. This gives rise to an algorithm $G'$ that computes the output of the PRG in the following way: First, $G'$ samples $A$, and for each $j\in[t]$, the $G'$ samples $I^j$, and $X^{j,r},X^{j,r-1}$ uniformly at random. Then, for each $i$ from $(r-1)\ell$ to $(r-2)\ell+1$, the algorithm computes $H(Y_i^1,\ldots Y_i^t)$ (notice that the relevant bits have already been fixed by $A,I^j,X^{j,r}$ and $X^{j,r-1}$) and outputs the hashed value. The total length of the output of $G'$ so far is $q=\ell\cdot t(m/\ell+\Omega(\log n/\ell))=tm+\Omega(t\cdot\log n)$. After finishing, the algorithm samples $X^{j,r-2}$ uniformly at random for every $j$, and continues this process for another $\ell$ indexes ($i$ from $(r-2)\ell$ to $(r-3)\ell+1$), and so on. This process of sampling and hashing continues until it gets to $i=1$, where in the $k$-th iteration, $G'$ samples $X^{j,r-k}$ for each $j$, and the hashes $H(Y_i^1,\ldots Y_i^t)$ for each $i$ between $(r-k)\ell$ to $(r-k-1)\ell+1$. This results with $tm+\Omega(t\cdot\log n)$ pseudorandom bits in every iteration.

Clearly, the output of the described $G'$ is equal to the output of the PRG. Moreover, the output in the $k$-th iteration is indistinguishable from uniform, even given the parts of $Y^j_{<(r-k-1)\ell}$ that have already been sampled up to the $k$-th iteration (that is, $A,I^j$ and

$$Y^j[k]:=Y^j_{(r-k-1)\ell-(I^j-1)},\ldots,Y^j_{(r-k-1)\ell-1}=g_A(X^{j,r-k})_{<I^j}).$$

More formally, for every $k\in[r-1]$, let $Z^k$ be the output of $G'$ in the $k$-th iteration. It follows from the proof of Theorem 7.1 that for every such $k$,

$$(A,I^1,\ldots,I^t,Y^1[k],\ldots,Y^t[k],Z^k)\overset{c}{\approx}(A,I^1,\ldots,I^t,Y^1[k],\ldots,Y^t[k],U_q). \tag{7}$$

Using an hybrid argument we can also see that

$$\begin{aligned}&(A,I^1,\ldots,I^t,Y^1[k],\ldots,Y^t[k],Z^1,\ldots,Z^k)\\&\overset{c}{\approx}(A,I^1,\ldots,I^t,Y^1[k],\ldots,Y^t[k],U_{k\cdot q}).\end{aligned} \tag{8}$$

The idea in [VZ12] is to output only $\Omega(t\cdot\log n)$ bits of the above algorithm in each iteration $k$, and to use the other $tm$ pseudorandom bits to sample the inputs $X^{1,r-k-1},\ldots X^{t,r-k-1}$ of $g$ for the next iteration. Since the output of $G'$ in each iteration is indistinguishable from uniform, the output of this process is pseudorandom by a simple hybrid argument.

Indeed, fix a distinguisher $E$, a constant $c\in\mathbb{N}$ and a large enough $n\in N$, and for each $\tau\in[r-1]$ let $G'(\tau)$ be the algorithm that samples $X^{1,r-k-1},\ldots X^{t,r-k-1}$ uniformly at random in the beginning of each iteration $k\leq\tau$, and uses the first $tm$ bits of the output of each iteration $k>\tau$ as $X^{1,r-k-1},\ldots X^{t,r-k-1}$. That is, $G'(r-1)$ is simply the algorithm $G'$ described above, while $G'(1)$ is the algorithm considered by [VZ12], that only uses randomness to sample $X^{1,r-1},X^{1,r},\ldots,X^{t,r-1},X^{t,r}$. Let $Z^1(\tau),\ldots,Z^{r-1}(\tau)$ be the output of $G'(\tau)$ in each iteration respectively, and let $Z^k(\tau)_{>tm}$ be the last $w-tm$ bits of $Z^k(\tau)$. Since the output of $G'$ is pseudorandom, we get that,

$$|\Pr\big[E(Z^1(r-1)_{>tm},\ldots,Z^{r-1}(r-1)_{>tm})=1\big]-\Pr\big[E(U_{(r-1)\cdot(q-tm)})=1\big]|<n^{-c}.$$

We want to show that it also holds that

$$\big|\Pr\big[E(Z^1(1)_{>tm},\ldots,Z^{r-1}(1)_{>tm})=1\big]-\Pr\big[E(U_{(r-1)\cdot(q-tm)})=1\big]\big|<2n^{-c},$$

and thus it is enough to show that

$$|\Pr\big[E(Z^1(r-1)_{>tm},\ldots,Z^{r-1}(r-1)_{>tm})=1\big]-\Pr\big[E(Z^1(1)_{>tm},\ldots,Z^{r-1}(1)_{>tm})=1\big]|<n^{-c}.$$

Assume towards a contradiction that the above does not hold. By an hybrid argument, there exists some $\tau\in[r-1]$ such that $E$ distinguish between $(Z^1(\tau)_{>tm},\ldots,Z^{r-1}(\tau)_{>tm})$ and $(Z^1(\tau+1)_{>tm},\ldots,Z^{r-1}(\tau+1)_{>tm})$ with advantage $n^{-c}/r$.

Observing that $(Z^{\tau+1}(\tau)_{>tm},\ldots,Z^{r-1}(\tau)_{>tm})$ can be computed from $Z^\tau(\tau)$ and $A,I^1,\ldots,I^t,Y^1[\tau],\ldots,Y^t[\tau]$, while $(Z^{\tau+1}(\tau+1)_{>tm},\ldots,Z^{r-1}(\tau+1)_{>tm})$ can be computed by the same function from $U_q$ and $A,I^1,\ldots,I^t,Y^1[\tau],\ldots,Y^t[\tau]$, we get the following by data processing. $E$ distinguishes between

$$A,I^1,\ldots,I^t,Y^1[\tau],\ldots,Y^t[\tau],Z^1(\tau)_{>tm},\ldots,Z^\tau(\tau)_{>tm},Z^\tau(\tau)_{\leq tm}$$

and

$$A,I^1,\ldots,I^t,Y^1[\tau],\ldots,Y^t[\tau],Z^1(\tau+1)_{>tm},\ldots,Z^\tau(\tau+1)_{>tm},U_{tm}$$

with the same advantage, $n^{-c}/r$. Since by definition $(Z^1(\tau),\ldots,Z^\tau(\tau))\equiv(Z^1(\tau+1),\ldots,Z^\tau(\tau+1))\equiv(Z^1,\ldots,Z^\tau)$, we get a contradiction to Equation (8).

To see that $G'(1)$ outputs more pseudorandom bits than the randomness used, observe that $G'(1)$ uses $2tm$ random bits to sample $X^{1,r-1},X^{1,r},\ldots,X^{t,r-1},X^{t,r}$, and outputs $\Omega(t\cdot\log n)$ pseudorandom bits in each iteration. Thus, for for $r=\Omega(m/\log n)$, $G'(1)$ an expanding function. □

## References


[BM82] Manuel Blum and Silvio Micali. "How to Generate Cryptographically Strong Sequences of Pseudo Random Bits". In: *Annual Symposium on Foundations of Computer Science (FOCS)*. 1982, pp. 112–117 (cit. on pp. 1, 3, 4).

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "On the Cryptographic Applications of Random Functions". In: *Advances in Cryptology: Proceedings of CRYPTO 84*. 1984, pp. 276–288 (cit. on p. 2).

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for all One-Way Functions". In: *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*. 1989, pp. 25–32 (cit. on pp. 2, 10, 25).

[GM84] Shafi Goldwasser and Silvio Micali. "Probabilistic Encryption". In: *Journal of Computer and System Sciences* (1984), pp. 270–299 (cit. on p. 2).

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. "How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority". In: *stoc19*. 1987, pp. 218–229 (cit. on p. 2).

[HHR06] Iftach Haitner, Danny Harnik, and Omer Reingold. "On the Power of the Randomized Iterate". In: *Advances in Cryptology – CRYPTO '06*. 2006, pp. 22–40 (cit. on pp. 2, 3).

[HILL99] Johan Hastad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "A pseudorandom generator from any one-way function". In: *SIAM Journal on Computing* (1999), pp. 1364–1396 (cit. on pp. 2–4, 8).

[Hol06a] Thomas Holenstein. "Pseudorandom Generators from One-Way Functions: A Simple Construction for Any Hardness". In: *Theory of Cryptography (TCC)*. 2006 (cit. on p. 2).

[Hol06b] Thomas Holenstein. "Strengthening key agreement using hard-core sets". PhD thesis. ETH Zurich, 2006 (cit. on pp. 2, 7, 25).

[HRV13] Iftach Haitner, Omer Reingold, and Salil Vadhan. "Efficiency Improvements in Constructing Pseudorandom Generators from One-Way Functions". In: *SIAM Journal on Computing* 42.3 (2013), pp. 1405–1430 (cit. on pp. 2, 3, 6, 25, 28).

[HRVW09] Iftach Haitner, Omer Reingold, Salil Vadhan, and Hoeteck Wee. "Inaccessible Entropy". In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*. 2009, pp. 611–620 (cit. on p. 6).

[HV17] Iftach Haitner and Salil Vadhan. "The many entropies in one-way functions". In: *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*. Springer, 2017, pp. 159–217 (cit. on pp. 2, 5, 28).

[ILL89] Russell Impagliazzo, Leonid A Levin, and Michael Luby. "Pseudo-random generation from one-way functions". In: *Annual ACM Symposium on Theory of Computing (STOC)*. 1989, pp. 12–24 (cit. on p. 21).

[Nao91] Moni Naor. "Bit Commitment Using Pseudorandomness". In: *Journal of Cryptology* (1991), pp. 151–158 (cit. on p. 2).

[Sha83] Adi Shamir. "On the generation of cryptographically strong pseudorandom sequences". In: *ACM Transactions on Computer Systems (TOCS)* 1.1 (1983), pp. 38–44 (cit. on p. 3).

[VZ12] Salil Vadhan and Colin Jia Zheng. "Characterizing pseudoentropy and simplifying pseudorandom generator constructions". In: *Annual ACM Symposium on Theory of Computing (STOC)*. 2012, pp. 817–836 (cit. on pp. 2, 3, 5, 8, 20–22).

[Yao82] Andrew C. Yao. "Theory and Applications of Trapdoor Functions". In: *Annual Symposium on Foundations of Computer Science (FOCS)*. 1982, pp. 80–91 (cit. on p. 10).

[YLW15] Yu Yu, Xiangxue Li, and Jian Weng. "Pseudorandom generators from regular one-way functions: New constructions with improved parameters". In: *Theoretical Computer Science* 569 (2015), pp. 58–69 (cit. on p. 25).

# A A PRG from Uniform One-Way Functions

In this part we explain how to modify our construction and proofs to hold when the one-way function is only secure against uniform adversaries. We derive the following theorem:

**Theorem A.1** (PRG construction, uniform setting)**.** *For any function $s(n) = \omega(1)$, there exists a construction of a PRG from a one-way function, that uses $O(s(n) \cdot n^3/\log^2 n)$ non-adaptive calls to the one-way function and a seed of length $O(s^2(n) \cdot n^4/\log n))$.*

Note that the seed length is worse than the seed length in the non-adaptive, non-uniform construction (Theorem 7.2). We can improve the seed length to be the same as in the non-uniform setting by using a more efficient hash function in the basic construction of bits unpredictability,

instead of a random matrix. For example, a $\log n$ factor can be saved using Toeplitz matrices (see for example [GL89; YLW15]).

In the uniform setting we cannot use non-uniformity in the security reductions. However, the only crucial use of non-uniformity in the security reduction of the PRG construction is in the last step, of pseudorandomness extraction from random-bits unpredictability. In the proof of Claim 7.7, we use a hybrid argument, for which the adversary needs to be able to identify which bits are unpredictable (namely, whether the index $i_n$ is in the set $\mathcal{S}(X_j)$) and which are not. The following definition allows the proof to go through also in the uniform setting. There are two main differences between the following definition to Definition 4.2. First, in Definition A.2 we allow the set $\mathcal{S}$ of unpredictable bits to be dependent on the distinguisher $E$ and on the key $a$ of the function. Second, the uniform distinguisher $E$ has oracle access to the characteristic function of $\mathcal{S}$.

**Definition A.2** (Random bits unpredictability, uniform setting)**.** *Let $m = m(n)$, $\ell = \ell(n)$ and $k = k(n)$ be efficiently computable integer functions. We say that a function family $g = \left\{g_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}\right\}_{a\in\{0,1\}^{\lambda(n)}}$ has $k$-*random-bits-unpredictability *if the following holds for every oracle-aidded* PPT *algorithm $E$ and every constant $c$. For every $n \in \mathbb{N}$, $x \in \{0,1\}^{m(n)}$ and $a \in \{0,1\}^{\lambda(n)}$, there exists a set $\mathcal{S}(x,a) \subseteq [\ell(n)]$, such that the following holds for $X_n \leftarrow \{0,1\}^{m(n)}$.*

1. *For every $i \in [\ell(n)]$, $\Pr[i \in \mathcal{S}(X_n, A_n)] \geq k(n)/\ell(n)$.*

2. *for every sequence $\{i_n\}_{n\in\mathbb{N}}$,*

$$\begin{aligned}|\Pr[E^{\chi_{\mathcal{S}}}(A_n, g_A(X_n)_{<i_n}, g_A(X_n)_{i_n}) = 1 \mid i_n \in \mathcal{S}(X_n, A_n)] \\ - \Pr[E^{\chi_{\mathcal{S}}}(A_n, g(X_n)_{<i_n}, U) = 1 \mid i_n \in \mathcal{S}(X_n, A_n)]| \leq n^{-c}.\end{aligned}$$

*where $\chi_{\mathcal{S}}(a,x,i)$ is the characteristic function of $\mathcal{S}(x,a)$ (namely, $\chi_{\mathcal{S}}(x,a,i) = 1$ iff $i \in \mathcal{S}(x,a)$), provided that all the queries of $E$ to $\chi_L$ are computed independently of the input $A_n, g_A(X_n)_{<i}$.*

We next explain how to construct a function with uniform random-bits unpredictability from a one-way functions. In Appendix A.2 we show how to use Definition A.2 to get Theorem A.1.

## A.1 OWF to Random Bits Unpredictability

The proof is using Holenstein's hard-core lemma [Hol06b]. The following version of the lemma is from [HRV13].

**Lemma A.3** (Hard-core lemma)**.** *Let $n$ be a security parameter. Let $h \colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ and $V \colon \{0,1\}^n \to \{0,1\}$ be poly-time computable functions. Let $\delta_0 = \delta_0(n) \in [0,1]$, $\delta = \delta(n) \in [\delta_0, 1]$ and $\gamma = \gamma(n) \geq 2^{-n/3}$. Assume that*

$$\Pr_{X\leftarrow\{0,1\}^n}[E(\delta_0, \gamma, h(X)) = V(X)] \leq 1 - \delta/2$$

*for every probabilistic Turing machine $E$ running in time $T = T(n)$ and large enough $n$. Then for every oracle-aided predictor $P$ running in time $T \cdot (\gamma\delta_0/n)^{O(1)}$ and all sufficiently large $n$, there exists a set $L \subseteq \{0,1\}^n$ of density at least $\delta$ such that*

$$\Pr_{X\leftarrow L}[P^{\chi_L}(h(X)) = V(X)] \leq 1/2 + \gamma$$

*where $\chi_L$ is the characteristic function of $L$, provided that all the queries of $P$ to $\chi_L$ are computed independently of the input $h(X)$.*

We focus on the setting on polynomial security in which case the hard-core lemma directly yields the following corollary (setting $T(n) = \text{poly}(n)$, $\delta_0 = 1/\text{poly}(n)$ and $\gamma = n^{-c}$).

**Corollary A.4.** *Let $n$ be a security parameter. Let $h\colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ and $V\colon \{0,1\}^n \to \{0,1\}$ be poly-time computable functions. Let $p = p(n) \in \text{poly}$ and $\delta = \delta(n) \in [1/p(n), 1]$. Assume that for every* PPT *$E$,*

$$\Pr_{X \leftarrow \{0,1\}^n}[E(p(n), h(X)) = V(X)] \leq 1 - \delta/2$$

*for all but finitely many $n$. Then for every* PPT *oracle-aided predictor $P$, constant $c \in \mathbb{N}$ and all sufficiently large $n$, there exists a set $L \subseteq \{0,1\}^n$ of density at least $\delta$ such that*

$$\Pr_{X \leftarrow L}[P^{\chi_L}(h(X)) = V(X)] \leq 1/2 + n^{-c}$$

*where $\chi_L$ is the characteristic function of $L$, provided that all the queries of $P$ to $\chi_L$ are computed independently of the input $h(X)$.*

The result for constructing random-bits unpredictability is stated in the following lemma.

**Lemma A.5** (OWF to random-bits unpredictability, uniform setting)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a one-way function. Then there exists an efficiently computable function family*

$$g' = \left\{ g'_a \colon \{0,1\}^{m'(n)} \to \{0,1\}^{\ell'(n)} \right\}_{a \in \{0,1\}^{\lambda(n)}}$$

*with $k'$-uniform-random-bits unpredictability, for $m'(n) = O(n^2/\log n)$, $\ell'(n) = O(n^2/\log n)$, $\lambda(n) = n^2$ and $k'(n) \geq m'(n) + n$.*

*Moreover, the construction uses $O(n/\log n)$ non-adaptive calls to $f$.*

*Proof sketch.* Let $g$ be the function family defined in Theorem 5.1, and for $r(n) = 2n/\log n + 3$, let $g' = g^r$, as defined in Theorem 6.1.

Fix a PPT algorithm $E$. We start with bounding the probability

$$\Pr_{A \leftarrow \{0,1\}^{\lambda(n)}, X \leftarrow \{0,1\}^n, I \leftarrow [\ell(n)]}[E(A, g_A(X)_{<I}) = g_A(X)_I].$$

Let $c \in \mathbb{N}$ be a large enough constant, $k = n + \log n$, and let $\mathcal{S}$ be as defined in the proof of Theorem 5.1. By a similar proof to the proof of Theorem 5.1, it holds that for every large enough $n$,

$$\Pr_{A \leftarrow \{0,1\}^{\lambda(n)}, X \leftarrow \{0,1\}^n, I \leftarrow [\ell(n)]}[E(A, g_A(X)_{<I}) = g_A(X)_I \mid I \in \mathcal{S}(X)] \leq 1/2 + 1/n^c.$$

Thus,

$$\begin{aligned}
&\Pr_{A \leftarrow \{0,1\}^{\lambda(n)}, X \leftarrow \{0,1\}^n, I \leftarrow [\ell(n)]}[E(A, g_A(X)_{<I}) = g_A(X)_I] \\
&\leq \Pr[I \notin \mathcal{S}(X)] + \Pr[I \in \mathcal{S}(X)] \cdot \Pr[E(A, g_A(X)_{<I}) = g_A(X)_I \mid I \in \mathcal{S}(X)] \\
&\leq \Pr[I \notin \mathcal{S}(X)] + \Pr[I \in \mathcal{S}(X)] \cdot (1/2 + 1/n^c) \\
&\leq 1 - k/2\ell + 1/n^c
\end{aligned}$$

Let $h(A,X,I) = A, g_A(X)_{<I}$, and $V(A,X,I) = g_A(X)_I$. By the hard-core lemma (Corollary A.4), for every oracle-aided PPT $P$ and all sufficiently large $n$, there exists a set $\mathcal{L}$ of density at least $(k/\ell - 2/n^c)$ such that

$$\Pr_{(A,X,I)\leftarrow\mathcal{L}}[P^{\chi_L}(h(A,X,I)) = V(A,X,I)] \leq 1/2 + n^{-2c}. \tag{9}$$

By the standard distinguishing to prediction reduction, we get that for every oracle-aided PPT $P$ and all sufficiently large $n$, there exists a set $\mathcal{L}$ of density at least $(k/\ell - 2/n^c)$ such that

$$\begin{aligned} |\Pr_{A\leftarrow\{0,1\}^{\lambda(n)},X\leftarrow\{0,1\}^n,I\leftarrow[\ell(n)]}[D^{\chi_{\mathcal{L}}}(A, g_A(X)_{<I}, g_A(X)_I) = 1 \mid (A,X,I)\in\mathcal{L}] \\ - \Pr_{A\leftarrow\{0,1\}^{\lambda(n)},X\leftarrow\{0,1\}^n,I\leftarrow[\ell(n)]}[D^{\chi_{\mathcal{L}}}(A, g(X)_{<I}, U) = 1 \mid (A,X,I)\in\mathcal{L}]| \leq 2n^{-2c} \end{aligned} \tag{10}$$

provided that all the queries of $D$ to $\chi_L$ are computed independently of the input. Let $\mathcal{S}_D(x,a) = \{i\colon (a,x,i)\in\mathcal{L}\}$. We get that $\mathrm{E}[|\mathcal{S}_D(X_n,A_n)|] = k - 2\ell/n^c \geq k-1$. Equation (10) and a similar reduction to the one in the proof of Theorem 6.1 (bits unpredictability to random bits unpredictability) concludes the proof.

In more detail, assume toward a contradiction that $g'$ does not have $k'$ random bits unpredictability. By definition, there exists $c\in\mathbb{N}$ and oracle-aided PPT $E$ that breaks the random bits unpredictability for infinitely many $n$'s, and for every sets $\mathcal{S}_{g'}(i,x_1,\ldots,x_r,a)$ that satisfy the first condition in Definition A.2. We use $E$ to construct a distinguisher $D$ that attempts to contradict Equation (10). That is, let $D$ be the oracle-aided algorithm that, given an oracle to a characteristic function $\chi_{\mathcal{S}}$ for some sets $\{\mathcal{S}(x,a)\}_{x,a}$, and input $(a, g_a(x)_{<i}, b)$, do the following:

- Use $\chi_{\mathcal{S}}$ to simulate an oracle $\chi_{\mathcal{S}^r}$, where $\mathcal{S}^r(i,x^1,\ldots,x^r,a)$ are the sets defined in Theorem 6.1, namely,

$$\mathcal{S}^r(i,x^1,\ldots,x^r,a) = (\bigcup_{j\in[r]}\{z+(j-1)n-(i-1)\colon z\in\mathcal{S}(a,x^j)\})\bigcap[\ell(n)\cdot(r(n)-1)]. \tag{11}$$

- Sample random inputs and use $\chi_{\mathcal{S}^r}$ to find an index $z\in[\ell'(n)]$ such that $E$ breaks the random bits unpredictability (with respect to the sets $\mathcal{S}^r$) with advantage $n^{-c}/2$ on index $z$, or abort if no such exists.

- Use a similar reduction to the one in the proof of Theorem 6.1 to distinguish $(A, g_A(X)_{<I}, g_A(X)_I)$ from $(A, g_A(X)_{<I}, U)$, and apply the distinguisher on the input. Namely, let $i^* = ((i-1) - (z-1)) \bmod \ell(n) + 1$ and $s = \lceil\frac{z+(i^*-1)}{\ell}\rceil$ be such that for $g'_A(i^*,X^1,\ldots,X^r)$, the index $z$ belongs to the $s$-th block, and $i$ be the index of $z$ inside the block. That is, $s$ and $i^*$ are such that $g^r_A(i^*,X^1,\ldots,X^r)_{\leq z} = g_A(X^1)_{\geq i^*}, g_A(X^2),\ldots,g_A(X^s)_{\leq i}$. The algorithm $D$ sample $X^1,\ldots,X^{s-1}$ uniformly at random and executes

$$E(a, g_a(X^1)_{>i^*}, g_a(X^2),\ldots,g_a(x)_{<i}, b).$$

By Equation (10), $D$ fails to distinguish $(A, g_A(X)_{<I}, g_A(X)_I)|_{I\in\mathcal{S}_D}$ from $(A, g_A(X)_{<I}, U)|_{I\in\mathcal{S}_D}$ for some set $\mathcal{S}_D$ with $\mathrm{E}[|\mathcal{S}_D(X_n,A_n)|]\geq k-1$. By the reduction in the proof of Theorem 6.1, $E$ must fail to break the random bits unpredictability of $g'$ on $(\mathcal{S}_D)^r$, as defined in Equation (11). Moreover, by the proof of Theorem 6.1, $(\mathcal{S}_D)^r$ satisfies the first condition in Definition A.2. □

### A.2 Random Bits Unpredictability to PRG

We now show how to adjust our construction and the proof of Theorem 7.2 to get Theorem A.1.

*Proof sketch of Theorem A.1.* Let $g$ be the function family promised by Lemma A.5. Using Definition A.2, we adjust the construction and proof of Theorem 6.1 as follow:

1. In the construction, in each copy of $g_A(X^j)$, we will use a different matrix $A^j$. This adds $t \cdot n^2 = \omega(n^4/\log n)$ bits to the seed.
2. In the proof of Claim 7.7 (the hybrid argument), instead of defining the function $q^j(a)$ and fixing the inputs $x^j$, we sample $g_{A^j}(X^j)_{\leq i_n}$, and use the oracle $\chi$ to sample $Q^j$ from the right distribution. Given an input $g_A(X)_{<i}, B$ to the reduction such that $i \in \mathcal{S}(X, A))$, we replace a random row $j$ that satisfy $i \in \mathcal{S}(X^j, A^j)$ with the input.

□

## B Next-Block Pseudoentropy

For completeness, we give here the definitions of Next-Block Pseudoentropy used by Haitner, Reingold, and Vadhan [HRV13] and Haitner and Vadhan [HV17].

### B.1 The non-uniform setting

We start with the definition from [HV17] for the non-uniform setting.

**Definition B.1** (Next-Block pseudoentropy, non-uniform setting)**.** *Let $m = m(n)$ be an integer function. A distribution ensemble $X = \left\{X_n = (X_n^1, \ldots, X_n^m)\right\}_{n\in\mathbb{N}}$ is said to have (non-uniform) next-block (Shannon) pseudoentropy (at least) $k = k(n)$ if there exists a (jointly distributed) random variable $Y = \left\{Y_n = (Y_n^1, \ldots, Y_n^m)\right\}_{n\in\mathbb{N}}$ such that, for all but finitely many $n \in \mathbb{N}$:*

1. *$\sum_{i=1}^m H(Y_n^i \mid X_n^{<i}) \geq k(n)$, and*
2. *For every $i = i(n) \in [m(n)]$,*

$$\left\{X_n^{<i(n)}, X_n^{i(n)}\right\}_{n\in\mathbb{N}} \approx \left\{X_n^{<i(n)}, Y_n^{i(n)}\right\}_{n\in\mathbb{N}}$$

*A function $g$ has next-block pseudoentropy if $\{g(U_n)\}_{n\in\mathbb{N}}$ has.*

### B.2 The unifrom setting

Next, we present the definition from [HRV13] for the uniform setting.

**Definition B.2** (Next-Block pseudoentropy, uniform setting)**.** *Let $m = m(n)$ be an integer function. A distribution ensemble $X = \left\{X_n = (X_n^1, \ldots, X_n^m)\right\}_{n\in\mathbb{N}}$ is said to have (uniform) next-block (Shannon) pseudoentropy (at least) $k = k(n)$, if for every constant $c \in \mathbb{N}$ and every oracle-aided algorithm $E$ that runs in time at most $n^c$, there exists a (jointly distributed) random variable $Y = \left\{Y_n = (Y_n^1, \ldots, Y_n^m)\right\}_{n\in\mathbb{N}}$ such that, for all but finitely many $n \in \mathbb{N}$:*

1. *$\sum_{i=1}^m H(Y_n^i \mid X_n^{<i}) \geq k(n)$, and*

*2. For every $i = i(n) \in [m(n)]$,*

$$|\Pr\Big[D^{O_{X,Y}}(X^n_{<i(n)}, X^n_{i(n)}) = 1\Big] - \Pr\Big[D^{O_{X,Y}}(X^n_{<i(n)}, Y^n_{i(n)}) = 1\Big]| \leq n^{-c} \cdot L(n)$$

*where $O_{X,Y}(i)$ for $i \in [m(n)]$ samples according to the joint distribution $(X_n, Y^i_n)$, and $L = L(n)$ is a bound on number of calls made by $D$ to $O_{X,Y}$ (including the challenge itself).*

*A function $g$ has next-block pseudoentropy if $\{g(U_n)\}_{n\in\mathbb{N}}$ has.*



ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il