How to Use Nondeterminism in Cryptography

Marshall Ball
New York University, NY, USA

Peter Crawford-Kahrl
New York University, NY, USA

Abstract

Nondeterministic reductions have yielded powerful results in the theory of computational complexity, yet are effectively useless in a cryptographic context. The reason for this is simple, a nondeterministic polynomial time adversary can trivially break almost any cryptographic primitive by simply guessing the "key."

In order to use this powerful nondeterministic tool kit in the cryptographic context, we initiate the study of cryptography against adversaries with *limited nondeterminism*: polynomial time nondeterministic algorithms that are restricted to just a few bits of nondeterminism. We demonstrate that limited nondeterministic security is sufficient to prove two foundational results that have eluded our grasp for decades: dream hardness amplification, and extracting $\omega(\log n)$ hardcore bits.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography; Security and privacy → Cryptography

Keywords and phrases limited nondeterminism, cryptography, computational complexity, hardness amplification, pseudorandom generators, hardcore bits

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.15

Funding *Marshall Ball*: This work was supported in part by the National Science Foundation CAREER Award #2443735. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

Acknowledgements The first author would like to thank Benny Applebaum, Iftach Haitner, and Ronen Shaltiel for enlightening discussions on this topic.

1 Introduction

Nondeterministic reductions are an incredibly powerful tool in computational complexity. Nondeterministic reductions have enabled us to isolate incredibly hard problems (e.g., proving circuit lower bounds [22, 30]) and make already hard problems extremely hard (e.g., constructing problems that are incredibly hard even on a random instance [11, 18, 27]). Despite the powerful applications in computational hardness, nondeterministic reductions have had relatively little[1] impact on traditional cryptography.

It is not difficult to see why. Nearly any cryptographic scheme is easily broken by a nondeterministic adversary that can "guess" a secret key. For example, consider the (minimal) case of a one-way function: a function f that is easy to evaluate, yet given $y = f(x)$ for a random x it is hard to find any preimage of y. If hard means hard for nondeterministic polynomial time, putting aside exactly what this means for now, then this notion is clearly impossible to achieve: a nondeterministic adversary can simply nondeterministically "guess" a preimage $x \in \{0,1\}^n$ and check that $f(x) = y$.

[1] One notable exception is the work of Barak, Ong, and Vadhan [3] which uses nondeterministic hardness assumptions to derandomize cryptographic proof systems and commitment schemes.

Consequently, if one wants to construct a cryptographic primitive B from underlying cryptographic primitive A, one cannot use a nondeterministic reduction to prove security (at least in any standard sense).

In this work, we propose a way around this quagmire, enabling the use of nondeterministic reductions in a cryptographic context: *limited nondeterminism*. Intuitively, in our example above if the adversary only has access to $n/2$ nondeterministic bits, then the adversary cannot guess a complete witness and the naive attack above will no longer succeed.

Limited nondeterministic adversaries

Before going further, it is worth discussing what limited nondeterministic means in more detail. We consider a *ϕ-limited nondeterministic adversary* to be a (probabilistic) polynomial time algorithm that has access to ϕ-nondeterministic bits. If we restrict ourselves to non-probabilistic adversaries this definition is straightforward for decision problems. For probabilistic adversaries, we consider a notion mirroring AM: the short witness is chosen *after* the random coins. For search problems, as we are concerned with in this work, there are a variety of interpretations of "nondeterministic algorithm that outputs multiple bits." For the purpose of this work, we adopt a broad and simple approach: we consider a limited nondeterministic algorithm to successfully solve a problem (after sampling the random coins) if *any* computational path is successful.

Limited nondeterminism has been extensively studied in the context of computational complexity. Beginning in the 1980s, Kintala and Fischer [23] initiated the study of $\mathsf{NP}[\log^k(n)]$, what can be computed with $\log^k(n)$ nondeterministic bits. Papadimitriou and Yannakakis [26] used this notion to analyze various optimization problems. Cai and Chen [7] (see also [8]) and later Flum et al. [13] rigorously connected the notion to parametrized complexity. We refer the reader to the brief survey by Goldsmith et al. [15], although much has happened since it was written. One notable example is Ryan Williams's framework for proving new circuit lower bounds [29].

Despite its utility in complexity theory, the notion of limited nondeterminism has, to our knowledge, received no attention in the cryptographic context.

Feasibility of limited nondeterministic security

Armed with this strong adversarial model, the first and perhaps most important question is whether *cryptographic* security against such adversaries is indeed feasible. To this end, note that any $\phi(n)$-nondeterministic adversary can be simulated in $2^{\phi(n)} \cdot \mathrm{poly}(n)$ time. So, for example, subexponential security immediately implies security against n^ϵ-limited nondeterministic adversaries (for any constant $\epsilon < 1$).

However, we believe security ϕ-nondeterministic adversaries to be a strictly weaker notion than $2^\phi \,\mathrm{poly}(n)$-security. While we cannot definitively show this, there is some readily available evidence. In the nonuniform setting a counting argument demonstrates a massive gap in the size of the two classes: while there are $2^{\tilde{O}(n^c)}$ n^ϵ-limited nondeterministic adversaries of size n^c, there are $2^{2^{n^\epsilon}}$ adversaries of size 2^{n^ϵ}. Moreover, the two notions can be readily separated in a relativized world.[2]

[2] Consider a random oracle and a "secret shared" inversion oracle. The secret shared inversion oracle allows one to find a preimage of the random oracle with a superpolynomial number of queries, but few queries reveal nothing.

Finally, we note that ϕ-limited nondeterministic security may be possible in regimes where 2^{ϕ}-security is not. Namely, building on classical time/space tradeoffs [12, 19], it was recently shown that a one-way function cannot be secure against nonuniform adversaries of size $2^{.81n}$ [21, 25]. However, these methods seem to critically rely on large amounts of nonuniform advice and thus it is reasonable to conjecture that one-way functions secure against nonuniform polynomial size adversaries with even $0.99n$-bits of nondeterminism exist.

One might hope a closer analogue of security against limited nondeterministic adversaries can be found in leakage-resilient cryptography. One can view the nondeterministic bits as a small amount of (computationally unbounded) leakage on a cryptographic secret. While this notion of leakage-resilient cryptography has had extensive coverage in information-theoretic settings, it has received scant attention in computational regimes (given the strong leakage model). Aggarwal and Maurer [1] studied this notion leakage-resilience (providing various equivalent models) in the context of *worst-case* complexity of computational problems. However, to our knowledge, we are the first to study it in average-case settings, let alone in the cryptographic context.

1.1 Our Results: How to Use Limited Nondeterminism

In this section we briefly outline our contributions.

Defining limited nondeterminism

As mentioned above, we formalize a notion of limited nondeterministic adversaries that is meaningful in a cryptographic context.

For simplicity *in this section, we will consider limited nondeterministic adversaries to be adversaries with n^{ϵ}-bits of nondeterminism* for some constant $\epsilon < 1$ (unless otherwise stated). We refer the reader to the body for lemmas and theorems that precisely quantify the amount of nondeterminism required.

"Dream" hardness amplification

Andrew Yao [31] showed the direct product amplified the security of a weak one-way function (where an adversary can't invert with 1-1/poly(n) probability) to a one-way function (where an adversary can't invert with even negligible probability). While Yao's analysis rules out negligible adversarial advantage, it fails to rule out even $\epsilon(n)$ adversarial advantage for *any* concrete negligible ϵ. While it was initially suggested that this failure to "dream hardness amplification" (security beyond negligible) was due limits of our techniques, subsequent work [10] has even found explicit counterexamples that resist being amplified beyond negligible via the direct product (under an assumption). We show that dream hardness amplification is possible for any weak one-way function that is secure against limited nondeterministic adversaries.

In more detail, we say a function f is *ϵ-one-way against limited nondeterministic adversaries* if (a) it is efficiently computable, and (b) no adversary succeeds in inverting with more than ϵ probability. We say f is *subexponentially-one-way* if the above holds and the success chance of all adversaries is $O(2^{-n^c})$ for all $c \in (0, 1)$. Our (informal) main hardness amplification result follows. For the full version, see Theorem 14.

▶ **Theorem** ("Dream" hardness amplification, informal)**.** *Suppose f is a 1/2-one-way function against limited nondeterministic adversaries. Then the direct product $f^{\times 2n}$ of f is subexponentially-one-way for limited nondeterministic adversaries.*

Extracting $\omega(\log n)$ hardcore bits

A hardcore predicate g for a one-way function f produces a bit $g(x)$ that looks random even given the OWF evaluated on the same point $f(x)$. Oded Goldreich and Leonid Levin [14] showed that every one-way function admits a (randomized) hardcore predicate. This result was perhaps the most important ingredient in the constructing pseudorandom generators from one-way functions and has found a myriad of other applications in learning and coding. While Goldreich and Levin showed how to extract not just one hardcore bit but even $O(\log n)$ hardcore bits from any one-way function, it has remained open how to extract more.[3] We show that again, if the one-way function is secure against limited nondeterminism, then a straightforward generalization of Goldreich and Levin's predicate suffices. As a corollary, this yields efficient constructions of PRGs from OWFs that match the state-of-the-art when assuming exponential security of the underlying OWF. Our result follows, and for more details see Theorem 18.

▶ **Theorem** (Hardcore bits, informal). *Suppose that f is a one-way function against limited nondeterministic adversaries that have access to $\phi(n)$ nondeterministic bits. Then it is possible to extract $\phi(n)$ hardcore bits. In fact, if A is a random $\ell \times n$ matrix with $\ell \leq \phi(n)$, then $(A, f(X), AX) \overset{c}{\approx} (A, f(X), U_\ell)$ for all PPT distinguishers (arithmetic over $\mathbb{F}_2$).*

The initial HILL construction of a PRG (see [17]) from one-way functions was an incredible breakthrough, but required fairly large seed and number of one-way function calls (relative to the security parameter). A long and beautiful line of work has sought to streamline such constructions, making them more efficient. Constructions due to [16], [24], and [28] provide a seed length of $\tilde{O}(n^3)$. Additionally, at the cost of assuming exponential security, [24] provides a PRG construction with seed length $O(n^2)$. To highlight the power of our techniques, we show that limited nondeterministic security suffices to recover MP's result, and additionally, the seed length given in our construction smoothly interpolates between the nonadaptive PRG construction in [24], and the state of the art constructions using exponential security. For example, if we consider adversaries with $\log^2 n$ nondeterministic bits, we obtain PRGs with seed length $\omega(n^4/\log^4 n)$, and if we have a linear number of nondeterministic bits (e.g., $.00001n$), then we recover the exponential security construction with seed length $\omega(n^2)$.

▶ **Theorem** (PRG Seed length, see Theorem 25). *Suppose that f is a one-way function against limited nondeterministic adversaries that have access to ϕ nondeterministic bits. Then, for any function $s(n) = \omega(1)$, there exists a construction of a (n.u.) poly-time secure PRG with seed length*

$$O\left(\frac{n^4}{\phi^2}\left[\left(\frac{s\log n}{\phi}\right)^2 + \frac{s\log n}{\phi}\right] + n^2\right).$$

Brief reflection

We have seen two foundational applications of limited nondeterministic reductions in cryptography, but suspect this is only the tip of the iceberg. Recent years have seen a proliferation of complexity leveraging and reliance on strong subexponential security assumptions. In this work, we have seen that two results known (folklore) to easily follow from complexity leveraging via super-polynomial time reductions can instead be deduced generically from weaker

[3] Bellare, Stepanovs, and Tessaro [4] showed that differing-inputs obfuscation suffices, however this a strong notion which still cannot construct from standard assumptions.

assumptions (via a weaker and only slightly more complex form of complexity leveraging). It is our hope that these notions and techniques will enable future researchers to reduce their assumption footprint in settings that seemingly necessitate complexity leveraging.

1.2 Technical Overview

Hardness amplification

We recall the standard Yao-style hardness amplification via the direct product (i.e., q-wise concatenation): if an algorithm inverts $f^{\times q}$ with noticeable success, there is an index i at which "planting" a single instance preserves much of this success. In the PPT setting, one would simply repeat this algorithm a polynomial number of times and take the first successful output, and thus invert a single copy of f with high probability. In the limited-nondeterminism setting we follow the same high-level approach, with a specific caveat. When a reduction repeats a sub-protocol, if witness bits are not reused in subsequent repetitions, even $O(n)$ repetitions can already consume a sublinear witness budget. We will need to ensure that we are able to reuse the same witness string in each run of a repeated sub-protocol, so we use a one-shot method: generate a list of candidate random tapes via (strongly) universal hashing, fix the witness bits, run the subroutine on each tape, and then verify which run succeeded, returning the first output that did so.

Our main technique revolves around our technical Lemma 7, which informally states the following: given a search problem, if there is a limited nondeterministic adversary A that, for every input in some good set, succeeds with probability at least δ, then for any given ϵ we can construct a limited nondeterministic adversary A' that uses $O(\log(1/\delta))$ additional witness bits[4] and succeeds at the same search problem with success probability greater than $1-\epsilon$. Then, the reduction for hardness amplification of a direct product follows; a Yao-style argument shows that there exists an index i and "good set" $\mathcal{G}$, such all $y \in \mathcal{G}$ result in a success chance comparable to the overall success (up to an acceptable loss), and that $\mathcal{G}$ has sufficiently large density. Lemma 7 allows us to boost the success probability for every input in that good set without overshooting our budget of nondeterministic bits. Subexponential hardness follows because if such a good set exists with a success chance at least 2^{-n^c}, we only require $\Theta(n^c)$ additional nondeterministic bits to boost the reduction's success probability to greater than $\frac{1}{2}$.

Extracting many hardcore bits

At first, one might consider using nondeterminism to prove that a single hardcore bit is extremely hard. There are issues with this approach; for example, suppose that there was a limited nondeterministic adversary that, given $f(x)$ and some extremely hardcore predicate of x, is able to return a single bit of x. Any style of list decoding reduction would need to extract n-bits, and without some type of guarantee that witness bits could be reused, would result in exceeding our nondeterminism budget. In addition, Applebaum et al. [2] showed that nondeterministic *black-box* polynomial-time reductions cannot succeed in constructing predicates that are hard to predict with inverse superpolynomial advantage.

Instead of focusing on one bit at a time, we will show that if f is a one-way function against limited nondeterministic adversaries that have access to $\phi(n)$ nondeterministic bits, and A is a random $\ell \times n$ matrix with $\ell \leq \phi(n)$, then $(A, f(X), AX) \overset{c}{\approx} (A, f(X), U_\ell)$ for all

[4] Of note, the random-bit complexity of A' increases polynomially, but here our focus is on the number of nondeterministic-bit budget.

PPT distinguishers (arithmetic over $\mathbb{F}_2$). To do so, we open Yao's distinguisher-to-predictor hybrid argument. Then, letting $A_1, A_2, \ldots$ denote the rows of the matrix A. Via a hybrid argument, there is some index i such that the bit $A_i X$ is hard to distinguish from a random bit when given $(f(X), A_{<i}X)$. Via a standard distinguisher-to-predictor argument, we can obtain an adversary that can predict $A_i X$ when given A, $A_{<i}X$ and $f(X)$. If we can predict $\langle r, x\rangle$ for a random vector r, then we can use the Goldreich-Levin hardcore reduction on this bit to compute x and thus invert $f(x)$. However, an issue that arises in the PPT setting is to somehow correctly sample the $(i-1)$-bit prefix $A_{<i}X$. If i is too large (e.g., $i = \omega(\log n)$), guessing is infeasible. In our limited-nondeterministic world however, we argue that with high enough probability it suffices to draw $A_{<i}$ at random, hardcode it, and then nondeterministically sample the required bits $A_{<i}X$ we need to run the Goldreich-Levin reduction. Since we spend $i - 1 < \ell$ nondeterministic bits doing so, as long as $\ell \leq \phi(n)$ we do not exceed our nondeterministic bits budget, the reduction goes through.

Extracting more than $\log n$ hardcore bits in this way "immediately" yields efficient PRGs. The high-level idea of efficient PRGs (see [24]) from exponentially hard one-way functions is that one can extract $\Theta(n)$ hardcore bits instead of $O(\log n)$, which means that the constructions need fewer iterations (and thus a shorter seed) overall. Thus, if we assume that we started with limited nondeterministic hardness that allows one to extract just as many bits, we get an equivalently efficient construction. That said, our results also apply to the intermediate regime, where our limited nondeterministic budget is too small to extract the full $\Theta(n)$ hardcore bits, but still allows $\omega(\log n)$ hardcore bits, which leads to PRGs with seed lengths in between those given by standard one-way functions and exponentially secure one-way functions. We do note that our approach only deals with the PRG construction in [24] as a proof of concept. We expect analogous limited nondeterministic techniques to apply to other PRG frameworks.

2 Preliminaries

We let log and ln denote logarithms base 2 and e, respectively. We use calligraphic letters to denote sets, capital sans-serif letters to denote algorithms, lowercase letters to denote values, vectors, and functions; and uppercase letters to denote matrices or random variables, with the exception of $V_n(\cdot)$, which denotes a predicate. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. Let $\{0,1\}^* := \bigcup_{n\in\mathbb{N}} \{0,1\}^n$. For a set $\mathcal{S}$, we write $x \xleftarrow{u} \mathcal{S}$ to denote that x was drawn uniformly at random from $\mathcal{S}$, and we denote the uniform distribution over $\{0,1\}^n$ by U_n. Let $\mathrm{negl}(\cdot)$ and $\mathrm{poly}(\cdot)$ denote a negligible function and polynomial, respectively. For a matrix $M \in \{0,1\}^{m\times n}$, we identify M with the function $M\colon \{0,1\}^n \to \{0,1\}^m$ defined by $M(x) = Mx \mod 2$. We write $X \overset{c}{\approx} Y$ for computational indistinguishability, and $X \overset{c}{\approx}_\epsilon Y$ when the advantage is $\leq \epsilon$.

2.1 Defining Limited Nondeterministic Turing Machines

The study of problems that can be solved with limited nondeterminism was initiated by Kintala and Fischer [23]. A related definition tailored to nonuniform models of computation (and more) was introduced by Cai and Chen [7]. Flum, Grohe, and Weyer [13] demonstrated connections between limited nondeterministic complexity and notions of fixed-parameter intractability.

However, in this work, we are interested not only in decision problems, but also the nondeterministic complexity of *search* problems. Various notions of nondeterministic computations of functions have been introduced, for example [5, 6, 11, 20]. These definitions follow a

similar template, the key difference is what it means to "compute a function nondeterministically." The most relaxed notion, considered also in this work, means that computation does not actually compute a function but a mapping relation, the function nondeterministically chooses an output and different computation paths need not lead to the same output. More restricted notions require all accepting computation paths to output a single unique output.

Our definition of limited nondeterministic computation is a generalization of that of Cai and Chen [7] to the context of search problems (following Drucker's adaptation [11] of nondeterministic search problems to the nonuniform context).

▶ **Definition 1.** *For any function* $\phi\colon \mathbb{N} \to \mathbb{N}$*, we say a Turing machine* M *is a* randomized limited nondeterministic Turing machine with $\phi(n)$ nondeterministic bits*, if* M *has three input tapes consisting of*

- $x_1, \ldots, x_n$ *input bits,*
- $w_1, \ldots, w_{\phi(n)}$ *nondeterministic (i.e., witness) bits,*
- $r_1, \ldots, r_{\rho(n)}$ *random bits where* $\rho(n) = \mathrm{poly}(n)$ *is an arbitrary polynomial,*

and one output tape. We denote the output of M *as* $\mathsf{M}(x, w, r)$*. The set of all such Turing machines we denote* $\mathrm{RLNTM}(\phi(n))$*. Since it is always possible to brute-force a logarithmic number of bits, throughout this paper we will always assume that* $\phi(n) = \omega(\log n)$ *and* $\phi(n) < n$*.*

We analyze the ability of limited nondeterministic Turing machines to perform several cryptographically interesting tasks (e.g., inversion of one-way functions, breaking collision-resistant hashes, or breaking hardcore bits).

Our results apply to collision-resistant hashes and one-way functions, but more generally, they apply to any search problem wherein the adversary can check their answer. More formally, many of our main results can be phrased in terms of a machine $\mathsf{M} \in \mathrm{RLNTM}(\phi(n))$ solving an *efficiently verifiable search problem*, i.e. search problems where an adversary is given some efficiently computable predicate $V_n\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, and an element $y \in \mathcal{Y}$, and asked to find any $x^* \in \mathcal{X}$ such that x^* satisfies $V_n(x^*, y) = 1$. We formalize this in the following definition.

▶ **Definition 2.** *Let* $\{\mathcal{X}_n\}_{n\in\mathbb{N}}$ *and* $\{\mathcal{Y}_n\}_{n\in\mathbb{N}}$ *be families of sets. Let* $V_n : \mathcal{X}_n \times \mathcal{Y}_n \to \{0,1\}$ *be an efficiently computable predicate (i.e., a function computable in polynomial time). The* (efficiently verifiable) search problem defined by V_n *is as follows: given* $y \in \mathcal{Y}_n$*, find* $x \in \mathcal{X}_n$ *such that* $V_n(x, y) = 1$*.*

Let M *be a randomized limited nondeterministic Turing machine that takes input* $y \in \mathcal{Y}_n$*, uses* $\phi(n)$ *nondeterministic bits and* $\rho(n)$ *random bits, and outputs an element of* $\mathcal{X}_n$*. We are interested in the following probability, which we refer to as* M's success probability (for the search problem defined by V_n)*:*

$$\Pr_{y,r}\left[\exists w \in \{0,1\}^{\phi(n)} : V_n(\mathsf{M}(y, w, r), y) = 1\right],$$

where the probability is taken over $y \xleftarrow{u} \mathcal{Y}_n$ *and* $r \xleftarrow{u} \{0,1\}^{\rho(n)}$

▶ Remark 3. Our results apply to any problem that can be phrased as an efficiently verifiable search problem. We remark that any theorem or lemma that refers to search problems can be translated to relate to one-way functions and/or collision-resistant hashes, by picking a V_n so that finding (x, y) that satisfies $V_n(x, y) = 1$ is equivalent to solving the problem of interest.

Specifically, for a one-way function f, we phrase the inversion task as follows: let $\mathcal{X}_n := \{0,1\}^n$ be the domain of f, let $\mathcal{Y}_n := f(\{0,1\}^n)$ be the image of f, and let $V_n \colon \{0,1\}^n \times f(\{0,1\}^n) \to \{0,1\}$ be defined by $V_n(x,y) = 1$ if $f(x) = y$, and 0 otherwise.

For a given family of (keyed) collision-resistant hash functions $\{\mathcal{H}_k\}_{k\in\mathcal{K}}$, where $\mathcal{K}$ is the set of keys, we let $\mathcal{X}_n := \{0,1\}^n \times \{0,1\}^n$ be the set of pairs of inputs, $\mathcal{Y}_n := \mathcal{K}$ be the set of keys, and let $V_n \colon \{0,1\}^n \times \{0,1\}^n \times \mathcal{K} \to \{0,1\}$ be defined by $V_n(x_1,x_2,k) = 1$ if $h_k(x_1) = h_k(x_2) \wedge x_1 \neq x_2$, and 0 otherwise. Then, the search problem becomes: given a key k, find two elements $x_1 \neq x_2$ that map to the same image under the map h_k.

For any function $g \colon \mathcal{X} \to \mathcal{Y}$ and $m \in \mathbb{N}$, we use $g^{\times m} \colon \mathcal{X}^m \to \mathcal{Y}^m$ to denote the direct product map defined by $g^{\times m}(x_1,\ldots,x_m) = (g(x_1),\ldots,g(x_m))$.

2.2 Key Technical Lemma

▶ **Definition 4.** *Let $\mathcal{U}$ be a finite set and let $\mathcal{H} \subset \{h \colon \mathcal{U} \to \mathbb{F}_2^k\}$ be a finite family of functions from $\mathcal{U}$ to $\mathbb{F}_2^k$. We say that $\mathcal{H}$ is a* strongly universal hash family *if, $\forall u, u' \in \mathcal{U}$ with $u \neq u'$, $\forall z, z' \in \mathbb{F}_2^k$, we have*

$$\Pr_{h \xleftarrow{u} \mathcal{H}}[h(u) = z \wedge h(u') = z'] = 2^{-2k}$$

▶ **Proposition 5** (folklore)**.** *Let $k, \ell > 0$. Then*

$$\mathcal{H}^{\ell,k} := \{h_{A,v} \colon F_2^\ell \to \mathbb{F}_2^k \mid A \in \mathbb{F}_2^{k\times\ell}, v \in \mathbb{F}_2^k\}$$

is a strongly universal hash family, where $h_{A,v}(x) := Ax + v$ with addition over $\mathbb{F}_2^k$.

▶ **Lemma 6.** *Let $\mathcal{H} \subseteq \{h \colon \mathcal{W} \to \mathcal{V}\}$ be a strongly universal hash family. Let $\mathcal{U} \subseteq \mathcal{V}$ be a finite set, and let $p = \frac{|\mathcal{U}|}{|\mathcal{V}|}$. Then*

$$\Pr_{h \xleftarrow{u} \mathcal{H}}[\exists w \in \mathcal{W} : h(w) \in \mathcal{U}] \geq 1 - \frac{1}{|\mathcal{W}|\, p}$$

Proof. Let $\mathcal{W} = \{w_1,\ldots,w_{|\mathcal{W}|}\}$ and let $X_i := [h(w_i) \in \mathcal{U}]$ be the indicator random variable for the event that $h(w_i)$ is in $\mathcal{U}$. Notice that $X_1,\ldots,X_{|\mathcal{W}|}$ are pairwise independent, $\mathsf{E}[X_i] = \Pr_{v \xleftarrow{u} V}[v \in \mathcal{U}] = p$, and $\mathsf{var}[X_i] = p(1-p)$. Define $S := \sum_{i=1}^{|\mathcal{W}|} X_i$. Note $\mathsf{E}[S] = p\,|\mathcal{W}|$ and $\mathsf{var}[S] = |\mathcal{W}|\,\mathsf{var}[X_1] = |\mathcal{W}|p(1-p)$. We seek to estimate $\Pr[S = 0]$. If $S = 0$, then $|S - \mathsf{E}[S]| = |\mathcal{W}|p$, so applying Chebyshev's inequality we obtain

$$\Pr[S = 0] \leq \Pr[|S - \mathsf{E}[S]| \geq |\mathcal{W}|p] \leq \frac{\mathsf{var}[S]}{(|\mathcal{W}|p)^2} = \frac{|\mathcal{W}|\,p(1-p)}{(|\mathcal{W}|p)^2} = \frac{1-p}{|\mathcal{W}|p} \leq \frac{1}{|\mathcal{W}|p}.$$

Thus, $\Pr_{h \xleftarrow{u} \mathcal{H}}[\exists w \in \mathcal{W} : h(w) \in \mathcal{U}] = \Pr[S \geq 1] \geq 1 - \frac{1}{|\mathcal{W}|p}$. ◀

▶ **Lemma 7** (probability boosting for efficiently verifiable search problems)**.** *Let $\epsilon, \delta \in (0,1)$. Let $\{\mathcal{X}_n\}_{n\in\mathbb{N}}$ and $\{\mathcal{Y}_n\}_{n\in\mathbb{N}}$ be families of sets. Let $V_n : \mathcal{X}_n \times \mathcal{Y}_n \to \{0,1\}$ be an efficiently computable function. Let $\{\mathcal{G}_n \subseteq \mathcal{Y}_n\}_{n\in\mathbb{N}}$ be a family of sets. Suppose that M is a randomized limited nondeterministic Turing machine that uses $\phi(n)$ nondeterministic bits and $\rho(n)$ random bits, and suppose that for all $y \in \mathcal{G}_n$, M satisfies*

$$\Pr_{r \xleftarrow{u} \{0,1\}^{\rho(n)}}\left[\exists w \in \{0,1\}^{\phi(n)} : V(\mathsf{M}(y,w,r),y) = 1\right] > \delta$$

then there exists a randomized limited nondeterministic Turing machine M' *that uses* $\phi(n) + \lceil \log \frac{2}{\delta} \rceil$ *nondeterministic bits and* $(\rho(n)+1)\lceil \log \frac{2}{\delta} \rceil \lceil \log \frac{1}{\epsilon} \rceil$ *random bits, and for all* $y \in \mathcal{G}_n$, M' *satisfies*

$$\Pr_{r \in \{0,1\}^{\rho(n)}} \left[\exists w \in \{0,1\}^{\phi(n)+\lceil \log \frac{2}{\delta} \rceil} : V(\mathsf{M}'(y,w,r),y) = 1 \right] > 1 - \epsilon.$$

The subtlety lies in the fact that we cannot boost the probability through simple repetition. In a standard PPT setting, we could always boost the success probability by using multiple runs of an algorithm, but we need to be very careful in the limited nondeterministic setting. It is only possible to rerun sub-protocols if we can reuse the same nondeterministic bits for each run. To get around this fact, we use the same nondeterministic bits for each run, and then, at the very end, verify which run was actually successful. This subtlety, while at first seemingly unimportant, can cause some serious issues when importing many black box reductions from other settings to the limited nondeterministic setting. If the success of a particular sub-protocol is required for a later step to be run successfully, or alternatively, if there is no way to guarantee that the exact same witness string can be reused for each sub-protocol, then many reductions are not importable.

Proof. Let $\mathcal{R}$ denote the set of random strings for which M succeeds, i.e.

$$\mathcal{R} := \left\{ r \in \{0,1\}^{\rho(n)} \;\middle|\; \exists w \in \{0,1\}^{\phi(n)} : V(\mathsf{M}(y,w,r),y) = 1 \right\}.$$

The main idea of the proof is to use some nondeterministic bits, along with a hash function, to generate a string, call it r_1, that is contained in $\mathcal{R}$ with probability at least $\frac{1}{2}$. We then repeat this ℓ times, generating a list of candidate strings $r_1, \ldots, r_\ell$, so that for properly chosen ℓ, the probability that at least one such string is in $\mathcal{R}$ is greater than $1-\epsilon$. We then verify that one of these is (with high probability) the correct answer by computing $V(\mathsf{M}(y,w,r_i),y)$ for $i = 1, \ldots, \ell$. Details follow.

Let $\ell := \lceil \log \frac{2}{\delta} \rceil$ and $m := \lceil \log \frac{1}{\epsilon} \rceil$. We define M' as follows:

- Parse the nondeterministic bits into $(w, w') \in \{0,1\}^{\phi(n)} \times \{0,1\}^{\ell}$
- Parse the random bits into $(A_1, v_1), \ldots, (A_m, v_m) \in \mathbb{F}_2^{\rho(n)\times \ell} \times \mathbb{F}_2^{\rho(n)}$.
- For $i \in [m]$, compute $\tilde{r}_i \leftarrow A_i w' + v_i$, with arithmetic over $\mathbb{F}_2$.
- For $i \in [m]$, compute $\tilde{x}_i \leftarrow \mathsf{M}(y, w, \tilde{r}_i)$.
- Return the first $\tilde{x}_i$ such that $V(\tilde{x}_i, y) = 1$ if one exists; otherwise return $\perp$.

To analyze the probability that M' does not output $\perp$, we first analyze the chance that at least one $\tilde{r}_i$ is in $\mathcal{R}$. We know that $\Pr_{r \in \{0,1\}^{\rho(n)}} \left[\exists w \in \{0,1\}^{\phi(n)} : V(\mathsf{M}(y,w,r),y) = 1 \right] > \delta$, hence $|\mathcal{R}| > \delta \cdot 2^{\rho(n)}$.

We can use Lemma 6 to compute the chance that a single $\tilde{r}_i$ works, as follows: let $i \in [m]$, and note

$$\begin{aligned} \Pr_{(A,v) \xleftarrow{u} \mathbb{F}_2^{n\times\ell} \times \mathbb{F}_2^n} [\exists w \in \{0,1\}^{\ell} : Aw + v \in \mathcal{R}] &\geq 1 - \frac{1}{|\{0,1\}^{\ell}|\,\delta} \\ &= 1 - \frac{1}{2^{\ell}\delta} \\ &\geq 1 - \frac{1}{2} \\ &= \frac{1}{2} \end{aligned}$$

since $\ell = \lceil \log \frac{2}{\delta} \rceil$, and so $\frac{1}{2^\ell \delta} \geq \frac{1}{2}$. We have drawn m independent hashes, so

$$\begin{aligned}
&\Pr[\exists w \in \{0,1\}^\ell, \exists i \in [m] : \tilde{r}_i \in \mathcal{R}] \\
&\quad = 1 - \Pr[\forall w \in \{0,1\}^\ell, \forall i \in [m] : \tilde{r}_i \notin \mathcal{R}] \\
&\quad = 1 - \prod_{i \in [m]} \Pr[\forall w \in \{0,1\}^\ell, \tilde{r}_i \notin \mathcal{R}] \\
&\quad > 1 - \frac{1}{2^m} \geq 1 - 2^{-\log \frac{1}{\epsilon}} = 1 - \epsilon
\end{aligned}$$

As long as there is some index i so that $\tilde{r}_i \in \mathcal{R}$, we can nondeterministically pick w so that $V(\mathsf{M}(y, w, \tilde{r}_i), y) = 1$. Thus,

$$\Pr_r [\exists w : V(\mathsf{M}'(y, w, r), y) = 1] \geq \Pr[\exists w \in \{0,1\}^\ell, \exists i \in [m] : \tilde{r}_i \in \mathcal{R}] > 1 - \epsilon,$$

which completes the proof. ◀

3 Hardness Amplification via Concatenation

The core idea of the next lemma and proof comes from the proof of Proposition 1 in [9], but is included here for completeness and adapted into our framework.

▶ **Definition 8.** *Given an efficiently verifiable search problem defined by $V_n : \mathcal{X}_n \times \mathcal{Y}_n \to \{0,1\}$, for any integer $q > 0$ we define the* q-concatenation of V_n, *denoted $V_n^{\otimes q} : \mathcal{X}_n^{\times q} \times \mathcal{Y}_n^{\times q} \to \{0,1\}$ by*

$$V_n^{\otimes q}(x_1, \ldots, x_q, y_1, \ldots, y_q) := V_n(x_1, y_1) \wedge V_n(x_2, y_2) \wedge \cdots \wedge V_n(x_q, y_q)$$

Note that the next result makes no specific reference to limited nondeterminism in its proof, and holds when $\phi(n) := 0$.

▶ **Lemma 9** (concatenations of search problems have good sets)**.** *Let $\epsilon = \epsilon(n), \delta = \delta(n) \in (0,1)$ be functions. Let $V_n : \mathcal{X}_n \times \mathcal{Y}_n \to \{0,1\}$ be efficiently verifiable search problem. Let q be any integer satisfying $q > \lceil \frac{2}{\delta} \log \frac{2}{\epsilon} \rceil$. Suppose that some Turing machine $\mathsf{M} \in \mathrm{RLNTM}(\phi(n))$ has success probability at least $\epsilon(n)$ for the q-concatenation $V_n^{\otimes q}$. Let M' be the Turing machine defined as follows: on input $y \in \mathcal{Y}_n$,*
1. *choose $y_1, \ldots, y_q \xleftarrow{u} \mathcal{Y}_n$ and $i \xleftarrow{u} [q]$ uniformly at random, and set $y_i \leftarrow y$*
2. *run M on input $(y_1, \ldots, y_q)$ to obtain $x_1, \ldots, x_q$, and output x_i.*
Then $\mathsf{M}' \in \mathrm{RLNTM}(\phi(n))$, and M' satisfies

$$\Pr_y \left[\Pr_r \left[\exists w \in \{0,1\}^{\phi(n)} : V_n(\mathsf{M}'(y, w, r), y) = 1 \right] \geq \frac{\epsilon}{\delta q} \right] \geq 1 - \frac{\delta}{2} .$$

Proof. Suppose that some Turing machine $\mathsf{M} \in \mathrm{RLNTM}(\phi(n))$ has success probability at least $\epsilon(n)$ for the q-concatenation $V_n^{\otimes q}$.

Let M' be the Turing machine as defined in the statement of the lemma. Following the proof of Proposition 1 in [9], we analyze the success probability of M'.

Denote the events

$$\begin{aligned}
\mathsf{Success}_{\mathsf{M}'}(y) &\equiv \left[\exists w \in \{0,1\}^{\phi(n)} : V_n(\mathsf{M}'(y, w, r), y) = 1 \right], \\
\mathsf{Success}_{\mathsf{M}}(y_1, \ldots, y_q) &\equiv \left[\exists w \in \{0,1\}^{\phi(n)} : V_n^{\otimes q}(\mathsf{M}(y_1, \ldots, y_q, w, r), y) = 1 \right].
\end{aligned}$$

Let $\mathcal{S} \subseteq \mathcal{Y}_n$ be any set of density $\frac{\delta}{2}$. Then, since $q > \lceil \frac{2}{\delta} \log \frac{2}{\epsilon} \rceil > \lceil \frac{2}{\delta} \ln \frac{2}{\epsilon} \rceil$, using a union bound we see that

$$\Pr_{y_1,\ldots,y_q} [\mathsf{Success}_{\mathsf{M}}(y_1,\ldots,y_q) \wedge \exists j \in [q] : y_j \in \mathcal{S}] \geq \epsilon - (1-\delta/2)^q \geq \epsilon/2.$$

Thus, as we are randomly guessing the index i, if $\exists j \in [q] : y_j \in \mathcal{S}$, the chance that we guess such an index correctly is at least $1/q$, and so

$$\Pr_{y_1,\ldots,y_q,i} [\mathsf{Success}_{\mathsf{M}}(y_1,\ldots,y_q) \wedge y_i \in \mathcal{S}] \geq \frac{\epsilon}{2q}.$$

Estimating the success chance from above, by our construction of M', we can also obtain

$$\begin{aligned}
&\Pr_{y_1,\ldots,y_q,i} [\mathsf{Success}_{\mathsf{M}}(y_1,\ldots,y_q) \wedge y_i \in \mathcal{S}] \\
&\quad = \Pr_{y_1,\ldots,y_q,i} [\mathsf{Success}_{\mathsf{M}}(y_1,\ldots,y_q) \mid y_i \in \mathcal{S}] \cdot \Pr_{y_1,\ldots,y_q,i} [y_i \in \mathcal{S}] \\
&\quad = \frac{\delta}{2} \cdot \Pr_{y_1,\ldots,y_q,i} [\mathsf{Success}_{\mathsf{M}}(y_1,\ldots,y_q) \mid y_i \in \mathcal{S}] \\
&\quad \leq \frac{\delta}{2} \cdot \max_{y \in \mathcal{S}} \Pr [\mathsf{Success}_{\mathsf{M}'}(y)]
\end{aligned}$$

Putting these estimates together, we see that for any set $\mathcal{S} \subseteq \mathcal{Y}_n$ of density $\delta/2$,

$$\max_{y \in \mathcal{S}} [\Pr [\mathsf{Success}_{\mathsf{M}'}(y)]] \geq \frac{\epsilon}{\delta q}. \tag{1}$$

By way of contradiction, suppose that

$$\Pr_y \left[\Pr \left[\mathsf{Success}_{\mathsf{M}'}(y) \geq \frac{\epsilon}{\delta q} \right] \right] \leq 1 - \delta/2$$

Then there would exist some set $\mathcal{B} \subseteq \mathcal{Y}_n$ of density at least $\delta/2$ so that for every $y \in \mathcal{B}$, $\Pr [\mathsf{Success}_{\mathsf{M}'}(y)] < \frac{\epsilon}{\delta q}$, contradicting (1). Therefore,

$$\Pr_y \left[\Pr \left[\mathsf{Success}_{\mathsf{M}'}(y) \geq \frac{\epsilon}{\delta q} \right] \right] \geq 1 - \frac{\delta}{2},$$

completing the proof. ◀

▶ **Theorem 10** (inverter for concatenations of search problems). *Let $\epsilon, \delta \in (0,1)$ be arbitrary. Let $V_n : \mathcal{X}_n \times \mathcal{Y}_n \to \{0,1\}$ be efficiently verifiable search problem. Let $q > \lceil \frac{2}{\delta} \log \frac{2}{\epsilon} \rceil$ be any integer. Suppose there exists some Turing machine $\mathsf{M} \in \mathrm{RLNTM}(\phi(n))$ that uses $\phi(n)$ nondeterministic bits and $\rho(n)$ random bits, and has success probability at least $\epsilon(n)$ for the q-concatenation $V_n^{\otimes q}$.*

Then there exists a limited nondeterministic Turing machine M' that uses

$$\phi(n) + \left\lceil \log \frac{2\delta q}{\epsilon} \right\rceil$$

nondeterministic bits, and

$$(\rho(n)+1) \left\lceil \log \frac{2\delta q}{\epsilon} \right\rceil \left\lceil \log \frac{2}{\delta} \right\rceil$$

random bits, and such that the success probability of M' for V_n is at least $1-\delta$.

Proof. At a high level, this result follows directly from Lemma 9, which yields a Turing machine that has a large set of inputs such that for each one of those inputs, the success probability is bounded below by a certain amount. Once such a set exists, we can instantiate Lemma 7 to significantly boost the success probability, and obtain the desired machine. Details follow.

From Lemma 9, we obtain a Turing machine M' such that

$$\Pr_y\left[\Pr_r\left[\exists w \in \{0,1\}^{\phi(n)} : V_n(\mathsf{M}'(y,w,r),y)=1\right] \geq \frac{\epsilon}{\delta q}\right] \geq 1-\frac{\delta}{2},$$

or in other words, there exists a set $\mathcal{G}_n \subseteq \mathcal{Y}_n$ of density at least $1-\frac{\delta}{2}$ such that, for all $y \in \mathcal{G}_n$, $\Pr_r\left[\exists w \in \{0,1\}^{\phi(n)} : V(\mathsf{M}'(y,w,r),y)=1\right] \geq \frac{\epsilon}{\delta q}$. We are thus set up to apply Lemma 7 This gives us a Turing machine M'' such that for M'' uses $\phi(n)+\left\lceil \log \frac{2\delta q}{\epsilon}\right\rceil$ nondeterministic bits, and $(\rho(n)+1)\left\lceil \log \frac{2\delta q}{\epsilon}\right\rceil \left\lceil \log \frac{2}{\delta}\right\rceil$ random bits, and for every $y \in \mathcal{G}$, M'' will succeed on that y with probability greater than $1-\delta/2$, hence the overall success chance of M'' is greater than $(1-\delta/2)\cdot \Pr_y[y \in \mathcal{G}_n] = (1-\delta/2)^2 > 1-\delta$, completing the proof. ◀

4 Results for Cryptographic Hardness Amplification

4.1 One-Way Functions

▶ **Definition 11.** *We say a polynomial-time Turing machine is a* sublinearly-limited nondeterministic Turing machine*, if there exists $c>0$ and $\epsilon \in (0,1)$ such that $\mathsf{M} \in \mathrm{RLNTM}(cn^\epsilon)$. We denote the class of such Turing machines as* sublinNTM*. In other words,*

$$\mathrm{sublinNTM} := \bigcup_{\substack{\epsilon\in(0,1)\\ c>0}} \mathrm{RLNTM}(cn^\epsilon).$$

▶ **Definition 12.** *Given a function $f\colon \{0,1\}^* \to \{0,1\}^*$ and $\mathsf{M} \in \mathrm{RLNTM}(\phi(n))$, we refer to the probability*

$$\Pr_{x,r}[\text{“}\mathsf{M} \textit{ inverts } f\text{”}] := \Pr_{x,r}\left[\exists w \in \{0,1\}^{\phi(n)} : \mathsf{M}(f(x),w,r) \in f^{-1}(f(x))\right]. \tag{2}$$

▶ **Definition 13** (one-way for limited nondeterminism)**.** *Given an efficiently computable function $f\colon \{0,1\}^* \to \{0,1\}^*$, and a collection C of randomized limited nondeterministic Turing machines, and a function $\alpha\colon \mathbb{N} \to [0,1]$, we say that f is* $\alpha(n)$-one-way for C *if for all $\mathsf{M} \in \mathsf{C}$, for all sufficiently large n,*

$$\Pr_{x,r}[\text{“}\mathsf{M} \textit{ inverts } f\text{”}] \leq \alpha(n).$$

We say that f is subexponentially-one-way for C *if for all $\epsilon \in (0,1)$ and for all $\mathsf{M} \in \mathsf{C}$, for all sufficiently large n,*

$$\Pr_{x,r}[\text{“}\mathsf{M} \textit{ inverts } f\text{”}] \leq \frac{1}{2^{n^\epsilon}}.$$

▶ **Theorem 14.** *Let $p(n) = \mathrm{poly}(n)$, and let $f\colon \{0,1\}^* \to \{0,1\}^*$ be $(1-\frac{1}{p(n)})$-one-way for* sublinNTM*. Then $f^{\times(2np(n))}$ is subexponentially-one-way for* sublinNTM*.*

Proof. Suppose not. Then there exists $\epsilon' \in (0,1)$ and there exists a randomized limited nondeterministic Turing machine $\mathsf{M} \in \text{sublinNTM}$, such that for infinitely many n,

$$\Pr_{x,r}\left[\text{“}\mathsf{M} \text{ inverts } f^{\times(2np(n))}\text{”}\right] > 2^{-n^{\epsilon'}}.$$

Thus, there exists $c > 0$ and $\epsilon'' \in (0,1)$ such that $\mathsf{M} \in \text{RLNTM}(cn^{\epsilon''})$

We recall how to phrase the inversion task of a one-way function in terms of our definition of a search problem. Let $\mathcal{X}_n := \{0,1\}^n$ be the domain of f, let $\mathcal{Y}_n := f(\{0,1\}^n)$ be the image of f, and let $V_n \colon \{0,1\}^n \times f(\{0,1\}^n) \to \{0,1\}$ be defined by $V_n(x,y) = 1$ if $f(x) = y$, and 0 otherwise.

Then we can simply instantiate Theorem 10. Set $\delta = 1/p(n)$, $\epsilon = 2^{-n^{\epsilon'}}$, and $q = 2np(n)$. We note that $\frac{2}{\delta}\log\frac{2}{\epsilon} = 2p(n)(1+n^{\epsilon'}) < 2np(n) = q$. Thus, applying Theorem 10, we obtain a randomized limited nondeterministic Turing machine M' that has success probability at least $1-\delta = 1 - 1/p(n)$, and that uses

$$cn^{\epsilon''} + \left\lceil \log\frac{2\delta q}{\epsilon} \right\rceil < cn^{\epsilon''} + \log(4n) + 1 + n^{\epsilon'} = O(n^{\max\{\epsilon',\epsilon''\}})$$

nondeterministic bits. Hence, $\mathsf{M}' \in \text{sublinNTM}$, and succeeds at inverting f with probability greater than $1 - \frac{1}{p(n)}$, contradicting the fact that f is $(1-\frac{1}{p(n)})$-one-way for sublinNTM. ◀

▶ **Corollary 15.** *Let $f \colon \{0,1\}^* \to \{0,1\}^*$ be $\frac{1}{2}$-one-way for* sublinNTM. *Then $f^{\times(2n)}$ is subexponentially-one-way for* sublinNTM.

4.2 Collision-Resistant Hash Functions

▶ **Definition 16.** *A* family of hash functions *is a collection of polynomial-time computable functions $\mathcal{H} := \left\{\mathcal{H}_n \colon \{0,1\}^{\ell_{\mathsf{key}}(n)} \times \{0,1\}^{\ell_{\mathsf{in}}(n)} \to \{0,1\}^{\ell_{\mathsf{out}}(n)}\right\}$, where n is the security parameter, satisfying $\ell_{\mathsf{out}}(n) < \ell_{\mathsf{in}}(n)$. We refer to $\ell_{\mathsf{in}}(n)$, $\ell_{\mathsf{out}}(n)$, $\ell_{\mathsf{key}}(n)$ as the input length, output length and key size of the hash function, respectively. We use $h_\kappa \colon \{0,1\}^{\ell}_{\mathsf{in}}(n) \to \{0,1\}^{\ell}_{\mathsf{out}}(n)$ to denote the function $\mathcal{H}_n(\kappa,\cdot)$ associated with the key $\kappa \in \{0,1\}^{\ell}_{\mathsf{key}}(n)$. We call a pair (x_0,x_1) satisfying $x_0 \neq x_1$ and $h_\kappa(x_0) = h_\kappa(x_1)$ a* collision *for h_κ.*

Given a collection C of randomized limited nondeterministic Turing machines, we say $\mathcal{H}_n$ is an ϵ-CRHF (collision-resistant hash function) for C if for every $\mathsf{M} \in \mathsf{C}$,

$$\Pr[\kappa \leftarrow \{0,1\}^{\ell_{\mathsf{key}}(n)} : M(\kappa) \text{ outputs a collision for } h_\kappa] < \epsilon$$

We say that $\mathcal{H}_n$ is a subexponentially-secure CRHF for C *if for all $\epsilon \in (0,1)$ and for all $\mathsf{M} \in \mathsf{C}$, for all sufficiently large n,*

$$\Pr[\kappa \leftarrow \{0,1\}^{\ell_{\mathsf{key}}(n)} : M(\kappa) \text{ outputs a collision for } h_\kappa] < \frac{1}{2^{n^\epsilon}}.$$

▶ **Theorem 17.** *Let $p(n) = \text{poly}(n)$, and let $\mathcal{H}_n$ be a $(1-\frac{1}{p(n)})$-CRHF for* sublinNTM. *Then $\mathcal{H}_n^{\times(2np(n))}$ is subexponentially-secure CRHF for* sublinNTM.

Proof. Identical to the proof of Theorem 14, using the following search problem. Let $\mathcal{X}_n := \{0,1\}^n \times \{0,1\}^n$ be the set of pairs of inputs, $\mathcal{Y}_n := \mathcal{K}$ be the set of keys, and let $V_n \colon \{0,1\}^n \times \{0,1\}^n \times \mathcal{K} \to \{0,1\}$ be defined by $V_n(x_1,x_2,k) = 1$ if $h_k(x_1) = h_k(x_2) \wedge x_1 \neq x_2$, and 0 otherwise. ◀

5 How to Extract Many Hardcore Bits

In this section, we show how OWF with ϕ-limited nondeterministic hardness yields ϕ hardcore bits (against PPT algorithms). In subsequent sections, we show that this result further implies that OWF against $\Omega(n)$-limited nondeterministic hardness yield PRG constructions matching the state-of-the-art efficiency. Prior constructions [16, 24] with matching efficiency required the one-way function to be secure against exponential time adversaries (a strictly stronger assumption). In both prior work and ours, the resulting PRG is only achieves negligible distinguishing advantage against polynomial time adversaries.

▶ **Theorem 18.** *Suppose that $f\colon \{0,1\}^* \to \{0,1\}^*$ is $\alpha(n)$-one-way for* RLNTM$(\phi(n))$*, for some $\alpha(n) = \mathrm{negl}(n)$. Let $\ell(n)$ be any function such that $\ell(n) \le \phi(n)$. Let $A \xleftarrow{u} \{0,1\}^{\ell(n)\times n}$ be a random matrix. Let $X \xleftarrow{u} \{0,1\}^n$. Then $(A, f(X), AX) \overset{c}{\approx} (A, f(X), U_\ell)$, where AX is computed with arithmetic over $\mathbb{F}_2$. That is to say, for all PPT distinguishers* D*,*

$$|\Pr[\mathsf{D}(A, f(X), AX) = 1] - \Pr[\mathsf{D}(A, f(X), U_\ell) = 1]| = \mathrm{negl}(n)$$

Proof. Let $A_1, \ldots, A_\ell$ denote the rows of A, and for any $j \le k$ let $A_{j:k}$ denote the matrix with rows $A_j, A_{j+1}, \ldots, A_k$. Then we can write $(A, f(X), AX) = (A_{1:\ell}, f(X), A_{1:\ell}X)$.

We will use a hybrid argument. For all $i = 0, \ldots, \ell$, let $H_i := (A_{1:\ell}, f(X), A_{1:i}X, U_{\ell-i})$, where by convention we take $U_0 := \emptyset$.

$$\begin{aligned}
H_0 &:= (A, f(X), U_\ell)\\
H_1 &:= (A, f(X), A_1X, U_{\ell-1})\\
H_2 &:= (A, f(X), A_1X, A_2X, U_{\ell-2})\\
&\vdots\\
H_{\ell-1} &:= (A, f(X), A_1X, A_2X, \ldots, A_{\ell-1}X, U_1)\\
H_\ell &:= (A, f(X), A_1X, A_2X, \ldots, A_\ell X)
\end{aligned}$$

We will prove that if there exists an $i \in [\ell-1]$ so that $H_i \not\approx H_{i+1}$, then f is not $\alpha(n)$-one-way for RLNTM$(\phi(n))$.

Suppose that there exists some $i \in [\ell-1]$ so that $H_i \not\approx H_{i+1}$, and for the time being, suppose that we know which index this is (we will modify the argument to guess the index at the end of the proof).

Then, there exists some PPT distinguisher D, and there exists $\epsilon(n) > 1/\operatorname{poly}(n)$, such that for infinitely many n,

$$|\Pr[\mathsf{D}(A_{1:\ell}, f(X), A_{1:i}X, U_{\ell-i}) = 1] - \Pr[\mathsf{D}(A_{1:\ell}, f(X), A_{1:i}X, A_{i+1}X, U_{\ell-i-1}) = 1]| > \epsilon(n)$$

If we think of $A_{i+2:\ell}$ and $U_{\ell-i-1}$ as internal coins that D can throw, then we can write this as

$$|\Pr[\mathsf{D}(A_{1:i+1}, f(X), A_{1:i}X, U_1) = 1] - \Pr[\mathsf{D}(A_{1:i+1}, f(X), A_{1:i}X, A_{i+1}X) = 1]| > \epsilon(n)$$

Using a standard "distinguisher-to-predictor" style argument, we can obtain an adversary A that satisfies

$$\Pr_{A_{1:i+1},X}[\mathsf{A}(A_{1:i+1}, f(X), A_{1:i}X) = A_{i+1}X] \ge \frac{1}{2} + \frac{\epsilon(n)}{2}$$

Then, an application of Corollary 20 ensures that there is a set, call it $\mathcal{G}_x \subseteq \{0,1\}^n$, such that $\Pr_X[X \in \mathcal{G}_x] > \epsilon(n)/2$, and for all fixed $x \in \mathcal{G}_x$,

$$\Pr_{A_{1:i+1}}[\mathsf{A}(A_{1:i+1}, f(x), A_{1:i}x) = A_{i+1}x] \ge \frac{1}{2} + \frac{\epsilon(n)}{4}\ .$$

For any fixed $x \in \mathcal{G}_x$, we can apply Corollary 20 again to find a set $\mathcal{G}_A(x) \subseteq \{0,1\}^{i \times n}$ so that $\Pr_{A_{1:i}}[A_{1:i} \in \mathcal{G}_A(x)] > \frac{\epsilon(n)}{8}$, and for all fixed $\hat{A}_{1:i} \in \mathcal{G}_A(x)$,

$$\Pr_{A_{i+1}}\left[\mathsf{A}(\hat{A}_{1:i}, A_{i+1}, f(x), \hat{A}_{1:i}x) = A_{i+1}x\right] \geq \frac{1}{2} + \frac{\epsilon(n)}{8} .$$

The idea is to then simply hard-code $\hat{A}_{1:i}$ and $\hat{A}_{1:i}x$ into our adversary, and run the standard Goldreich-Levin hardcore bit reduction to invert f. The trick is that while we can easily hard-code $\hat{A}_{1:i}$, we need a few nondeterministic bits to guess the i-bit string $\hat{A}_{1:i}x$. If ℓ is too large, we would not be able to correctly guess this string with non-negligible chance. However, with i bits of nondeterminism, we can safely guess this string, as long as $x \in \mathcal{G}_x$ and $\hat{A}_{1:i} \in \mathcal{G}_A(x)$. Since $i \leq \ell$, as long as $\ell \leq \phi$, we will have enough budget of nondeterministic bits. We observe that independence yields $\Pr[X \in \mathcal{G}_x \wedge \hat{A}_{1:i} \in \mathcal{G}_A(x)] \geq \frac{\epsilon(n)^2}{16}$, which is still greater than $1/\operatorname{poly}(n)$. Conditioned on this event, the GL reduction [14,31] gives a decoder for finding x with a success chance that is also $> 1/\operatorname{poly}(n)$. Finally, we need to guess the correct index to actually apply the reduction, which we can do with probability $\geq 1/\ell$, and $\ell < n$, so overall, we obtain an inverting algorithm that uses exactly ℓ bits of nondeterminism and, with probability $> 1/\operatorname{poly}(n)$, finds an element of $f^{-1}(f(x))$. ◀

We do need the following lemma and corollary, which is likely folklore, but included here with proof for completeness.

▶ **Lemma 19.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $D\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a predicate, and let $\alpha > 0$. Suppose that $\Pr_{x,y}[D(x,y) = 1] > \alpha$. Then for all $0 \leq \beta < \alpha$, there exists a set $\mathcal{G} \subseteq \mathcal{X}$ such that $\Pr_x[x \in \mathcal{G}] \geq \frac{\alpha-\beta}{1-\beta}$ and for all $x \in \mathcal{G}$, $\Pr_y[D(x,y) = 1] > \beta$.*

Proof. We apply an averaging argument. Let $\mathcal{G} := \{x \in \mathcal{X} : \Pr_y[D(x,y) = 1] > \beta\}$. By way of contradiction, suppose that $\Pr_{x,y}[D(x,y) = 1] > \alpha$ and $\Pr_x[x \in \mathcal{G}] < \frac{\alpha-\beta}{1-\beta}$. Then

$$\begin{aligned}
\Pr_{x,y}[D(x,y) = 1] &= \Pr_{x,y}[D(x,y) = 1 | x \in \mathcal{G}] \Pr_x[x \in \mathcal{G}] + \Pr_{x,y}[D(x,y) = 1 | x \notin \mathcal{G}] \Pr_x[x \notin \mathcal{G}] \\
&\leq \Pr_{x,y}[D(x,y) = 1 | x \in \mathcal{G}] \frac{\alpha - \beta}{1 - \beta} + \Pr_{x,y}[D(x,y) = 1 | x \notin \mathcal{G}] \Pr_x[x \notin \mathcal{G}] \\
&\leq \Pr_{x,y}[D(x,y) = 1 | x \in \mathcal{G}] \frac{\alpha - \beta}{1 - \beta} + \beta \Pr_x[x \notin \mathcal{G}] \\
&\leq \frac{\alpha - \beta}{1 - \beta} + \beta \Pr_x[x \notin \mathcal{G}] \leq \frac{\alpha - \beta}{1 - \beta} + \beta \left(1 - \frac{\alpha - \beta}{1 - \beta}\right) = \alpha
\end{aligned}$$

and thus $\alpha < \Pr_{x,y}[D(x,y) = 1] \leq \alpha$, a contradiction. ◀

▶ **Corollary 20.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $D\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a predicate, and let $\mu > 0$. Suppose that $\Pr_{x,y}[D(x,y) = 1] > 1/2 + \mu$. Then there exists a set $\mathcal{G} \subseteq \mathcal{X}$ such that $\Pr_x[x \in \mathcal{G}] \geq \mu/2$ and for all $x \in \mathcal{G}$, $\Pr_y[D(x,y) = 1] > 1/2 + \mu/2$.*

Proof. Apply Lemma 19, with $\alpha = 1/2 + \mu$, and $\beta = \mu/2$, and note $\frac{\alpha-\beta}{1-\beta} \geq \alpha - \beta$. ◀

6 A More Efficient PRG

In this subsection, we show how to use the result from Section 5 to adapt PRG constructions (from OWF) that previously required exponential hardness assumptions to instead use linear limited nondeterministic hardness assumptions.

The state-of-the-art in efficient PRG constructions can be summarized as follows:

- [16]: Assuming exponentially-secure one-way functions, there exists a PRG construction that (adaptively) invokes the OWF $\omega(logn)$ times with seed length $\omega(n \log n)$.
- [24]: Assuming exponentially-secure one-way functions, there exists a PRG construction that *nonadaptively* invokes the OWF $\omega(logn)$ times with seed length $\omega(n^2)$.

In both cases, this efficiency (over constructions from poly-secure OWF) is achieved by exploiting the fact that exponentially-secure OWF yield a linear number of hardcore bits. While many other beautiful ideas are involved in these constructions (and we encourage the reader to delve deeper into the details of these amazing papers!), exponential-security is *only* used to get a linear number of hardcore bits from linear hashes (i.e. extended Goldreich-Levin).

6.1 Bits-Unpredictability Definitions

For the remainder of this section, we switch to a nonuniform adversary model. Specifically, we adjust our definition of randomized limited nondeterministic Turing machines with $\phi(n)$ nondeterministic bits to include an additional read-only advice tape $a_1, \ldots, a_{p(n)}$ for some $p(n) = \text{poly}(n)$. None of our proofs make use of the advice tape, so to reduce notation we will not explicitly write the dependence on the advice tape. However, it is required by the black box machinery in [24], which deals with nonuniform computation.

We use the same definitions as given in [24], repeated here for clarity.

▶ **Definition 21** (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,ϵ)-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 $X_n \xleftarrow{u} \{0,1\}^{m(n)}$ and $A \xleftarrow{u} \{0,1\}^{\lambda(n)}$:*

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

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

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

▶ **Definition 22** (Random bits unpredictability). *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,ϵ)-random-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 $X_n \xleftarrow{u} \{0,1\}^{m(n)}$ and $A \xleftarrow{u} \{0,1\}^{\lambda(n)}$:*

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

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

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

6.2 Results for Efficient PRGs

▶ **Theorem 23** (Bits-unpredictability from RLNTM). *Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a one-way function for* RLNTM$(\phi(n))$ *and let $\mathcal{M}_n = \{0,1\}^{n\times n}$ be the family of all $n \times n$ matrices. Let*

$$g = \left\{g_M\colon \{0,1\}^n \to \{0,1\}^{2n}\right\}_{M\in\mathcal{M}_n}$$

be defined by $g_M(x) := M(f(x)), M(x)$. Then g has $(n+\phi(n))$-bits-unpredictability.

Proof. This effectively follows immediately from Theorem 18. For an extended discussion, see the full version of this paper. ◀

We note that the above construction, when applied to standard OWF that is just secure against PPT adversaries, yields a function family with $(n+\log n)$-bits-unpredictability. In the case of an OWF for RLNTM$(\phi(n))$, we have a dramatic increase of bits-unpredictability, which will allow us to follow the constructions in [24] to construct a PRG with more efficient seed lengths. The key technical piece is what we have just shown, so it simply remains to follow the parameters through the constructions in [24].

▶ **Corollary 24.** *Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a one-way function for* RLNTM$(\phi(n))$. *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 $m'(n) = O(n^2/\phi(n))$, $\ell'(n) = O(n^2/\phi(n))$, $\lambda(n) = n^2$, and $k'(n) \geq m'(n)+n$. Moreover, the construction uses $\lceil 2n/\phi(n)\rceil + 3$ nonadaptive calls to f.

Proof. Assume $\phi(n) > \log(n)+1$. We instantiate Theorem 6.1 from [24] with $r(n) := \lceil 2n/\phi(n)\rceil + 3$, and the fact that the function family g given in Theorem 23 has $(n+\phi(n))$-bits-unpredictability. ◀

▶ **Theorem 25.** *Let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a one-way function for* RLNTM$(\phi(n))$. *Then, for any function $s(n) = \omega(1)$, there exists a construction of a (n.u.) poly-time secure PRG with seed length*

$$O\left(\frac{n^4 s^2 \log^2 n}{\phi^4(n)} + \frac{n^4 s \log n}{\phi^3(n)} + n^2\right).$$

Proof. The proof is nearly identical to the proof of Theorem 7.2 from [24], but uses Corollary 24. Once that piece is in place, it is just a matter of checking that parameters propagate correctly. For more details, see the full version of this paper ◀

7 Hardness Amplification via Randomized Iterates

Finally, we will illustrate how one can use limited nondeterministic tools for other types of hardness amplification. One of the main issues with hardness amplification via the direct product is that the input length blows up by a multiplicative factor, and so the construction is not "security-preserving". If one allows additional constraints on f, such as regularity, then better constructions exist, such as the randomized iterate (as given in [16]).

We will show how limited nondeterminism can also be applied to the randomized iterate. Very roughly, one of the key technical pieces is that our technique hinges on the existence of some large set of "good" inputs so that for every element in that set, the success probability

is lower bounded by some function. Then we can use the probability boosting Lemma 7 to make the overall success chance approximately the same as the density of our good set. In the context of the direct product, we showed that such a good set must exist for at least one of the indices of the direct product.

In the context of the randomized iterate, as [16] argue, there is some index of the iterate where one can plant the input to achieve a similar effect. The main difference is that the proof proceeds via contradiction, arguing that if we have a one-way function, then there must be sufficiently large "failing sets", or else using Lemma 7 we could boost the success probability too high.

▶ **Definition 26.** *Let $f\colon \{0,1\}^* \to \{0,1\}^*$ be a function. We say that f is* regular *if for all $n \in \mathbb{N}$, for all $x, x' \in \{0,1\}^n$, $\left|f^{-1}(f(x))\right| = \left|f^{-1}(f(x'))\right|$.*

▶ **Definition 27** (Definition 3.1 from [16])**.** *Let $n \in \mathbb{N}$, $f\colon \{0,1\}^n \to \{0,1\}^n$, and let $\mathcal{H}$ be a family of pairwise-independent length-preserving hash functions over strings of length n. For $k \in \mathbb{N}$, $x \in \{0,1\}^n$, and $\bar{h} = (h_1, \ldots, h_{k-1}) \in \mathcal{H}^{k-1}$, define the kth* randomized iterate *$f^k\colon \{0,1\}^n \times \mathcal{H}^{k-1} \to \{0,1\}^n$ recursively as*

$$f^k(x, \bar{h}) := f(h_{k-1}(f^{k-1}(x, (h_1, \ldots, h_{k-1}))))$$

where $f^1(x) := f(x)$. To make notation easier, for $m > k-1$, we let $f^k(x, h_1, \ldots, h_m) = f^k(x, h_1, \ldots, h_{k-1})$, which is to say we ignore unneeded extra hash functions as input to the randomized iterate.

▶ **Definition 28.** *Let $f\colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ and let A be an algorithm trying to invert f. We say that f has a $(\delta(n), \epsilon(n))$-*failing-set *for A if for large enough n, there exists a set $\mathcal{S}(n) \subseteq \{0,1\}^{\ell(n)}$ such that*

1. $\Pr[f(U_n) \in \mathcal{S}(n)] \geq \delta(n)$
2. $\forall y \in \mathcal{S}(n), \Pr[\exists w : \mathsf{A}(y, w) \in f^{-1}(y)] < \epsilon(n)$

The following lemma and proof are inspired by Claim 6.2 in [16], but are adapted to our setting of limited nondeterminism.

▶ **Lemma 29.** *Let $f\colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ be a $(1-\epsilon)$-one-way function for* $\mathrm{RLNTM}(\phi(n))$*, and let $\gamma(n)$ be any polytime computable function such that $\log(1/\gamma) = O(n^c)$ for some $c > 0$. Any algorithm using $\phi(n) - \lceil \log(2/\gamma) \rceil$ nondeterministic bits has an $(\epsilon - 2^{-n}, \gamma)$-failing set.*

Proof. Suppose, by way of contradiction, that $f\colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ is $(1-\epsilon)$-one-way, that γ be polytime computable, and there exists an algorithm A that uses $\phi(n) - \lceil \log(2/\gamma) \rceil$ nondeterministic bits and $\rho(n)$ random bits, such that it does not have a $(\epsilon - 2^{-n}, \gamma)$-failing-set, i.e. for infinitely many n's, there exists $\mathcal{L}(n) \subseteq \{0,1\}^{\ell(n)}$ such that

1. $\Pr[f(U_n) \in \mathcal{L}(n)] \geq 1 - \epsilon + 2^{-n}$
2. $\forall y \in \mathcal{L}(n), \Pr_r[\exists w : \mathsf{A}(y, w, r) \in f^{-1}(y)] \geq \gamma$

Intuitively, we construct an algorithm M^{A} that runs A as a subprotocol and uses the probability boosting Lemma 7, assuming that $y \in \mathcal{L}$. This would imply that for all $y \in \mathcal{L}$, M^{A} would invert successfully with probability extremely close to 1, while using $\phi(n) + \lceil \log(2/\gamma) \rceil$ nondeterministic bits. In fact, by using $(\rho(n)+1)\lceil \log \frac{2}{\gamma} \rceil \lceil \log \frac{1}{2^{-n}} \rceil = \mathrm{poly}(n)$ random bits, we can ensure that for all $y \in \mathcal{L}$, the probability of inverting y is greater than $1 - 2^{-n}$. Thus, the probability of inverting f is greater than $\Pr[f(U_n) \in \mathcal{L}(n)](1-2^{-n}) \geq (1-\epsilon+2^{-n})(1-2^{-n}) > 1-\epsilon$, contradicting the fact that f was $(1-\epsilon)$-one-way. ◀

▶ **Theorem 30.** *Let $\epsilon > 1/\operatorname{poly}(n)$ be some polynomial-time computable function, and let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a regular $(1-\epsilon)$-one-way function for $\mathrm{RLNTM}(\phi(n))$. Let $k = k(n) = \lceil 4n/\epsilon(n) \rceil$, let $\mathcal{H}$ be a family of pairwise-independent length-preserving hash functions, and let f^{k+1} be the kth randomized iterate. Let*

$$g\colon \{0,1\}^n \times \mathcal{H}^k \to f(\{0,1\}^n) \times \mathcal{H}^k$$

be defined as

$$g(x, \bar{h}) := (f^{k+1}(x, \bar{h}), \bar{h}).$$

Let ϵ' be a polytime computable function. Then for any function γ that satisfies $\gamma < \frac{\epsilon'^4 \epsilon^5}{9 \cdot 4^5 n^5}$, g is ϵ'-one-way for $\mathrm{RLNTM}(\phi(n) - \lceil \log(2/\gamma) \rceil)$.

Proof. By way of contradiction, suppose that f is a regular $(1-\epsilon)$-one-way function for $\mathrm{RLNTM}(\phi(n))$, and suppose that γ is some polytime computable function satisfying $\gamma < \frac{\epsilon'^4 \epsilon^5}{9 \cdot 4^5 n^5}$ and there exists an algorithm $\mathsf{A} \in \mathrm{RLNTM}(\phi(n) - \lceil \log(2/\gamma) \rceil)$, such that for infinitely many n's, A inverts g with probability greater than ϵ'. Let

$$\epsilon_{\mathsf{A}} := \Pr[\exists w : \mathsf{A}(f(U_n), w) \in f^{-1}(f(U_n))],$$

so that for infinitely many n's, $\epsilon_{\mathsf{A}} > \epsilon'$.

Then, recalling that $k(n) = \lceil 4n/\epsilon(n) \rceil$ applying Lemma 32, we obtain an algorithm B^{A} such that for all $\mathcal{S}(n) \subseteq f(\{0,1\}^n)$ with $\Pr[f(U_n) \in \mathcal{S}(n)] \geq \epsilon/2$, we have

$$\Pr[\mathsf{B}^{\mathsf{A}}(f(U_n)) \in f^{-1}(f(U_n)) \wedge f(U_n) \in \mathcal{S}(n)] \geq \frac{\epsilon_{\mathsf{A}}^4}{9k^5} = \frac{\epsilon_{\mathsf{A}}^4 \epsilon^5}{9 \cdot 4^5 n^5} > \frac{\epsilon'^4 \epsilon^5}{9 \cdot 4^5 n^5} .$$

Thus, B^{A} has no $(\epsilon/2, \frac{\epsilon'^4 \epsilon^5}{9 \cdot 4^5 n^5})$ fooling sets. However, applying Lemma 29, we know that B^{A} must have an $(\epsilon/2, \gamma)$ failing set, and by assumption $\gamma < \frac{\epsilon'^4 \epsilon^5}{9 \cdot 4^5 n^5}$, which is a contradiction. ◀

▶ **Corollary 31.** *Let $\epsilon > 1/\operatorname{poly}(n)$ be some polynomial-time computable function, and let $f\colon \{0,1\}^n \to \{0,1\}^n$ be a regular $(1-\epsilon)$-one-way function for $\mathrm{RLNTM}(\phi(n))$. Let $k = k(n) = \lceil 4n/\epsilon(n) \rceil$, let $\mathcal{H}$ be a family of pairwise-independent length-preserving hash functions, and let f^{k+1} be the kth randomized iterate. Let*

$$g\colon \{0,1\}^n \times \mathcal{H}^k \to f(\{0,1\}^n) \times \mathcal{H}^k$$

be defined as

$$g(x, \bar{h}) := (f^{k+1}(x, \bar{h}), \bar{h}).$$

Let ϵ' be a polytime computable function. Then g is ϵ'-one-way for

$$\mathrm{RLNTM}\left(\phi(n) - O(\log n) - 4\log(1/\epsilon')\right).$$

▶ **Lemma 32.** *Suppose that there exists a randomized limited nondeterministic algorithm A that uses ℓ nondeterministic bits and a polynomial time computable function ϵ_{A} such that for infinitely many n's, A inverts g with probability greater than ϵ_{A}. There exists an algorithm B^{A} which runs A as a subroutine exactly once and uses no additional nondeterministic bits, such that for any set $\mathcal{S}(n) \subseteq f(\{0,1\}^n)$ with $\Pr[f(U_n) \in \mathcal{S}(n)] \geq \epsilon/2$, it holds that*

$$\Pr[\mathsf{B}^{\mathsf{A}}(f(U_n)) \in f^{-1}(f(U_n)) \wedge f(U_n) \in \mathcal{S}(n)] \geq \frac{\epsilon_{\mathsf{A}}^4}{9k^5} \in \Omega\left(\epsilon_{\mathsf{A}}^4/k^5\right)$$

Proof. We directly import results from Section 6.3 in [16]. The algorithm in question is built using M^{A}, which is defined as follows:

M^{A}:

On input $(y, h_1, \ldots, h_i) \in f(\{0,1\}^n) \times \mathcal{H}^i$, where $i \in [k]$

- If $i = k$, let $w = y$.
  Otherwise, choose $h_{i+1}, \ldots, h_k \in \mathcal{H}$ uniformly at random and let

$$w = f^{k-1}(h_{i+1}(y), h_{i+2}, \ldots, h_k)$$

- Apply $\mathsf{A}(w, h_1, \ldots, h_k)$ to get $(x, h'_1, \ldots, h'_k)$.
- Return $f^i(x, h_1, \ldots, h_{i-1})$

Then define B^{A} as follows: on input $y \in \{0,1\}^n$, choose a random $i \in [k]$ and $\bar{h} \in \mathcal{H}^i$, and return $\bar{h}_i(\mathsf{M}^{\mathsf{A}}(y, \bar{h}))$. The analysis in Section 6.3 of [16] proved that

$$\Pr[\mathsf{B}^{\mathsf{A}}(f(U_n)) \in f^{-1}(f(U_n)) \wedge f(U_n) \in \mathcal{S}(n)] \geq \epsilon_{\mathsf{A}}^4 / 9k^5$$

which completes our proof. ◀

References

1. Divesh Aggarwal and Ueli Maurer. The leakage-resilience limit of a computational problem is equal to its unpredictability entropy. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 686–701. Springer, 2011. doi:10.1007/978-3-642-25385-0_37.
2. Benny Applebaum, Sergei Artemenko, Ronen Shaltiel, and Guang Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions (extended abstract). In David Zuckerman, editor, *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, volume 33 of *LIPIcs*, pages 582–600. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CCC.2015.582.
3. Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 299–315. Springer, 2003. doi:10.1007/978-3-540-45146-4_18.
4. Mihir Bellare, Igors Stepanovs, and Stefano Tessaro. Poly-many hardcore bits for any one-way function and a framework for differing-inputs obfuscation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 102–121. Springer, 2014. doi:10.1007/978-3-662-45608-8_6.
5. Ronald V. Book, Timothy J. Long, and Alan L. Selman. Quantitative relativizations of complexity classes. *SIAM J. Comput.*, 13(3):461–487, 1984. doi:10.1137/0213030.
6. Ronald V. Book, Timothy J. Long, and Alan L. Selman. Qualitative relativizations of complexity classes. *J. Comput. Syst. Sci.*, 30(3):395–413, 1985. doi:10.1016/0022-0000(85)90053-4.
7. Liming Cai and Jianer Chen. On the amount of nondeterminism and the power of verifying (extended abstract). In Andrzej M. Borzyszkowski and Stefan Sokolowski, editors, *Mathematical Foundations of Computer Science 1993, 18th International Symposium, MFCS'93, Gdansk, Poland, August 30 - September 3, 1993, Proceedings*, volume 711 of *Lecture Notes in Computer Science*, pages 311–320. Springer, 1993. doi:10.1007/3-540-57182-5_23.
8. Liming Cai, Jianer Chen, Rodney G. Downey, and Michael R. Fellows. On the structure of parameterized problems in NP. *Inf. Comput.*, 123(1):38–49, 1995. doi:10.1006/INCO.1995.1156.

9 Ran Canetti, Ron Rivest, Madhu Sudan, Luca Trevisan, Salil Vadhan, and Hoeteck Wee. Amplifying collision resistance: A complexity-theoretic treatment. In *Annual International Cryptology Conference*, pages 264–283. Springer, 2007.

10 Yevgeniy Dodis, Abhishek Jain, Tal Moran, and Daniel Wichs. Counterexamples to hardness amplification beyond negligible. In *Theory of Cryptography Conference*, pages 476–493. Springer, 2012. doi:10.1007/978-3-642-28914-9_27.

11 Andrew Drucker. Nondeterministic direct product reductions and the success probability of SAT solvers. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 736–745. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.84.

12 Amos Fiat and Moni Naor. Rigorous time/space tradeoffs for inverting functions. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 534–541. ACM, 1991. doi:10.1145/103418.103473.

13 Jörg Flum, Martin Grohe, and Mark Weyer. Bounded fixed-parameter tractability and $\log^2 n$ nondeterministic bits. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*, volume 3142 of *Lecture Notes in Computer Science*, pages 555–567. Springer, 2004. doi:10.1007/978-3-540-27836-8_48.

14 Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, 1989. doi:10.1145/73007.73010.

15 Judy Goldsmith, Matthew A. Levy, and Martin Mundhenk. Limited nondeterminism. *SIGACT News*, 27(2):20–29, 1996. doi:10.1145/235767.235769.

16 Iftach Haitner, Danny Harnik, and Omer Reingold. On the power of the randomized iterate. In *Annual International Cryptology Conference*, pages 22–40. Springer, 2006. doi:10.1007/11818175_2.

17 Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. doi:10.1137/S0097539793244708.

18 Alexander Healy, Salil P. Vadhan, and Emanuele Viola. Using nondeterminism to amplify hardness. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 192–201. ACM, 2004. doi:10.1145/1007352.1007389.

19 Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Trans. Inf. Theory*, 26(4):401–406, 1980. doi:10.1109/TIT.1980.1056220.

20 Lane A. Hemaspaandra and Mitsunori Ogihara. *The Complexity Theory Companion*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2002. doi:10.1007/978-3-662-04880-1.

21 Shuichi Hirahara, Rahul Ilango, and R. Ryan Williams. Beating brute force for compression problems. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 659–670. ACM, 2024. doi:10.1145/3618260.3649778.

22 Richard M. Karp and Richard J. Lipton. Some connections between nonuniform and uniform complexity classes. In Raymond E. Miller, Seymour Ginsburg, Walter A. Burkhard, and Richard J. Lipton, editors, *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 302–309. ACM, 1980. doi:10.1145/800141.804678.

23 Chandra M. R. Kintala and Patrick C. Fischer. Computations with a restricted number of nondeterministic steps (extended abstract). In John E. Hopcroft, Emily P. Friedman, and Michael A. Harrison, editors, *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*, pages 178–185. ACM, 1977. doi:10.1145/800105.803407.

24 Noam Mazor and Rafael Pass. Counting unpredictable bits: a simple prg from one-way functions. In *Theory of Cryptography Conference*, pages 191–218. Springer, 2023. doi:10.1007/978-3-031-48615-9_7.

25 Noam Mazor and Rafael Pass. The non-uniform perebor conjecture for time-bounded kolmogorov complexity is false. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 80:1–80:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ITCS.2024.80.

26 Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of facets (and some facets of complexity). In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 255–260. ACM, 1982. doi:10.1145/800070.802199.

27 Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, Redondo Beach, California, USA, November 12-14, 2000*, pages 32–42. IEEE Computer Society, 2000. doi:10.1109/SFCS.2000.892063.

28 Salil Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 817–836, 2012. doi:10.1145/2213977.2214051.

29 Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 231–240. ACM, 2010. doi:10.1145/1806689.1806723.

30 Ryan Williams. Non-uniform ACC circuit lower bounds. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011*, pages 115–125. IEEE Computer Society, 2011. doi:10.1109/CCC.2011.36.

31 Andrew C Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pages 80–91. IEEE, 1982.