Electronic Colloquium on Computational Complexity, Report No. 19 (2012)

On the complexity of constructing pseudorandom functions (especially when they don't exist)

Eric Miles* Emanuele Viola*

March 1, 2012

Abstract

We study the complexity of black-box constructions of pseudorandom functions (PRF) from one-way functions (OWF) that are secure against non-uniform adversaries. We show that if OWF do not exist, then given as an oracle any (inefficient) hard-to-invert function, one can compute a PRF in polynomial time with only $k(n)$ oracle queries, for any $k(n) = \omega(1)$ (e.g. $k(n) = \log^* n$). This result shows a limitation of a certain class of techniques for proving efficiency lower bounds on the construction of PRF from OWF. Our result builds on the work of Reingold, Trevisan, and Vadhan (TCC '04), who show that when OWF do not exist there is a pseudorandom *generator* (PRG) construction that makes only one oracle query to the hard-to-invert function. Our proof combines theirs with the Nisan-Wigderson generator (JCSS '94), and with a recent technique by Berman and Haitner (TCC '12).

Working in the same context (i.e. when OWF do not exist), we also construct a poly-time PRG with arbitrary polynomial stretch that makes non-adaptive queries to an (inefficient) one-bit-stretch oracle PRG. This contrasts with the well-known adaptive stretch-increasing construction due to Goldreich and Micali.

Both above constructions simply apply an affine function (parity or its complement) to the query answers. We complement this by showing that if the post-processing is restricted to only taking projections then non-adaptive constructions of PRF, or even linear-stretch PRG, can be ruled out.

We also use a result by Applebaum, Ishai, and Kushilevitz (J. Comput. '06) to rule out simple "hash-query-extract" PRF constructions, assuming the existence of OWF computable in logarithmic space.

*Northeastern University. Email: {enmiles,viola}@ccs.neu.edu. Supported by NSF grant CCF-0845003.

ISSN 1433-8092

1 Introduction

The notion of pseudorandomness is fundamental to the study of both cryptography and computational complexity. An efficient algorithm $G : \{0,1\}^n \to \{0,1\}^{n+s}$ is a (cryptographic) *pseudorandom generator* (PRG) with stretch s if no efficient adversary can distinguish a random output from a uniformly random string, except with negligible advantage. That is, for all poly(n)-time adversaries A, $|\Pr[A(G(U_n)) = 1] - \Pr[A(U_{n+s}) = 1]| < 1/n^{\omega(1)}$. A family of functions $\mathcal{F} = \{F_k \mid k \in \{0,1\}^n\}$ is a *pseudorandom function* (PRF) if no efficient adversary with oracle access can distinguish a random function in $\mathcal{F}$ from a uniformly random function, except with negligible advantage. That is, for all poly(n)-time adversaries A, $\left|\Pr_{F_k \leftarrow \mathcal{F}}[A^{F_k} = 1] - \Pr_{F \leftarrow \mathcal{U}}[A^F = 1]\right| < 1/n^{\omega(1)}$.

As the unconditional existence of PRG/PRF would imply P $\neq$ NP, their security is typically shown via a reduction to a hardness assumption. The weakest possible assumption is the existence of one-way functions (OWF), functions which are easy to compute but hard to invert. It is known that the existence of OWF is sufficient to construct PRG, i.e. there exists a construction $G^f : \{0,1\}^n \to \{0,1\}^{n+s}$ that has black-box access to a function $f : \{0,1\}^\ell \to \{0,1\}^\ell$ and computes a PRG whenever f is a OWF [HILL99]. In addition, it is known that a PRG with stretch $s \geq n$ is sufficient to construct PRF, again in a black-box manner [GGM86]. These can be combined to show that OWF suffice to construct PRF.

The efficiency of cryptographic constructions. Despite an intense research effort [BM84, Yao82, GGM86, GL89, HILL99, GKL93, HHR06, Hol06, HRV10, VZ12], constructions of PRG and PRF based on OWF remain relatively inefficient. Efficiency here can be measured in several different ways, including the seed length n relative to the OWF input length ℓ, the number of queries made to the OWF, and the circuit size of the construction. For example, the very recent work of Vadhan and Zheng [VZ12] gives a PRG with seed length $n = O(\ell^3)$, which is the best known. Also, the [VZ12] construction makes $\widetilde{O}(\ell^3)$ queries to the OWF, and of course this number lower bounds the circuit size. For PRF constructions, these parameters are even larger. However, it would be desirable to have PRG and PRF constructions where these parameters are smaller, especially if theoretical constructions aim to have direct practical applications.

Unfortunately, lower bounds on the efficiency of these constructions remain elusive. Essentially the only lower bound known is due to Gennaro et al. [GGKT05], who show that any PRG construction G^f must make at least $\Omega(s/\log T)$ queries per output when the OWF f has security T. Perhaps surprisingly, it is consistent with current knowledge that there exists a PRF construction that has seed length $n = O(\ell)$ and makes only a single query to the OWF per output.

Reingold, Trevisan and Vadhan [RTV04] offer a possible explanation for our inability to prove stronger lower bounds on PRG constructions. Their work also provides a useful taxonomy of cryptographic constructions, and before explaining their result we briefly review the portion of this that is relevant for us. (See §2 for formal definitions.) Most cryptographic constructions, including [HILL99] and [GGM86], are of a type known as *fully black-box*: G

has only black-box access to f, and any adversary A breaking G^f yields an efficient adversary C^A with black-box access to A that breaks f. Another type of construction, called weakly black-box in [RTV04], simply guarantees that G^f is a PRG whenever f is hard to invert; i.e., C may depend arbitrarily on the adversary A. We suggest the alternative terminology *primitive black-box*, to signify both that only the primitive (and not the adversary) is treated as a black-box, and that this is a "cruder" form of reduction.

The result of [RTV04] can be stated as follows: there exists an infinitely-often primitive black-box PRG construction $G^f : \{0,1\}^n \to \{0,1\}^{n+1}$ from a OWF $f : \{0,1\}^\ell \to \{0,1\}^\ell$, where G makes only one query to f and has seed length $n = 2\ell$.

We note that primitive constructions are pathological for two reasons. First, one does not expect to rule them out (since the construction can just ignore the oracle and construct a PRG in the "real-world"). Second, they do not immediately yield improved "real-world" efficiency (indeed, the construction in [RTV04] uses [HILL99] as a component, as do we).

Still, primitive constructions are important because any efficiency lower bound must account for them. For example, the lower bound of [GGKT05] also applies in the primitive black-box setting (in the sense that a construction breaking the aforementioned query/stretch tradeoff implies an unconditional pseudorandom generator). After more than twenty years since the seminal results in [GGM86, HILL99] and the result by Goldreich and Micali mentioned below, primitive constructions appear to offer the only available explanation for the lack of progress on efficiency lower bounds for fundamental cryptographic constructions.

Our results on PRF constructions. Our main result is an extension of [RTV04] to pseudorandom functions. We show that there is an (infinitely-often) primitive black-box PRF construction that makes only $k(n)$ queries to the OWF per output, for any $k(n) = \omega(1)$ (e.g. $k(n) = \log^* n$). Thus, one must avoid this construction to prove a super-constant lower bound on the query complexity of PRF constructions. This holds for OWF that are secure against non-uniform adversaries; it is an interesting open problem to obtain such a construction in the uniform setting.

Theorem 1.1. *For every $k(n) = \omega(1)$, there is a $\mathrm{poly}(n)$-time oracle algorithm $F^{(\cdot)} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ that makes $k(n)$ non-adaptive oracle queries and satisfies the following: for some $\ell = \Theta(\sqrt{n})$ and every function $f : \{0,1\}^\ell \to \{0,1\}^\ell$ that is hard to invert by poly-size circuits, $F^f(\cdot, U_n)$ is a PRF for infinitely many input lengths.*

The seed length of our construction is quadratic in ℓ, rather than linear as in [RTV04]. This stems from our use of the Nisan-Wigderson PRG [NW94] in the construction. Reducing the seed length of this PRG to $O(\ell)$ is a well-known open problem, and such an improvement would also reduce the seed length of our construction to $O(\ell)$.

We also make a modest step towards circumventing the above obstacle to proving black-box negative results. We observe that one can rule out a simple, yet arguably natural PRF construction using non-black-box techniques (jumping ahead, the work by Applebaum et al. [AIK06]). This natural construction is to "hash, then extract"; that is, we let the seed of the PRF specify a pairwise-independent (or even k-wise independent) hash function h and a

seed s of an extractor Ext, and output $\mathrm{Ext}(f(h(x)), s)$. We prove a negative result for such constructions whenever the hash function and the extractor are *linear*, that is, for any fixed seed they are a linear function of the input. We note that standard construction of hash functions [CW79, CG89, ABI86] and extractors [HILL99, Tre01] are indeed linear.

Theorem 1.2. *If there is a OWF computable in logarithmic space, and in particular if factoring is hard, then there is a OWF f such that $\mathcal{F} = \{F_{h,s}(x) := \mathrm{Ext}(f(h(x)), s)\}$ is not a PRF for any functions h and* Ext *that are linear for every fixed seed.*

To our knowledge, it is not known how to rule out such constructions using black-box techniques, primitive or otherwise.

The role of adaptivity in PRG constructions. Our main result also has implications for constructions that increase the stretch of a PRG. Though the definition of a PRG only requires stretch $s \geq 1$, all cryptographic and derandomization applications of which we are aware require much larger stretch, e.g. linear ($s = \Omega(n)$). An important and well-known result, due to Goldreich and Micali, is that the existence of a PRG with stretch $s = 1$ implies the existence of a PRG with stretch $s = \mathrm{poly}(n)$ for any desired polynomial. We briefly recall the construction that establishes this result (cf. [Gol01, §3.3.2]).

For a one-bit-stretch generator $G : \{0,1\}^n \to \{0,1\}^{n+1}$ and a positive integer k, let $G^k(x)$ denote the $(n+1)$-bit string resulting from k iterative applications of G, using x as the input for the first invocation, and the first n bits of the previous output as the input for subsequent invocations. Then, the "stretch-increasing" construction $H^{(\cdot)} : \{0,1\}^n \to \{0,1\}^m$ is

$$H^G(x) := G^1(x)_{n+1} \circ G^2(x)_{n+1} \circ \cdots \circ G^m(x)_{n+1}. \tag{1}$$

That is, H iteratively queries G as described above, and outputs the final bit of each answer.

An aspect of this construction of particular interest to us is that the queries are adaptive, in the sense that the ith query can be determined only after the answer to the $(i-1)$th query has been received. The presence of adaptivity in such constructions is especially important when considering the existence of cryptographic primitives in "low" complexity classes. The celebrated work of Applebaum et al. [AIK06], in combination with the (non-adaptive) construction of Haitner et al. [HRV10], demonstrates the existence of a PRG computable in NC^0 under the assumption that there exists a OWF computable in logarithmic space. However, the resulting PRG has *sub-linear* stretch, and the application of construction (1) would place it outside of NC^0.

Our results on adaptivity. We show that, in the primitive black-box setting, there is a non-adaptive stretch-increasing construction with arbitrary polynomial stretch. This in fact follows from Theorem 1.1, because the queries made by the PRF construction are non-adaptive, and because any PRG is also a OWF. This again holds under the assumption that the one-bit-stretch generator is secure against non-uniform adversaries.

Theorem 1.3. *For every constant $c = O(1)$, there is a $\mathrm{poly}(n)$-time oracle algorithm $H^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n^c}$ that makes n^c non-adaptive oracle queries and satisfies the following: for some $\ell = \Theta(\sqrt{n})$ and every one-bit-stretch PRG $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$, $H^G(\cdot)$ is a PRG for infinitely many input lengths. In addition, $H^{(\cdot)}$ has the form*

$$H^G(x) := \langle G(q_1(x)), r_1(x)\rangle \oplus t_1(x) \ \circ \ \cdots \ \circ \ \langle G(q_{n^c}(x)), r_{n^c}(x)\rangle \oplus t_{n^c}(x)$$

where $q_i : \{0,1\}^n \to \{0,1\}^\ell$ specifies the ith query, $r_i : \{0,1\}^n \to \{0,1\}^{\ell+1}$ specifies the ith parity function, and $t_i : \{0,1\}^n \to \{0,1\}$ specifies whether to complement the ith bit.

In Theorem 1.3, the post-processing consists of applying an input-dependent affine function to the query answers (this is also true in Theorem 1.1). We complement this result by showing that the post-processing cannot be weakened to taking projections. More specifically, we give a black-box separation showing that non-adaptive linear-stretch constructions cannot have post-processing that only takes projections of the query answers. This means that, in particular, there is no non-adaptive PRF construction with projection post-processing.

Theorem 1.4. *For all sufficiently large ℓ and for $n \le 2^{\sqrt{\ell}}$, there is no fully black-box construction $H^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n+s}$ of a generator with stretch $s \ge 5n/\log n$ and error $\epsilon \le 1/4$ from any one-bit-stretch generator $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ with error $\delta \ge 2^{-\sqrt{\ell}/30}$ and with security reduction size $t \le 2^{\sqrt{\ell}/30}$ of the form*

$$H^G(x) := G(q_1(x))_{b_1(x)} \circ \cdots \circ G(q_{n+s}(x))_{b_{n+s}(x)}$$

where $q_i : \{0,1\}^n \to \{0,1\}^\ell$ specifies the i-th query and $b_i : \{0,1\}^n \to [\ell+1]$ specifies the bit of the i-th answer to output.

Note that this holds even if G is secure against non-uniform adversaries. Theorem 1.4 indeed complements Theorem 1.3, because we observe in Theorem 4.3 that this impossibility result also extends to the primitive black-box setting, in the sense that any such construction implies NP/poly $\neq$ P/poly. (Recall that primitive black-box constructions cannot be ruled out without ruling out the existence of PRG, because if the latter exist a construction can just ignore the oracle and output a PRG). Note that the post-processing in the Goldreich-Micali construction (1) consists of taking projections, and thus linear-stretch constructions require either adaptive queries or post-processing the answers in a more sophisticated way than projecting.

It was pointed out to us by Benny Applebaum that Theorem 1.4 can be strengthened to rule out even AC^0 post-processing; we elaborate on this improvement in §1.2.

1.1 Techniques

Our main idea behind the proofs of Theorems 1.1 and 1.3 is to combine the construction of [RTV04] with the Nisan-Wigderson PRG [NW94]. The [RTV04] construction is proved secure by a case analysis, depending on the existence or non-existence of OWF. If OWF

exist, we use the results of Håstad et al. [HILL99] and Goldreich et al. [GGM86] that PRF also exist; the construction then ignores its oracle and simply outputs a PRF.

If OWF do not exist, this means that the oracle cannot be computed by poly-size circuits (since it is assumed to be hard to invert). We then use Goldreich-Levin [GL89] to transform the oracle into a Boolean function that is hard to compute by any family of poly-size circuits. Until now this is the argument in [RTV04]. (Actually [RTV04] is more involved because it works even in the uniform setting.) We next apply the Nisan-Wigderson construction to get an arbitrary polynomial-stretch PRG. This gives Theorem 1.3.

To turn this into a PRF and thus prove Theorem 1.1 we employ a recent technique by Berman and Haitner [BH12]. First, we observe that for every constant c one can obtain a "weak PRF" that is secure against adversaries which make at most $n^{\epsilon \cdot c}$ queries and have distinguishing advantage $\geq 1/n^{\epsilon \cdot c}$. This is obtained by hashing the input to select one of the n^c bits generated via Nisan-Wigderson as above.

To obtain a single PRF that has this security for every constant c, we XOR $k = \omega(1)$ copies of the weak PRF that are secure with respect to constants $c = 1, 2, \ldots, k$.

Since the Nisan-Wigderson construction is non-adaptive and each copy of the weak PRF is independent, this construction makes k non-adaptive queries to its oracle.

We note that, as is well-known, the proof of correctness of the Nisan-Wigderson construction requires non-uniformity, and this is what prevents this result from applying in the uniform setting.

To break the "hash, then extract" PRF construction (Theorem 1.2), we use the OWF computable in NC^0 given by Applebaum et al. [AIK06]. Then, every $F_{h,s} \in \mathcal{F}$ is computable by a low-degree polynomial and so can be distinguished by the results of Alon et al. [AKK$^+$03].

We now explain the proof of Theorem 1.4, the impossibility result for non-adaptive constructions with projection post-processing. Our proof is similar to the lower bound by Gennaro et al. mentioned previously [GGKT05], though we do not bound the number of queries. For simplicity, we first explain the proof in the case in which the construction always outputs the same bit of the answers, say the first bit (i.e. $b_i(x) = 1$ for all i).

We start by considering a (non-explicit) PRG $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ that is hard to break even for circuits that have oracle access to G. Such PRG are obtained in an unpublished manuscript of Impagliazzo [Imp96] and in a work by Zimand [Zim98]. (They work in a slightly different setting, however, obtaining a PRG with high probability in the random oracle model. For completeness we present a streamlined version of their arguments in §5.) By padding, we can modify our oracle to have the extra property that $G(x)_1 = x_1$ for every x. But now, H doesn't need to query G because each output bit $G(q_i(x))_{b_i(x)}$ can be replaced with $q_i(x)_1$. So we can consider an adversary A that breaks H^G by simply checking, given a challenge $z \in \{0,1\}^m$, whether there exists an x such that $z_i = q_i(x)_1$ for all i. This breaks H as soon as the output length is $\geq |x| + 1$. Since H doesn't use G anymore, neither does the adversary A. Hence the ability to access A does not compromise the security of G, contradicting Definition 2.1.

To generalize our result to constructions that output different bits (i.e. not always the

first one), we identify a set of indices $T \subseteq [\ell + 1]$ of size $\ell(1 - \Theta(1/\log \ell))$, such that for most input strings $x \in \{0,1\}^n$, most of the bits $b_i(x)$ chosen by H fall inside T. We exploit this fact by designing an oracle PRG G that reveals the first $|T|$ bits of its input on the set T; that is, $G(x)|_T = x_1 x_2 \cdots x_{|T|}$ for every input x. We then consider an adversary A that distinguishes H^G from uniform by examining, for every $x \in \{0,1\}^n$, only the bits i such that $b_i(x) \in T$, and checking if each bit matches the corresponding bit from the query $q_i(x)$. This turns out to break H as soon as the the output length is $\geq |x| + \Omega(|x|/\log |x|)$ (we do not attempt to optimize this value and content ourselves with anything sublinear). On the other hand, A depends on G just because of the knowledge of the set T, which means that oracle access to A does not compromise the security of G, again contradicting 2.1.

To obtain the result for primitive constructions, we observe that A can be computed in NP/poly, and hence under the assumption that NP/poly = P/poly we obtain a distinguisher.

1.2 More related work

The earlier work [Vio05] (which was later extended by [Lu06]) analyzes a type of pseudorandom generator construction that is very similar to ours. The constructions in [Vio05] make non-adaptive queries to an oracle one-way function, and then apply an arbitrary unbounded-fan-in constant-depth circuit (AC^0) to the outputs; [Vio05] shows that such constructions cannot have linear stretch. At first glance this construction is incomparable to Theorem 1.4, because it starts from a weaker primitive (one-way function instead of one-bit-stretch generator) but on the other hand allows for AC^0 postprocessing instead of just projections.

However, it was pointed out to us by Benny Applebaum that a strengthening of Theorem 1.4 follows from [Vio05] when combined with the works [AIK06] and [HRV10]. Specifically, a version of Theorem 1.4 holds even if the construction H is allowed to apply an AC^0 circuit to the output of the one-bit-stretch oracle PRG G (rather than just taking projections). We now elaborate on this improvement. (We also remark that at the moment this establishes a strengthened negative result only for constructions that start from a uniform hardness assumption, because Theorem 1.1 in [Vio05] is only proved for those.)

Assume that there exists a black-box construction $H^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n+s}$ of a PRG from a one-bit-stretch PRG which has the form $H^G(x) := C_x(G(q_1(x)), \ldots, G(q_{\mathrm{poly}(n)}(x)))$, where C_x is an AC^0 circuit generated arbitrarily from x and the functions q_i are arbitrary as before. Let $G^{(\cdot)}_{\mathrm{HRV}} : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ be the black-box construction of a PRG from a OWF given by [HRV10, Theorem 6.1]. This construction has the form $G^f_{\mathrm{HRV}}(x) := C'(x, f(x'_1), \ldots, f(x'_t))$ where C' is an NC^1 circuit and the x'_i are disjoint projections of the input x. Then, we can apply the compiler from [AIK06, Remark 6.7] to obtain a black-box construction $G^{(\cdot)}_{\mathrm{AIK}} : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ of a PRG from a OWF of the form $G^f_{\mathrm{AIK}}(x) := C''(x, f(x'_1), \ldots, f(x'_t))$, where now C'' is an NC^0 circuit (and thus is also an AC^0 circuit). (For both G_{HRV} and G_{AIK} the seed length is $\ell = \mathrm{poly}(m)$, where m is the input length of the oracle OWF, though the [AIK06] compiler does increase the seed length.) Finally, by combining H and G_{AIK}, we obtain a black-box construction $H^{(\cdot)}_* : \{0,1\}^n \to \{0,1\}^{n+s}$ of a PRG from a OWF which has the form $H^f_*(x) := C'''_x(f(q_1(x)), \ldots, f(q_{\mathrm{poly}(n)}(x)))$ where C'''_x is an AC^0

circuit. This is a contradiction to [Vio05, Theorem 1.1] when the oracle $f : \{0,1\}^m \to \{0,1\}^k$ has $\log^{\omega(1)} m < k \leq m^{O(1)}$ and the stretch s is greater than $n \cdot \log^{O(1)} m/k = o(n)$.

Finally, we mention that in a concurrent work, Bronson, Juma and Papakonstantinou [BJP11] also study non-adaptive black-box PRG constructions and obtain results which are incomparable to ours.

Organization In §2 we formally define the types of black-box constructions we consider. In §3 we give our PRF construction (Theorem 1.1) and the corresponding stretch-increasing construction (Theorem 1.3). In §4 we prove the black-box separation result (Theorem 1.4). Finally, in §5 we construct the one-bit-stretch oracle generator used in §4.

2 Black-box constructions

Here we give the formal definitions of the black-box constructions that we consider. To explain and motivate these, we start by sketching the proof of correctness of the Goldreich-Micali construction (1).

Suppose there is an adversary A that distinguishes $H^G(U_n)$ from U_m with advantage greater than $\epsilon \cdot m$. Using a hybrid argument, one can show that there exists a $k \in [m]$ such that A distinguishes the distributions $U_{k-1} \circ \left(H^G(U_n)|_{[m-(k-1)]}\right)$ and $U_k \circ \left(H^G(U_n)|_{[m-k]}\right)$ with advantage greater than ϵ. Then, we define a probabilistic oracle circuit $C^{(\cdot)}$ as follows: on input $(x, b) \in \{0,1\}^n \times \{0,1\}$, $C^{A,G}$ computes $H^G(x)$ using its oracle to G, chooses $y \in \{0,1\}^{k-1}$ uniformly at random, and then outputs $A\left(y \circ b \circ H^G(x)|_{[m-k]}\right)$. Depending on whether (x, b) was chosen from U_{n+1} or from $G(U_n)$, the input C gives to A will come from one of the two hybrid distributions that A can distinguish between, and so C distinguishes G with advantage greater than ϵ, contradicting G's pseudorandomness.

This argument is an example of a black-box reduction: it applies to any (possibly hard to compute) functions G and A, provided that we are given oracle access to them. We now formally define stretch-increasing PRG constructions in the fully black-box setting.

Definition 2.1 (Fully black-box stretch-increasing construction)**.** *An oracle function $H^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n+s}$ is a* fully black-box stretch-increasing construction *with security reduction size t of a generator with stretch s and error ϵ from any one-bit-stretch oracle generator $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ with error δ if the following holds:*

For every 1-bit stretch generator $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ and every adversary A, if A distinguishes H^G with advantage ϵ, i.e.

$$\left|\Pr[A(H^G(U_n)) = 1] - \Pr[A(U_{n+s}) = 1]\right| \geq \epsilon$$

then there is an oracle circuit $C^{(\cdot)}$ of size t that, when given oracle access to both A and G, distinguishes G with advantage δ, i.e.

$$\left|\Pr[C^{A,G}(G(U_\ell)) = 1] - \Pr[C^{A,G}(U_{\ell+1}) = 1]\right| \geq \delta.$$

We next formally define primitive black-box constructions. These differ from the above in that the adversary C may depend arbitrarily on A (i.e. C is not required to treat A as a black-box), but C is only required to exist in the case when A is efficient. We work in the asymptotic setting for these definitions because our results are cleaner to state in that setting. We also note that our primitive black-box constructions will hold for infinitely many (as opposed to sufficiently large) input lengths.

Definition 2.2 (Infinitely-often primitive black-box stretch-increasing construction)**.** *Let ℓ be a security parameter, and let $n = n(\ell)$ and $s = s(\ell)$. An oracle function $H^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n+s}$ is an* infinitely-often primitive black-box stretch-increasing construction *with stretch s if the following holds:*

For every c there exists c' such that for every ℓ_0 there exists $\ell \geq \ell_0$ such that for every $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$, if there exists a circuit A of size at most n^c that distinguishes H^G with advantage at least $1/n^c$, i.e.

$$\left|\Pr\left[A\left(H^G(U_n)\right) = 1\right] - \Pr\left[A\left(U_{n+s}\right) = 1\right]\right| \geq 1/n^c$$

then there exists a circuit $C^{(\cdot)}$ of size at most $\ell^{c'}$ that distinguishes G with advantage at least $1/\ell^{c'}$, i.e.

$$\left|\Pr\left[C^G\left(G(U_\ell)\right) = 1\right] - \Pr\left[C^G\left(U_{\ell+1}\right) = 1\right]\right| \geq 1/\ell^{c'}.$$

Definition 2.3 (Infinitely often primitive black-box PRF construction)**.** *Let ℓ be a security parameter and let $n = n(\ell)$. A set of oracle functions $\mathcal{F} = \left\{f^{(\cdot)} : \{0,1\}^n \to \{0,1\}\right\}$ is an* infinitely-often primitive black-box PRF construction *if the following holds:*

For every c there exists c' such that for every ℓ_0 there exists $\ell \geq \ell_0$ such that for every $g : \{0,1\}^\ell \to \{0,1\}^\ell$, if there exists a circuit $A^{(\cdot)}$ of size at most n^c that distinguishes $\mathcal{F}^g$ with advantage at least $1/n^c$, i.e.

$$\left|\Pr_{f \leftarrow \mathcal{F}}\left[A^{f^g} = 1\right] - \Pr_{f \leftarrow \mathcal{U}}\left[A^f = 1\right]\right| \geq 1/n^c$$

then there exists a circuit $C^{(\cdot)}$ of size at most $\ell^{c'}$ that inverts g with probability at least $1/\ell^{c'}$, i.e.

$$\Pr\left[C^g\left(g(U_\ell)\right) \in g^{-1}(g(U_\ell))\right] \geq 1/\ell^{c'}.$$

3 Non-adaptive primitive black-box constructions

In this section we prove Theorems 1.1 and 1.3. We first state the definitions of OWF and hard to compute functions that we will use.

Definition 3.1 (One-way function)**.** *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function. f is* hard to invert *if for all constants c, there is a constant ℓ_0 such that for all $\ell \geq \ell_0$ and every oracle circuit $C^{(\cdot)}$ of size at most ℓ^c we have $\Pr[C^f(f(U_\ell)) \in f^{-1}(f(U_\ell))] < 1/\ell^c$. If in addition f is computable by circuits of size $\mathrm{poly}(\ell)$, f is a* one-way function*.*

Definition 3.2 (Hard to compute infinitely often)**.** *Let $f : \{0,1\}^* \to \{0,1\}$ be a Boolean function. f is* hard to compute infinitely often *if for every c and ℓ_0, there exists $\ell > \ell_0$ such that for every circuit C of size at most ℓ^c, we have* $\Pr[C(U_\ell) = f(U_\ell)] < 1/2 + 1/\ell^c$.

In what follows we will sometimes make the assumption that "OWF do not exist", which means that for any function f that is hard to invert, every $\mathrm{poly}(\ell)$-sized circuit family fails to compute f on infinitely many input lengths. The following lemma constructs a function that is hard to compute infinitely often from one that is hard to invert, when OWF do not exist. This was also proved in [RTV04] in the uniform setting. Our proof, which relies on non-uniformity, is a bit simpler.

Lemma 3.3. *Assume that OWF do not exist, and let $f : \{0,1\}^* \to \{0,1\}^*$ be hard to invert. Then the Boolean function $f'(x,r) := \langle f(x), r\rangle$ is hard to compute infinitely often.*

Proof. Assume for contradiction that there exist constants c and ℓ_0 such that for all $\ell \geq \ell_0$, there exists a circuit C of size $\leq \ell^c$ such that $\Pr[C(U_\ell, U'_\ell) = \langle f(U_\ell), U'_\ell\rangle] \geq 1/2 + 1/\ell^c$, where U_ℓ and U'_ℓ denote independent instances of the uniform distribution on $\{0,1\}^\ell$ (we assume for simplicity that f is length-preserving). Then by the Goldreich-Levin theorem, there exist constants c' and ℓ'_0 such that for all $\ell \geq \ell'_0$, there exists a circuit C' of size $\leq \ell^{c'}$ such that $\Pr[C'(U_\ell) = f(U_\ell)] \geq 1/\ell^{c'}$. Now notice that C' computes a weak OWF; that is, the function computed by C' can only be inverted on strictly less than a $1 - 1/(2\ell^{c'})$ fraction of inputs by circuits of size $\mathrm{poly}(\ell)$ for sufficiently large ℓ, because any circuit which inverts C' on a $1 - 1/(2\ell^{c_1})$ fraction of inputs also inverts f on at least a $1/(2\ell^{c_1})$ fraction of inputs. However, using the standard direct product construction (originally due to Yao [Yao82]; see also [Gol01, Thm. 2.3.2]), this implies the existence of a OWF, contradicting the assumption that OWF do not exist. □

3.1 Stretch-increasing construction

In this subsection we prove Theorem 1.3, the non-adaptive stretch-increasing construction; this can be viewed as a warmup for our PRF construction in the subsequent subsection. We use the following definition of PRG.

Definition 3.4 (Pseudorandom generator)**.** *A function $G : \{0,1\}^n \to \{0,1\}^{n+s}$ is a (T,ϵ)-*pseudorandom generator *if $s \geq 1$ and for every oracle circuit $C^{(\cdot)}$ of size $\leq T$, we have $\left|\Pr[C^G(G(U_n)) = 1] - [C^G(U_{n+s}) = 1]\right| < \epsilon$.*

By virtue of Lemma 3.3, Theorem 1.3 will actually hold when the oracle is any function that is hard to invert. For completeness and to justify the term "stretch-increasing", we note that any one-bit-stretch PRG is hard to invert.

Lemma 3.5. *If $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ is a $(p(\ell), 1/p(\ell))$-pseudorandom generator for all polynomials p and sufficiently large ℓ, then it is hard to invert.*

Proof. Assume for contradition that there exist constants c and ℓ_0 such that for all $\ell > \ell_0$ there exists a circuit C of size $\leq \ell^c$ and an $\epsilon \geq 1/\ell^c$ such that $\Pr[C(G(U_\ell)) \in G^{-1}(G(U_\ell))] = \epsilon$. Then, define an adversary $A^{(\cdot)} : \{0,1\}^{\ell+1} \to \{0,1\}$ as follows: on input y, A^G computes $x = C(y)$, uses its oracle to G to check if $G(x) = y$, and outputs 1 iff this holds. We clearly have $|A| = \text{poly}(\ell)$ and $\Pr[A(G(U_\ell)) = 1] = \epsilon$.

Let $T \subseteq \text{Im}(G)$ be the set of outputs that C inverts, and note that $\sum_{y \in T} \Pr[G(U_\ell) = y] = \epsilon$. For each $y \in T$ we have $\Pr[G(U_\ell) = y] \geq 1/2^\ell$, and so $|T|/2^\ell \leq \epsilon$. Then, since A will only output 1 on inputs that C can invert and since no string outside $\text{Im}(G)$ can be inverted, we have $\Pr[A(U_{\ell+1}) = 1] = |T|/2^{\ell+1} \leq \epsilon/2$, and thus A distinguishes G from uniform with advantage $\geq \epsilon/2 = 1/\text{poly}(\ell)$. □

In order to apply the Nisan-Wigderson construction, we recall the notion of designs.

Definition 3.6 (Design)**.** *A collection of sets $S_1, \ldots, S_d \subseteq [n]$ is an (ℓ, α)-design if*

1. $\forall i : \ |S_i| = \ell$.
2. $\forall i \neq j : \ |S_i \cap S_j| \leq \alpha$.

Lemma 3.7 ([NW94])**.** *For any integers d and ℓ such that $\log d \leq \ell \leq d$, there exists a collection $S_1, \ldots, S_d \subseteq [4\ell^2]$ which is an $(\ell, \log d)$-design. For this collection, on input $j \in [d]$ the set S_j can be constructed in time* $\text{poly}(\ell)$.

We now give the proof of Theorem 1.3.

Theorem 3.8 (Theorem 1.3 restated)**.** *Let ℓ be a security parameter, and let $n = 17\ell^2$. Then for any constant $c > 1$, there exists an infinitely-often primitive black-box stretch-increasing construction $H^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n^c}$ from any one-bit-stretch generator $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$. In addition, $H^{(\cdot)}$ is computable in time* $\text{poly}(n)$*, and has the form*

$$H^G(x) := \langle G(q_1(x)), r_1(x) \rangle \oplus t_1(x) \ \circ \ \cdots \ \circ \ \langle G(q_{n^c}(x)), r_{n^c}(x) \rangle \oplus t_{n^c}(x)$$

where $q_i : \{0,1\}^n \to \{0,1\}^\ell$ specifies the ith query, $r_i : \{0,1\}^n \to \{0,1\}^{\ell+1}$ specifies the ith parity function, and $t_i : \{0,1\}^n \to \{0,1\}$ specifies whether to complement the ith bit.

Proof. If OWF exist, then by the results of [HILL99] there exists a PRG $H' : \{0,1\}^n \to \{0,1\}^{n^c}$. Then, the construction $H^{(\cdot)}$ is simply $H^G(z) := H'(z)$. Note that this can be achieved in the form stated in the theorem by setting $r_i(z) = 0^{\ell+1}$ for all i and z, and choosing the t_i appropriately to compute each bit of H'.

Now assume that OWF do not exist. Let $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ be any function, and define $f : \{0,1\}^{2\ell+1} \to \{0,1\}$ as $f(x,r) := \langle G(x), r \rangle$. Fix a constant $c > 1$, and define $n = 4(2\ell+1)^2$ (which is at most $17\ell^2$ for sufficiently large ℓ). Let $S_1, \ldots, S_{n^c} \subseteq [n]$ be the $(2\ell+1, c \log n)$ design guaranteed by Lemma 3.7. Then, the construction $H^G : \{0,1\}^n \to \{0,1\}^{n^c}$ is defined as

$$H^G(z) := f(z|_{S_1}) \circ \cdots \circ f(z|_{S_{n^c}}).$$

If there exists a polynomial p and a circuit family of size $p(\ell)$ which distinguishes G from uniform with advantage at least $1/p(\ell)$, then the theorem is trivially true. Thus, we can take G to be $(p(\ell), 1/p(\ell))$-pseudorandom for all polynomials p and sufficiently large ℓ. We will show that if H^G can be distinguished from random by an efficient adversary, then f can be computed efficiently with probability noticeably bigger than $1/2$, contradicting Lemmas 3.3 and 3.5.

Assume for contradiction that there exists a constant c_0 and a circuit family A of size n^{c_0} that distinguishes $H^G(U_n)$ from U_{n^c} with advantage $1/n^{c_0}$. Using the well-known equivalence between distinguishing and next-bit predicting [Yao82], this implies the existence of an $i \in [n^c]$ and a circuit family $A' : \{0,1\}^{i-1} \to \{0,1\}$ of size $n^{O(c_0)}$ such that $\Pr\left[A'(H^G(U_n)|_{[i-1]}) = H^G(U_n)_i\right] \geq 1/2 + 1/n^{c+c_0}$. Separating out the part of the input indexed by S_i, this can be rewritten as

$$\Pr_{(x,y)\leftarrow(U_{2\ell+1},U_n)}\left[A'(H^G(z)|_{[i-1]}) = H^G(z)_i\right] \geq 1/2 + 1/n^{c+c_0}, \tag{2}$$

where $z \in \{0,1\}^n$ is defined by $z|_{S_i} = x$ and $z|_{\overline{S_i}} = y|_{\overline{S_i}}$. By an averaging argument, there is a way to fix $y \in \{0,1\}^n$ such that (2) holds; from here on we assume that this y is fixed. For each $j \in [i-1]$, define the function $f_j : \{0,1\}^{2\ell+1} \to \{0,1\}$ as $f_j(x) := f(z)$, where now z is defined by $z|_{S_i \cap S_j} = x_1 x_2 \cdots x_{|S_i \cap S_j|}$ and $z|_{\overline{S_i \cap S_j}} = y|_{\overline{S_i \cap S_j}}$. Note that since $S_i \cap S_j \leq c \log n$ and y is fixed, each f_j is computable by a circuit family of size $\mathrm{poly}(n) = \mathrm{poly}(\ell)$. Finally, define the circuit family $A'' : \{0,1\}^{2\ell+1} \to \{0,1\}$ as $A''(x) := A'(f_1(x), \ldots, f_{i-1}(x))$. It can be easily checked that A'' has size $\mathrm{poly}(\ell)$ and correctly computes f on a random input with probability at least $1/2 + 1/n^{c+c_0}$. □

3.2 PRF construction

We now extend the previous construction to get a low-query, non-adaptive primitive black-box PRF construction from any OWF f. The proof again proceeds via a case analysis, as follows. In the case when OWF exist, [HILL99] and [GGM86] give a PRF. If OWF do not exist, we again use $\langle f(x), r\rangle$ in the Nisan-Wigderson construction. By combining this with a pairwise-independent hash function, we obtain for any i a "weak PRF" $\mathcal{F}_i$, which has security $n^{\Omega(i)}$ when $i = O(1)$. Then by taking $k(n) = \omega(1)$ and $\mathcal{F} := \bigoplus_{j \leq k} \mathcal{F}_j$, and showing a reduction from breaking $\mathcal{F}_i$ to breaking $\mathcal{F}$, we obtain that $\mathcal{F}$ is a PRF because any poly-size circuit breaking $\mathcal{F}$ contradicts the hardness of $\mathcal{F}_i$ for sufficiently large $i = O(1) \leq k$.

Theorem 3.9 (Theorem 1.1 restated)**.** *Let ℓ be a security parameter, and let $n = 16\ell^2$. For any $k = k(n) = \omega(1)$, there is an infinitely-often primitive black-box PRF construction $\mathcal{F} = \left\{F^{(\cdot)} : \{0,1\}^n \to \{0,1\}\right\}$ from any oracle function $f : \{0,1\}^\ell \to \{0,1\}^\ell$, of the form*

$$F^f(x) := \bigoplus_{1 \leq i \leq k} \langle f(q_i(x)), r_i(x)\rangle \oplus t(x)$$

where $q_i : \{0,1\}^n \to \{0,1\}^\ell$ specifies the ith query, $r_i : \{0,1\}^n \to \{0,1\}^\ell$ specifies the ith parity function, and $t : \{0,1\}^n \to \{0,1\}$ specifies whether to complement the output bit. The

functions q_i, r_i, and t are specified by the $O(n)$-bit seed of $F^{(\cdot)} \in \mathcal{F}$ and are all $\mathrm{poly}(n)$*-time computable.*

Proof. Note that the theorem is trivially true for any oracle that is not hard to invert, so we assume throughout that f is hard to invert.

If OWF exist, then by [HILL99] and [GGM86] we know that infinitely-often PRF exist (in fact they exist for all sufficiently large input lengths), so we can take $F^{(\cdot)}$ to be the construction that ignores its oracle and outputs such a PRF. This can be achieved in the stated form by setting $r_i(x) = 0^\ell$ for all i and x, and choosing t appropriately to compute the PRF.

Now assume that OWF do not exist. We give the construction $\mathcal{F}_i$, from which we will construct $\mathcal{F} := \bigoplus_{j \le k} \mathcal{F}_j$.

Let $f' : \{0,1\}^* \to \{0,1\}$ be defined on even input lengths by $f'(x,r) := \langle f(x), r \rangle$. For any even $\ell \in \mathbb{N}$, let $n = 4\ell^2$.[1] For an integer $i \le n/\log n$, let $\mathcal{H} = \{h : \{0,1\}^n \to [n^i]\}$ be a pairwise-independent hash family, and let $S_1, \ldots, S_{n^i} \subseteq [n]$ be the $(\ell, i \log n)$-design guaranteed by Lemma 3.7. (The bound $i \le n/\log n$ is to guarantee $n^i \le 2^n$.)

Then, $\mathcal{F}_i = \left\{ F_{h,z} : \{0,1\}^n \to \{0,1\} \;\middle|\; h \in \mathcal{H},\ z \in \{0,1\}^n \right\}$ is defined as

$$F_{h,z}(x) := f'\left(z\big|_{S_{h(x)}} \right)$$

Note that $F_{h,z}(x)$ has the form $\langle f(q_i(x)), r_i(x) \rangle$. As i is bounded, $F_{h,z}$ is computable (with oracle access to f) in time n^α for a universal constant α independent of i.

The following claim relates the hardness of distinguishing $\mathcal{F}_i$ to that of computing f'.

Claim. *If there exists a circuit of size $\le n^{i/4}$ that distinguishes $\mathcal{F}_i$ with advantage $\ge 1/n^{i/4}$, then there exists a circuit of size $\ell^{O(i)}$ that computes $f'(U_\ell)$ with probability $\ge 1/2 + 1/\ell^{3i}$.*

Before proving this claim, we show how it implies the theorem. Let $k = k(n)$ be any monotonic non-decreasing integer function such that $k = \omega(1)$ and $k \le n/\log n$, and define $\mathcal{F} := \bigoplus_{j \le k} \mathcal{F}_j$. We will show that a distinguisher of size n^c for $\mathcal{F}$ implies the existence of a distinguisher of size $n^{O(c)}$ for $\mathcal{F}_i$. Then by choosing an appropriate $i = \Theta(c)$ and letting n be sufficiently large to guarantee $k \ge i$, this will imply the existence of a poly-size circuit computing f', in contradiction to Lemma 3.3.

Assume for contradiction that there exist constants c and n_0 and a circuit (family) $A^{(\cdot)}$ of size n^c such that $A^{(\cdot)}$ distinguishes $\mathcal{F}$ from uniform with advantage $\ge 1/n^c$ for all input lengths $n \ge n_0$. For any $i \le k$, we construct a circuit $A_i^{(\cdot)}$ that distinguishes $\mathcal{F}_i$ from uniform with the same advantage on the same input lengths, as follows: $A_i^{\mathcal{O}}$ simulates $A^{(\cdot)}$, and answers its oracle queries with $\mathcal{O} \oplus \bigoplus_{j \ne i} \mathcal{F}_j$. The key point is that if $\mathcal{O} = \mathcal{F}_i$ then the simulated oracle is $\mathcal{F}$, and if $\mathcal{O}$ is uniform then the simulated oracle is uniform. The size of A_i is $\le n^c \cdot k(n) \cdot n^\alpha$, where n^α is the size needed to compute each $\mathcal{F}_j$. Let $c' = c + O(1)$ be a constant (independent of i) such that $|A_i| \le n^{c'}$, and note that A_i distinguishes $\mathcal{F}_i$ with advantage $\ge 1/n^{c'}$ on all input lengths $n \ge n_0$.

[1] We are now using ℓ to refer to the input length of f', which is twice the input length of f.

Now let $i = \lceil 4c' \rceil$, and let $n'_0 \geq n_0$ be the smallest integer such that $k(n'_0) \geq i$. Then A_i has size $\leq n^{i/4}$ and distinguishes $\mathcal{F}_i$ with advantage $\geq 1/n^{i/4}$ on all input lengths $n \geq n'_0$. By the claim this gives a circuit of size $\ell^{O(1)}$ that computes $f'(U_\ell)$ with probability $1/2 + 1/\ell^{O(1)}$ for all input lengths $\ell \geq \sqrt{n'_0}/2$, which contradicts Lemma 3.3.

We now prove the claim.

Proof of Claim. Let $A^{(\cdot)}$ be an oracle circuit of size $\leq n^{i/4}$ such that

$$\left| \Pr_{F_{h,z} \leftarrow \mathcal{F}_i} \left[A^{F_{h,z}} = 1 \right] - \Pr_{F \leftarrow \mathcal{U}} \left[A^F = 1 \right] \right| \geq \frac{1}{n^{i/4}}.$$

Let $B : \{0,1\}^{n^i} \to \{0,1\}$ be the circuit of size $|A| \cdot n^{O(i)}$ which, on input x, selects a uniform $h \in \mathcal{H}$ and simulates $A^{(\cdot)}$ by answering query $q \in \{0,1\}^n$ with $x_{h(q)} \in \{0,1\}$. By construction we have

$$\Pr_{z \in \{0,1\}^n} \left[B\left(f'(z|_{S_1}), \ldots, f'(z|_{S_{n^i}}) \right) = 1 \right] = \Pr_{F_{h,z} \leftarrow \mathcal{F}_i} \left[A^{F_{h,z}} = 1 \right].$$

Let E be the event, over the choice of a uniform function $F \leftarrow \mathcal{U}$ and a uniform $h \in \mathcal{H}$, that $A^{F \circ h}$ makes two queries $q \neq q'$ such that $h(q) = h(q')$; it can be shown that $\Pr_{F,h}[E] < |A|^2/n^i \leq 1/n^{i/2}$ by a collision-probability argument. Then,

$$\Pr_{z \in \{0,1\}^{n^i}} \left[B(z) = 1 \mid \neg E \right] = \Pr_{F \leftarrow \mathcal{U}} \left[A^F = 1 \right]$$

and thus

$$\left| \Pr_{z \in \{0,1\}^n} \left[B\left(f'(z|_{S_1}), \ldots, f'(z|_{S_{n^i}}) \right) = 1 \right] - \Pr_{z \in \{0,1\}^{n^i}} \left[B(z) = 1 \right] \right| \geq \frac{1}{n^{i/4}} - \Pr[E] > \frac{1}{2n^{i/4}}.$$

(Technically, this inequality holds either for B or for the circuit which outputs the opposite of B; we take B to be the circuit for which it holds. Also, note that we can take B to be a deterministic circuit by fixing the choice of $h \in \mathcal{H}$ that maximizes the above difference.)

By the Nisan-Wigderson analysis (cf. proof of Theorem 3.8), the fact that each distinct $S_j, S_{j'}$ have overlap $\leq i \log n$ implies the existence of a circuit C of size $\leq |B| \cdot n^{O(i)} = \ell^{O(i)}$ that computes f' correctly on a $1/2 + 1/2n^{5i/4} \geq 1/2 + 1/\ell^{3i}$ fraction of inputs of size ℓ. □

This completes the proof of the theorem. □

3.3 An impossible PRF construction

Here we briefly mention a seemingly natural approach for constructing PRF from OWF, and show that it fails for a specific choice of the OWF. For simplicity of notation we take the PRF and OWF to have the same input length n.

The approach is to "hash, then extract"; that is, we let the seed of the PRF specify a pairwise-independent hash function $h : \{0,1\}^n \to \{0,1\}^n$, and a seed $s \in \{0,1\}^m$ of an extractor $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$, and output

$$F^f(x) := \mathrm{Ext}(f(h(x)), s).$$

More generally, one can hash the input to poly(n) k-wise independent samples for $k = O(1)$, apply f to each sample, and then extract. All the considerations in this section apply to this more general construction as well.

We observe that such an approach cannot produce a PRF without either (a) violating many widely-held cryptographic assumptions or (b) relying on properties of Ext other than its output being statistically close to uniform.

Theorem 1.2. *If there is a OWF computable in logarithmic space, and in particular if factoring is hard, then there is a OWF f such that $\mathcal{F} = \{F_{h,s}(x) := \mathrm{Ext}(f(h(x)), s)\}$ is not a PRF for any functions h and* Ext *that are linear for every fixed seed.*

Proof sketch. To show that this approach cannot work, we use the NC^0 OWF given by Applebaum, Ishai and Kushilevitz [AIK06], any linear hash function [CW79, CG89, ABI86], and any extractor that becomes linear when the seed is fixed [HILL99, Tre01].

Theorem 3.10 ([AIK06])**.** *If there is a OWF computable in logarithmic space, then there is a OWF computable in* NC^0*.*

Because any NC^0 function is computable by a degree $d = O(1)$ polynomial, using these components we obtain that every $F_{h,s} \in \mathcal{F}$ is computable by a degree-d polynomial. Then, using the results of Alon et al. [AKK+03], there is a a poly-time adversary making $2^{O(d)}$ queries that has distinguishing advantage $\Omega(1)$. □

We also mention that such a construction can be broken by essentially the same argument even when f is a linear-stretch PRG (a stronger primitive than OWF), using the NC^0 construction of such PRG due to [AIK08] which is secure under the (somewhat non-standard) assumption of Alekhnovich [Ale03].

Finally, we briefly consider a more general type of construction than the above. These are constructions realized by poly-size bounded-depth circuits with parity gates. Reasoning as above, one can infer that if there are one-way functions computable in logarithmic space then such a construction yields a PRF computable by bounded-depth circuits with parity gates. Such PRF can be broken in quasi-polynomial time [RR97, KL01], while a polynomial-time distinguisher is unlikely to exist [Vio11].

4 Fully black-box stretch-increasing constructions

In this section we prove Theorem 1.4. The key property we require of our one-bit-stretch oracle G, stated in the next theorem, is that it reveals a large portion of its input, i.e. most of the output bits are simply copied from the input.

Theorem 4.1. *Let $\ell, d \in \mathbb{N}$ be sufficiently large with $d \leq \ell/2$. Then, for any subset $T \subseteq [\ell+1]$ with $|T| = \ell - d$ and any oracle A, there exists a generator $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ such that*

1. *G is $(2^{d/30}, 2^{-d/30})$-pseudorandom against adversaries with oracle access to A (and G).*
2. *For every input $x \in \{0,1\}^\ell$, $G(x)|_T = x_1 x_2 \cdots x_{\ell-d}$.*

We defer the proof of this theorem to Section 5, and instead start by showing how it is used to prove Theorem 1.4. First, we need a simple technical lemma showing that for any stretch-increasing construction of the specified form, we can find a large set of indices inside which most $b_i(x)$ fall for most choices of x.

Lemma 4.2. *Let $n, d, s, \ell \in \mathbb{N}$ with $d < \ell$. Let $\{b_i : \{0,1\}^n \to [\ell+1]\}_{i \in [n+s]}$ be a collection of $n+s$ functions. Then, there exists a set $T \subseteq [\ell+1]$ of size $\ell - d$ such that*

$$\Pr_x \left[|\{i : b_i(x) \in T\}| \geq (n+s) \cdot \left(1 - \frac{4(d+1)}{\ell+1}\right)\right] \geq \frac{3}{4}.$$

Proof. Let $S \subseteq [\ell+1]$ denote a random subset of size $d+1$. We have $\Pr_{x,i,S}[b_i(x) \in S] = (d+1)/(\ell+1)$, and so we can fix some S so that $\Pr_{x,i}[b_i(x) \in S] \leq (d+1)/(\ell+1)$. This can be restated as $\mathbb{E}_x\left[\Pr_i\left[b_i(x) \in S\right]\right] \leq (d+1)/(\ell+1)$, and so by Markov's inequality we have $\Pr_x[\Pr_i[b_i(x) \in S] \geq 4(d+1)/(\ell+1)] \leq 1/4$. Letting $T := [\ell+1] \setminus S$ completes the proof. □

We now prove Theorem 1.4.

Theorem 1.4. *For all sufficiently large ℓ and for $n \leq 2^{\sqrt{\ell}}$, there is no fully black-box construction $H^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n+s}$ of a generator with stretch $s \geq 5n/\log n$ and error $\epsilon \leq 1/4$ from any one-bit-stretch generator $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ with error $\delta \geq 2^{-\sqrt{\ell}/30}$ and with security reduction size $t \leq 2^{\sqrt{\ell}/30}$ of the form*

$$H^G(x) := G(q_1(x))_{b_1(x)} \circ \cdots \circ G(q_{n+s}(x))_{b_{n+s}(x)}$$

where $q_i : \{0,1\}^n \to \{0,1\}^\ell$ specifies the i-th query and $b_i : \{0,1\}^n \to [\ell+1]$ specifies the bit of the i-th answer to output.

Proof. Let $H^{(\cdot)}$ be a construction of the specified form. Fix a parameter $d := \ell/\log n$. Fix $T \subseteq [\ell+1]$ to be the subset of size $\ell - d$ guaranteed by Lemma 4.2. For each $x \in \{0,1\}^n$, let I_x denote the set $\{i : b_i(x) \in T\} \subseteq [n+s]$. Using $s = 5n/\log n$, the chosen value for d, and the fact that $|I_x|$ is an integer, the bound from Lemma 4.2 can be restated as $\Pr_x\left[|I_x| \geq n+1\right] \geq 3/4$ for sufficiently large n and ℓ. In the remainder of the proof, we refer to x such that $|I_x| \geq n+1$ as *good*.

Let T^{-1} denote a transformation such that $T^{-1}(j) = k$ if j is the kth smallest element of T (this is simply to provide a mapping from G's output bits to the corresponding revealed input bits). The adversary $A : \{0,1\}^{n+s} \to \{0,1\}$ is defined as the function which accepts exactly the set

$$\{z : \exists x \in \{0,1\}^n \text{ such that } x \text{ is good and } \forall i \in I_x, z_i = q_i(x)_{T^{-1}(b_i(x))}\}.$$

Let $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ be the PRG guaranteed by Theorem 4.1 using these choices of T and A. We claim that A distinguishes $H^G(U_n)$ from U_{n+s} with advantage at least $1/4$. To see this, consider z which is a uniformly chosen output of H^G, i.e. $z = H^G(x)$ for $x \leftarrow U_n$.

Because x is good with probability at least $3/4$, and because $H^G(x)_i = q_i(x)_{T^{-1}(b_i(x))}$ for all $i \in I_x$ by item 2 of Theorem 4.1, we have $\Pr[A(H^G(U_n)) = 1] \geq 3/4$. Conversely, for the case where A's input is chosen from U_{n+s}, we have the following calculation:

$$\begin{aligned}
\Pr_{z \leftarrow U_{n+s}}[A(z) = 1] &= \Pr_z\left[\exists x : \ x \text{ is good} \wedge \forall i \in I_x : z_i = q_i(x)_{T^{-1}(b_i(x))}\right] \\
&\leq \sum_{\substack{x \in \{0,1\}^n \\ x \text{ is good}}} \Pr_z\left[\forall i \in I_x : z_i = q_i(x)_{T^{-1}(b_i(x))}\right] \\
&\leq \sum_{\substack{x \in \{0,1\}^n \\ x \text{ is good}}} 2^{-(n+1)} \\
&\leq \frac{1}{2}.
\end{aligned}$$

(The second inequality follows from the fact that $|I_x| \geq n+1$ for x that are good.)

Finally, note that item 1 in Theorem 4.1 (along with the choice of d and the upper bound on n) implies that there is no oracle circuit C of size at most $2^{\sqrt{\ell}/30}$ such that $C^{A,G}$ distinguishes G with advantage at least $2^{-\sqrt{\ell}/30}$. Therefore, H does not meet the conditions of Definition 2.1 for the stated parameters. □

Next, we show that this theorem can be extended to the primitive black-box setting.

Theorem 4.3. *Let $n = n(\ell) \leq 2^{\sqrt{\ell}}$ and $s = s(n) \geq 5n/\log n$. Let $H^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n+s}$ be a primitive black-box stretch-increasing construction with stretch s from any family of one-bit-stretch generators $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$. If H has the form*

$$H^G(x) := G(q_1(x))_{b_1(x)} \circ \cdots \circ G(q_{n+s}(x))_{b_{n+s}(x)}$$

and the q_i and b_i are computable by poly(n)*-sized circuits, then NP/poly $\neq$ P/poly.*

Proof. Let H be a primitive black-box stretch-increasing construction of the specified form. Let G and I_x be defined as in Theorem 1.4 (the oracle A against which G is secure is not relevant here). Because the q_i, b_i functions are computable by poly(n)-size circuits, there is a poly(n)-size circuit family which computes the string $H^G(x)|_{I_x}$ on input x, while making *no* oracle calls to G. As a result, we can define a non-deterministic poly(n)-size circuit family which distinguishes H^G from uniform with advantage $1/4$: on input $z \in \{0,1\}^{n+s}$, the circuit non-deterministically guesses $x \in \{0,1\}^n$, and accepts iff $|I_x| \geq n+1$ and $z|_{I_x} = H^G(x)|_{I_x}$. The proof that this is indeed a distinguisher for H^G is identical to the argument given for Theorem 1.4.

Now assume for contradiction that NP/poly = P/poly, i.e. that every non-deterministic circuit family can be simulated by a deterministic circuit family with only a polynomial increase in size. Then, there is a poly(n)-size deterministic circuit family that distinguishes H^G from uniform with noticeable advantage. By the definition of a primitive black-box construction, there must also be such a circuit family that distinguishes G, contradicting G's pseudorandomness. □

5 Constructing the oracle generator

In this section we prove Theorem 4.1 (restated for convenience), which gives the one-bit-stretch oracle generator used in the proofs of our negative results (Theorems 1.4 and 4.3).

Theorem 4.1. *Let $\ell, d \in \mathbb{N}$ be sufficiently large with $d \leq \ell/2$. Then, for any subset $T \subseteq [\ell+1]$ with $|T| = \ell - d$ and any oracle A, there exists a generator $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ such that*

1. *G is $(2^{d/30}, 2^{-d/30})$-pseudorandom against adversaries with oracle access to A (and G).*
2. *For every input $x \in \{0,1\}^\ell$, $G(x)|_T = x_1 x_2 \cdots x_{\ell-d}$.*

On constructing the oracle. A direct proof that a random function $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ is a pseudorandom generator even for circuits that have oracle access to G does not seem immediate to us. The existence of such oracles is shown via an indirect route in an unpublished manuscript of Impagliazzo [Imp96] and – in a slightly different scenario – in a work by Zimand [Zim98]. Both works proceed by considering an oracle one-way function, and then applying standard constructions of generators from one-way functions (for which one can now use [HILL99] or [HRV10]).

We proceed by first considering a hard-to-invert oracle permutation π, and then using the Goldreich-Levin hardcore bit [GL89] to get one bit of stretch. This approach will have security exponential in the input length of π, and so we can apply π to the relatively few $(\Theta(\ell/\log \ell))$ bits outside of $|T|$, and then use padding to get a generator G on ℓ bits that reveals most of its input

We know of two ways to demonstrate the existence of such a permutation π. One is via a theorem in [GGKT05] which uses a clever encoding argument to prove that a random permutation is hard to invert with very high probability. They show that if there exists a small circuit which inverts a permutation π on some fraction of inputs, then π can be succinctly encoded when the circuit is given as advice. Then, since only a small number of permutations have succinct encodings, the probability that a random π can be sufficiently inverted by a fixed circuit is small, and a union bound over circuits gives the result.

The second way, and the one that we use here, is an arguably more direct argument showing that any fixed circuit with access to a fixed auxiliary oracle has negligible probability (over the choice of permutation) of sufficiently inverting the permutation. This method is from [Imp96] and [Zim98] (though they consider general length-preserving functions rather than permutations), and hinges on a combinatorial trick which originally appeared in [GKL93]. Briefly, it is shown that for a fixed circuit C, the expected number of subsets of size k that are inverted by C is not too large. Then, Markov's inequality is used to show that the probability that C inverts *any* set of size $m \approx k^2$ is small, since to do so C would have to invert *each* of its $\binom{m}{k}$ subsets of size k (this is the combinatorial trick).

We now turn to the formal proof of Theorem 4.1. There are two main ingredients; the first is the well-known Goldreich-Levin hard-core bit theorem [GL89]. It can be checked that the standard proof of this theorem relativizes; we omit the details.

Theorem 5.1. *Let $f : \{0,1\}^d \to \{0,1\}^m$ be a function, and let A be any oracle. Let C be an oracle circuit of size T such that $\Pr[C^A(f(U_d), U'_d) = \langle U_d, U'_d \rangle] \geq 1/2 + \epsilon$. Then, for d sufficiently large, there exists an oracle circuit B of size at most $\alpha \cdot T \cdot (d/\epsilon)^2$ (where α is a universal constant) such that $\Pr[B^A(f(U_d)) = U_d] \geq \epsilon^3/8d$.*

The second ingredient is the fact that there exist permutations π which are hard to invert even for adversaries that have access to π and to an arbitrary fixed auxiliary oracle.

Theorem 5.2. *Let $d \in \mathbb{N}$ be sufficiently large. Then for any oracle A, there exists a permutation $\pi : \{0,1\}^d \to \{0,1\}^d$ that is $(2^{d/5}, 2^{-d/5})$-hard to invert against adversaries with oracle access to π and A.*

Before giving the proof, we state and prove two lemmas. The aforementioned combinatorial trick, due to [GKL93], is given by the following lemma.

Lemma 5.3. *Let U be a finite set, let $\Gamma = \{\phi : U \to \{0,1\}\}$ be a family of predicates on U, and let p_k be an upper bound on the probability that ϕ chosen uniformly from Γ returns true for every element in a subset of size k, i.e.*

$$\forall K \subseteq U, |K| = k : \Pr_{\phi \leftarrow \Gamma}\left[\prod_{x \in K} \phi(x) = 1\right] \leq p_k.$$

Then, for any m such that $k \leq m \leq |U|$, we have

$$\Pr_{\phi \leftarrow \Gamma}\left[\exists M \subseteq U, |M| \geq m : \prod_{x \in M} \phi(x) = 1\right] \leq \frac{\binom{|U|}{k} \cdot p_k}{\binom{m}{k}}.$$

Proof. Let $\phi(X)$ denote $\prod_{x \in X} \phi(x)$. We have $\mathbb{E}\left[|\{K \subseteq U : |K| = k \text{ and } \phi(K) = 1\}|\right] \leq \binom{|U|}{k} \cdot p_k$ by linearity of expectation. Then the lemma follows from double counting, because for any set $M \subseteq U$ of size m, $\phi(M) = 1$ iff $\phi(K) = 1$ for every one of the $\binom{m}{k}$ subsets $K \subseteq M$ of size k. □

We now explain why this lemma is helpful. Following [Imp96] and [Zim98], we bound the probability (over the permutation π) that a fixed circuit C of size s inverts a fixed set K of size k; this is done by considering the probability that any k out of the at most ks distinct queries made by C on inputs from K are mapped by π to K; specifically, we bound

$$p_k \leq \binom{ks}{k} \cdot \left(\frac{k}{|U|}\right)^k \approx \frac{s^k}{\binom{|U|}{k}}.$$

The factor of s^k means that we cannot use a union bound over all $\binom{|U|}{k}$ subsets of size k. So we instead use Lemma 5.3, choosing m so that $\binom{m}{k} \approx s^{2.3k}$, which makes the probability of inverting a set of size m small enough to use a union bound over all circuits.

We also require a bound on the number of oracle circuits of a given size.

Lemma 5.4. *There are at most $2^{s(3+4\log s)}$ oracle circuits of size s which have access to two oracles π and A.*

Proof. We define the size of a circuit to be the number of wires it has; this is also an upper bound on the number of gates. For each wire in the circuit, we must specify two things:

- which gate it is an output of (or if it is an input wire) and which position it is in for this gate
- which gate it is an input of (or if it is an output wire) and which position it is in for this gate

Note that the positions are relevant for wires incident on oracle gates, as the functions computed by these gates may not be symmetric. Specifying either incident gate for a given wire takes $\log s$ bits (as there are at most s gates), and likewise each position can be specified with $\log s$ bits. Therefore, each of the s wires can be specified with $4\log s$ bits. Finally, for each gate, we must specify which of the five types it is ($\wedge, \vee, \neg, \pi$-oracle or A-oracle), which takes three bits. □

Proof of Theorem 5.2. We will in fact show that a random π has the desired property with probability at least $1-2^{-2^{d/4}}$. Fix an oracle A and an oracle circuit C of size s. Fix a subset $K \subseteq \{0,1\}^d$ of size k; we will first bound the probability that C inverts all of K. Let Q_x^π denote the set of at most s distinct queries that $C^{A,\pi}(x)$ makes to π (for some choice of x and π), and let $Q_K^\pi := \bigcup_{x\in K} Q_x^\pi$. We assume without loss of generality that the last query that C makes to π is the string that C outputs (this is justified because any circuit which does not query its output string can be modified into one that does with an increase in size that is so small as to not affect the union bound below).

A necessary condition for C to invert all of K is that $\pi^{-1}(x) \in Q_K^\pi$ for all $x \in K$. Since $|Q_K^\pi| \le ks$, we can bound this by

$$\begin{aligned}
\Pr_\pi\left[\forall x \in K : \pi^{-1}(x) \in Q_K^\pi\right] &\le \Pr_\pi\left[\exists X \subseteq Q_K^\pi : \bigcup_{x\in X} \pi(x) = K\right] \\
&\le \binom{ks}{k} \cdot \left(\frac{k}{2^d}\right)\left(\frac{k-1}{2^d-1}\right)\cdots\left(\frac{1}{2^d-k+1}\right) \\
&\le \left(\frac{eks}{2^d}\right)^k.
\end{aligned}$$

We now apply Lemma 5.3 in the obvious way: U is $\{0,1\}^d$, and there is a predicate $\phi_\pi \in \Gamma$ for each permutation π, where $\phi_\pi(x) = 1$ iff $C^{A,\pi}(x) = \pi^{-1}(x)$. By the lemma, the probability that there exists a set M of size $m \ge k$ such that C inverts every element of M is bounded from above by $(e^2 \cdot k \cdot s/m)^k$. Choosing $k = 2^{d/3}, m = 2^{4d/5}$ and $s = 2^{d/5}$, this is bounded by $2^{-2^{d/3}}$ for sufficiently large d. By Lemma 5.4, there are at most $2^{2^{d/5}\cdot\Theta(d)}$ circuits of size $2^{d/5}$, and so the probability over the choice of π that there *exists* a circuit of size $2^{d/5}$ which inverts a set of size at least $2^{4d/5}$ is at most $2^{-2^{d/3}+2^{d/5}\cdot\Theta(d)} < 2^{-2^{d/4}}$ for sufficiently large d. Therefore, π is $(2^{d/5}, 2^{-d/5})$-hard to invert with probability at least $1-2^{-2^{d/4}}$. □

We may now give the proof of Theorem 4.1.

Proof of Theorem 4.1. Let the oracle A and the subset T be given. Recall that $|T| = \ell - d$, and let $\pi : \{0,1\}^d \to \{0,1\}^d$ be the permutation guaranteed by Theorem 5.2 which is $(2^{d/5}, 2^{-d/5})$-hard to invert against adversaries with oracle access to π and A. Then, the generator G treats its input $x \in \{0,1\}^\ell$ as $(x_1, x_2, x_3) \in \{0,1\}^{\ell-2d} \times \{0,1\}^d \times \{0,1\}^d$, and outputs the $(\ell+1)$-bit string defined as follows:

$$G(x)|_{[\ell+1]\setminus T} = \pi(x_3) \circ \langle x_3, x_2 \rangle \qquad\qquad G(x)|_T = x_1 \circ x_2.$$

Now assume for contradiction that there exists an oracle circuit $C : \{0,1\}^{\ell+1} \to \{0,1\}$ of size at most $2^{d/30}$ such that $\Pr[C^{A,G}(G(U_\ell)) = 1] - \Pr[C^{A,G}(U_{\ell+1}) = 1] \geq 2^{-d/30}$ (dropping the absolute value w.l.o.g.). Because the permutation π is the only part of G's output which may be "difficult" to compute, we can take C to have oracles (A, π) instead of (A, G) at the cost of increasing C's size by a factor of $\text{poly}(d)$. We construct a probabilistic oracle circuit $IP : \{0,1\}^d \times \{0,1\}^d \to \{0,1\}$ which, on input (x, y), tries to compute $\langle \pi^{-1}(x), y \rangle$. $IP^{A,\pi}(x,y)$ performs the following steps:

1. chooses a random string $z \in \{0,1\}^{\ell-2d}$ and a random bit $b \in \{0,1\}$
2. constructs the $(\ell+1)$-bit string w defined by $w|_{[\ell+1]\setminus T} = x \circ b,\ w|_T = z \circ y$
3. computes $C^{A,\pi}(w)$ and outputs $C^{A,\pi}(w) \oplus 1 \oplus b$

We clearly have $|IP| \leq |C| \cdot \text{poly}(d) \leq 2^{d/30} \cdot \text{poly}(d)$. Consider the behavior of $IP^{A,\pi}$ on a uniformly random input (x, y). It is easy to see that the string w is distributed according to $U_{\ell+1}$. If we condition on the chosen bit b being equal to $\langle \pi^{-1}(x), y \rangle$ (which happens with probability $1/2$), then w is distributed according to $G(U_\ell)$. For brevity, let E_{IP} denote the event $IP^{A,\pi}(x,y) = \langle \pi^{-1}(x), y \rangle$, and let E_b denote the event $b = \langle \pi^{-1}(x), y \rangle$. Then,

$$\begin{aligned}
\Pr[E_{IP}] &= \frac{1}{2}\left(\Pr[E_{IP} \mid E_b] + \Pr[E_{IP} \mid \overline{E_b}]\right) \\
&= \frac{1}{2}\left(\Pr[C^{A,\pi}(w) = 1 \mid E_b] + \left(1 - \Pr[C^{A,\pi}(w) = 1 \mid \overline{E_b}]\right)\right) \\
&= 1/2 + \Pr[C^{A,\pi}(w) = 1 \mid E_b] - \Pr[C^{A,\pi}(w) = 1] \\
&= 1/2 + \Pr[C^{A,\pi}(G(U_\ell)) = 1] - \Pr[C^{A,\pi}(U_{\ell+1}) = 1] \\
&\geq 1/2 + 2^{-d/30}.
\end{aligned}$$

The probabilities are over both (x, y) and the internal randomness of IP; by a standard averaging argument, we can fix the internal randomness of IP to get a deterministic circuit which computes $\langle \pi^{-1}(x), y \rangle$ on a random (x, y) with the same success probability. Then for sufficiently large d, Theorem 5.1 gives an oracle circuit of size at most $2^{d/30} \cdot \text{poly}(d) \cdot O(d^2 \cdot 2^{2d/30}) \leq 2^{d/5}$ that, when given access to A and π, inverts π with probability at least $2^{-3d/30}/8d \geq 2^{-d/5}$ over its input, contradicting the hardness of π. □

Acknowledgements. We are very grateful to Benny Applebaum for several useful comments, and especially for pointing out the strengthening of Theorem 1.4 and allowing us to include a proof in §1.2. We also would like to thank Russell Impagliazzo for sharing [Imp96] with us, and the anonymous TCC referees for helpful feedback.

References

[ABI86] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized algorithm for the maximal independent set problem. *Journal of Algorithms*, 7:567–583, 1986.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. *SIAM J. Comput.*, 36(4):845–888, 2006.

[AIK08] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in NC^0. *Computational Complexity*, 17(1):38–69, 2008.

[AKK+03] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing low-degree polynomials over GF(2). In *7th Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, volume 2764 of *Lecture Notes in Computer Science*, pages 188–199. Springer, 2003.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307, 2003.

[BH12] Itay Berman and Iftach Haitner. From non-adaptive to adaptive pseudorandom functions. In *9th Theory of Cryptography Conference (TCC)*, 2012.

[BJP11] Josh Bronson, Ali Juma, and Periklis A. Papakonstantinou. Limits on the stretch of non-adaptive constructions of pseudo-random generators. In *8th Theory of Cryptography Conference (TCC)*, 2011.

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. on Computing*, 13(4):850–864, November 1984.

[CG89] Benny Chor and Oded Goldreich. On the power of two-point based sampling. *Journal of Complexity*, 5(1):96–106, 1989.

[CW79] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. System Sci.*, 18(2):143–154, 1979.

[GGKT05] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. of the ACM*, 33(4):792–807, October 1986.

[GKL93] Oded Goldreich, Hugo Krawczyk, and Michael Luby. On the existence of pseudorandom generators. *SIAM J. Comput.*, 22(6):1163–1175, 1993.

[GL89] Oded Goldreich and Leonid Levin. A hard-core predicate for all one-way functions. In *21st ACM Symp. on the Theory of Computing (STOC)*, pages 25–32, 1989.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.

[HHR06] Iftach Haitner, Danny Harnik, and Omer Reingold. Efficient pseudorandom generators from exponentially hard one-way functions. In *Coll. on Automata, Languages and Programming (ICALP)*, pages 228–239, 2006.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[Hol06] Thomas Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2006.

[HRV10] Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *42nd ACM ACM Symp. on the Theory of Computing (STOC)*, pages 437–446, 2010.

[Imp96] Russell Impagliazzo. Very strong one-way functions and pseudo-random generators exist relative to a random oracle. Manuscript, 1996.

[KL01] Matthias Krause and Stefan Lucks. On the minimal hardware complexity of pseudorandom function generators. In *Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 419–430, 2001.

[Lu06] Chi-Jen Lu. On the complexity of parallel hardness amplification for one-way functions. In *3rd Theory of Cryptography Conference (TCC)*, pages 462–481, 2006.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. of Computer and System Sciences*, 49(2):149–167, 1994.

[RR97] Alexander Razborov and Steven Rudich. Natural proofs. *J. of Computer and System Sciences*, 55(1):24–35, August 1997.

[RTV04] Omer Reingold, Luca Trevisan, and Salil Vadhan. Notions of reducibility between cryptographic primitives. In *1st Theory of Cryptography Conference (Feb 19-21, 2004: Cambridge, MA, USA)*. Springer-Verlag, 2004.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *J. of the ACM*, 48(4):860–879, 2001.

[Vio05] Emanuele Viola. On constructing parallel pseudorandom generators from one-way functions. In *20th IEEE Conf. on Computational Complexity (CCC)*, pages 183–197, 2005.

[Vio11] Emanuele Viola. The communication complexity of addition. 2011.

[VZ12] Salil P. Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In *ACM Symp. on the Theory of Computing (STOC)*, 2012.

[Yao82] Andrew Yao. Theory and applications of trapdoor functions. In *23rd IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 80–91. IEEE, 1982.

[Zim98] Marius Zimand. Efficient privatization of random bits. In *"Randomized Algorithms" satellite workshop of the 23rd Symposium on Mathematical Foundations of Computer Science*, 1998. Available at http://triton.towson.edu/∼mzimand/pub/rand-privat.ps.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de