A Dichotomy for Local Small-Bias Generators

Benny Applebaum* Andrej Bogdanov† Alon Rosen‡

February 26, 2015

Abstract

We consider pseudorandom generators in which each output bit depends on a constant number of input bits. Such generators have appealingly simple structure: they can be described by a sparse input-output dependency graph G and a small predicate P that is applied at each output. Following the works of Cryan and Miltersen (MFCS'01) and by Mossel et al (STOC'03), we ask: which graphs and predicates yield "small-bias" generators (that fool linear distinguishers)?

We identify an explicit class of degenerate predicates and prove the following. For most graphs, all *non-degenerate* predicates yield small-bias generators, $f\colon \{0,1\}^n \to \{0,1\}^m$, with output length $m = n^{1+\epsilon}$ for some constant $\epsilon > 0$. Conversely, we show that for most graphs, *degenerate* predicates are not secure against linear distinguishers, even when the output length is linear $m = n + \Omega(n)$. Taken together, these results expose a dichotomy: every predicate is either very hard or very easy, in the sense that it either yields a small-bias generator for almost all graphs or fails to do so for almost all graphs.

As a secondary contribution, we attempt to support the view that small-bias is a good measure of pseudorandomness for local functions with large stretch. We do so by demonstrating that resilience to linear distinguishers implies resilience to a larger class of attacks.

Keywords: small-bias generator, dichotomy, local functions, NC0.

*School of Electrical Engineering at Tel-Aviv University, Email: benny.applebaum@gmail.com. Supported by Alon Fellowship, ISF grant 1155/11, Israel Ministry of Science and Technology (grant 3-9094), GIF grant 1152/2011, the Check Point Institute for Information Security, and by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC.

†Department of Computer Science and Engineering and Institute for Theoretical Computer Science and Communications, Chinese University of Hong Kong. Email: andrejb@cse.cuhk.edu.hk. Supported in part by Hong Kong RGC GRF grant CUHK 410309.

‡Efi Arazi School of Computer Science, IDC Herzliya. Email: alon.rosen@idc.ac.il. Supported by the Israel Science Foundation (grant No. 334/08).

1 Introduction

In recent years there has been interest in the study of cryptographic primitives that are implemented by *local* functions, that is functions in which each output bit depends on a constant number of input bits. This study has been in large part spurred by the discovery that, under widely accepted cryptographic assumptions, local functions can achieve rich forms of cryptographic functionality, ranging from one-wayness and pseudorandom generation to semantic security and existential unforgeability [6].

Local functions have simple structure: they can be described by a sparse input-output dependency graph and a sequence of small predicates applied at each output. Besides allowing efficient parallel evaluation, this simple structure makes local functions amenable to analysis, and gives hope for proving highly non-trivial statements about them. Given that the cryptographic functionalities that local functions can achieve are quite complex, it is very interesting and appealing to try to understand which properties of local functions (namely, graphs and predicates) are necessary and sufficient for them to implement such functionalities.

In this work we focus on the study of local pseudorandom generators with large stretch. We give evidence that for most graphs, all but a handful of "degenerate" predicates yield pseudorandom generators with output length $m = n^{1+\varepsilon}$ for some constant $\varepsilon > 0$. Conversely, we show that for almost all graphs, degenerate predicates are not secure even against linear distinguishers. Taken together, these results expose a dichotomy: every predicate is either very hard or very easy, in the sense that it either yields a generator for almost all graphs or fails to do so for almost all graphs.

1.1 Easy, sometimes hard, and almost always hard predicates

Recall that a pseudorandom generator is a length increasing function $f\colon \{0,1\}^n \to \{0,1\}^m$ such that no efficiently computable test can distinguish with noticeable advantage between the value $f(x)$ and a randomly chosen $y \in \{0,1\}^m$, when $x \in \{0,1\}^n$ is chosen at random. The additive *stretch* of f is defined to be the difference between its output length m and its input length n.

In the context of constructing local pseudorandom generators of superlinear stretch, we may assume without loss of generality that all outputs apply the same predicate $P\colon \{0,1\}^d \to \{0,1\}$.[1] We are interested in understanding which d-local functions $f_{G,P}\colon \{0,1\}^n \to \{0,1\}^m$, described by a graph G and a predicate P, are pseudorandom generators. For a predicate P, we will say

- P is *easy* if $f_{G,P}$ is *not* pseudorandom (for a given class of adversaries) for every G,
- P is *sometimes hard* if $f_{G,P}$ is pseudorandom for some G, and
- P is *almost always hard* if $f_{G,P}$ is pseudorandom for a $1 - o(1)$ fraction of G.[2]

Cryan and Miltersen [17] and Mossel et al. [28] identified several classes of predicates that are easy for polynomial time algorithms when the stretch is a sufficiently large linear function. These include (1) unbalanced predicates, (2) linear predicates, (3) predicates that are biased towards one

[1] If this is not the case, project on the outputs labeled by the most frequent predicate.

[2] One cannot hope for *always* hard predicates, for which $f_{G,P}$ is pseudorandom for *all* graphs, as some *easy* graphs fatally fail to provide pseudorandomness. This is the case, for example, when the graph connects two outputs to the same d inputs. In fact, an inverse polynomial fraction of all dependency graphs (with m outputs, n inputs, and degree d) are easy.

input (i.e., $\Pr_w[P(w) = 1] \neq \frac{1}{2}$), and (4) predicates that are biased towards a pair of inputs (i.e., $\Pr_w[P(w) = w_i \oplus w_j] \neq \frac{1}{2}$). We call such predicates *degenerate*. By a case-analysis, it can be showed that degenerate predicates include all predicates of locality at most four [17, 28].

On the positive side, Mossel et al. [28] also gave examples of five-bit predicates that are sometimes (exponentially) hard against linear distinguishers. Applebaum et al. [5] show that when the locality is sufficiently large, almost always hard predicates against linear distinguishers exist.

Pseudorandomness against linear distinguishers means that there is no subset of output bits whose XOR has noticeable bias. This notion, due to Naor and Naor [29], was advocated in the context of local pseudorandom generators by Cryan and Miltersen [17]. A bit more formally, for a function $f : \{0,1\}^n \to \{0,1\}^m$, we let

$$\mathsf{bias}(f) = \max_L \left|\Pr[L(f(\mathcal{U}_n)) = 1] - \Pr[L(\mathcal{U}_m) = 1]\right|,$$

where the maximum is taken over all affine functions $L : \mathbb{F}_2^m \to \mathbb{F}_2$. A small-bias generator is a function f for which $\mathsf{bias}(f)$ is small (preferrably negligible) as a function of n.

1.2 Our Results

We fully classify predicates by showing that all predicates that are not known to be *easy*, are *almost always hard*.

Theorem 1.1 (**Non-degenerate predicates are hard).** *Let $P : \{0,1\}^d \to \{0,1\}$ be any non-degenerate predicate. Then, for every $\varepsilon < 1/4$ and $m = n^{1+\varepsilon}$:*

$$\Pr_G[\mathsf{bias}(f_{G,P}) \leq \delta(n)] > 1 - o(1),$$

where $\delta(n) = \exp(-\Omega(n^{1/4-\varepsilon}))$ and G is randomly chosen from all d-regular hypergraphs with n nodes (representing the inputs) and m hyperedges (representing the outputs).

The theorem shows that, even when locality is large, the only easy predicates are degenerate ones, and there are no other "sources of easiness" other than ones that already appear in predicates of locality 4 or less.

Conversely, we show that degenerate predicates are easy for *linear* distinguishers (as opposed to general polynomial-time distinguishers).

Theorem 1.2 (**Linear tests break degenerate predicates).** *For every $m = n + \Omega(n)$, and every degenerate predicate $P : \{0,1\}^d \to \{0,1\}$*

$$\Pr_G[\mathsf{bias}(f_{G,P}) > 1/\mathrm{poly}(n)] > 1 - o(1),$$

where G is randomly chosen from all d-regular hypergraphs with n nodes and m hyperedges.

The proof of Thm. 1.2 mainly deals with degenerate predicates that are correlated with a pair of their inputs; In this case, we show that the non-linear distinguisher which was previously used in [28] and was based on a semi-definite program for MAX-2-LIN [21] can be replaced with a simple linear distinguisher. (The proof for other degenerate predicates follows from previous works).

Taken together, Theorems 1.1 and 1.2 expose a dichotomy: a predicate can be either easy (fail for almost all graphs) or hard (succeeds for almost all graphs). One possible interpretation of our

results is that, from a designer point of view, a strong emphasis should be put on the choice of the predicate, while the choice of the input-output dependency graph may be less crucial (since if the predicate is appropriately chosen then most graphs yield a small-bias generator). In some sense, this means that constructions of local pseudorandom generators with large stretch are robust: as long as the graph G is "typical," any non-degenerate predicate can be used (our proof classifies explicitly what is a typical family of graphs and in addition shows that even a mixture of different non-degenerate predicates would work).

1.3 Why Polynomial Stretch?

While Applebaum et al. [6] give strong evidence that local pseudorandom generators exist, the stretch their construction achieves is only sublinear ($m = n + n^{1-\varepsilon}$). In contrast, the regime of large (polynomial or even linear) stretch is not as well understood, and the only known constructions are based on non-standard assumptions. (See Section 1.5.)

Large-stretch local generators are known to have several applications in cryptography and complexity, such as secure computation with constant overhead [25] and strong (average-case) inapproximability results for constraint-satisfaction problems [7]. These results are not known to follow from other (natural) assumptions. It should be mentioned that it is possible to convert small polynomial stretch of $m = n^{1+\varepsilon}$ into arbitrary (fixed) polynomial stretch of $m = n^c$ at the expense of constant blow-up in the locality. (This follows from standard techniques, see [4] for details). Hence, it suffices to focus on the case of $m = n^{1+\varepsilon}$ for some fixed ε.

The proof of Theorem 1.1 yields exponentially small bias when $m = O(n)$, and sub-exponential bias for $m = n^{1+\varepsilon}$ where $\varepsilon < 1/4$. We do not know whether this is tight, but it can be shown that some non-degenerate predicates become easy (to break on a random graph) when the output length is $m = n^2$ or even $m = n^{3/2}$. In general, it seems that when m grows the number of hard predicates of locality d decreases, till the point $m^\star$ where all predicates become easy. (By [28], $m^\star \leq n^{d/2}$.) It will be interesting to obtain a classification for larger output lengths, and to find out whether a similar dichotomy happens there as well.

1.4 Why Small-Bias?

Small-bias generators are a strict relaxation of cryptographic pseudorandom generators in that the tests $L : \mathbb{F}_2^m \to \mathbb{F}_2$ are restricted to be affine (as opposed to arbitrary efficiently computable functions). Even though affine functions are, in general, fairly weak distinguishers, handling them is a necessary first step towards achieving cryptographic pseudorandomness. In particular, affine functions are used extensively in cryptanalysis and security against them already rules out an extensive class of attacks.

For local pseudorandom generators with linear stretch, Cryan and Miltersen conjectured that affine distinguishers are as powerful as polynomial-time distinguishers [17]. In Section 5, we attempt to support this view by showing that resilience against small-bias, by itself, leads to robustness against other classes of attacks.

Small-bias generators are also motivated by their own right being used as building blocks in constructions that give stronger forms of pseudorandomness. This includes constructions of local cryptographic pseudorandom generators [7, 4], as well as pseudorandom generators that fool low-degree polynomials [14], small-space computations[24], read-once formulas[11].

1.5 Related Work

The function $f_{G,P}$ was introduced by Goldreich [22] who conjectured that when $m = n$, one-wayness should hold for a random graph and a random predicate. This view is supported by the results of [22, 30, 3, 16, 27, 20, 26] who show that a large class of algorithms (including ones that capture DPLL-based heuristics) fail to invert $f_{G,P}$ in polynomial-time.

At the linear regime, i.e., when $m = n + \Omega(n)$, it is shown in [12] that if the predicate is degenerate the function $f_{G,P}$ can be *inverted* in polynomial-time. (This strengthens the results of [17, 28] who only give distinguishers.) Recently, a strong self-amplification theorem was proved in [13] showing that for $m = n + \Omega_d(n)$ if $f_{G,P}$ is hard-to-invert over tiny (sub-exponential small) fraction of the inputs with respect to sub-exponential time algorithm, then the same function is actually hard-to-invert over almost all inputs (with respect to sub-exponential time algorithms).

Pseudorandom generators with *sub-linear* stretch can be implemented by 4-local functions based on standard intractability assumptions (e.g., hardness of factoring, discrete-log, or lattice problems) [6], or even 3-local functions based on the intractability of decoding random linear codes [8]. However, it is unknown how to extend this result to polynomial or even linear stretch since all known stretch amplification procedures introduce a large (polynomial) overhead in the locality. In fact, for the special case of 4-local functions (in which each output depends on at most 4 input bits), there is a provable separation: Although such functions can compute sub-linear pseudorandom generators [6] they *cannot* achieve polynomial-stretch [17, 28].

Alekhnovich [1] conjectured that for $m = n + \Theta(n)$, the function $f_{G,P}$ is pseudorandom for a random graph and when P is a *randomized* predicate which computes $z_1 \oplus z_2 \oplus z_3$ and with some small probability $p < \frac{1}{2}$ flips the result. Although this construction does not lead directly to a local function (due to the use of noise), it was shown in [7] that it can be derandomized and transformed into a local construction with linear stretch. (The restriction to linear stretch holds even if one strengthen Alekhnovich's assumption to $m = \text{poly}(n)$.)

More recently, [4] showed that the pseudorandomness of $f_{G,P}$ with respect to a random graph and output length m, can be reduced to the one-wayness of $f_{H,P}$ with respect to a random graph H and related output length m'. The current paper complements this result as it provides a criteria for choosing the predicate P.[3]

2 Techniques and Ideas

In this section we give an overview of the proof of our Theorem 1.1. Let $f : \{0,1\}^n \to \{0,1\}^m$ be a d-local function where each output bit is computed by applying some d-local predicate $P : \{0,1\}^d \to \{0,1\}$ to a (ordered) subset of the inputs $S \subseteq [n]$.[4] Any such function can be described by a list of m d-tuples $G = (S_1, \ldots, S_m)$ and the predicate P. Under this convention, we let $f_{G,P} : \{0,1\}^n \to \{0,1\}^m$ denote the corresponding d-local function.

We view G as a d-regular hypergraph with n nodes (representing inputs) and m hyperedges

[3] The reduction of [4] has some overhead which leads only to weak (inverse-polynomial) security. This is fixed (without violating locality) in the case of linear stretch, but in the case of polynomial stretch it only yield inverse-polynomial security (for arbitrary fixed polynomial). Furthermore, in the polynomial regime, the predicate is required to be of the form $P(w) = w_1 \oplus P'(w_2, \ldots, w_d)$. Fortunately, such predicates can be classified as hard in our dichotomy.

[4] We can assume that the same predicate is being used for all outputs at the expense of shortening the output length by a constant factor (as there are only 2^{2^d} different predicates).

(representing outputs) each of size d. (We refer to such a graph as an (m, n, d)-graph.) Since we are mostly interested in polynomial stretch we think of m as $n^{1+\varepsilon}$ for some fixed $\varepsilon > 0$, e.g., $\varepsilon = 0.1$.

We would like to show that for almost all (m, n, d)-graphs G, the function $f_{G,P}$ fools all linear tests T, where P is non-degenerate. Following [28], we distinguish between *light* linear tests which depend on less than $k = \Omega(n^{1-2\varepsilon})$ outputs, and *heavy* tests which depend on more than k outputs.

Recall that a non-degenerate predicate satisfies two forms of "non-linearity": (1) (2-*resilient*) P is uncorrelated with any linear function that involves less than 3 variables; and (2) (*degree* 2) the algebraic degree of P as a polynomial over $\mathbb{F}_2$ is at least 2. Both properties are classical design criteria which are widely used in practical cryptanalysis (cf. [31]). It turns out that the first property allows to fool light tests and the second property fools heavy tests.

2.1 Fooling light tests

Our starting point is a result of [28] which shows that if the predicate is the parity predicate $\oplus$ and the graph is a good expander, the output of $f_{G,\oplus}(\mathcal{U}_n)$ *perfectly* fools all light linear tests. In terms of expectation, this can be written as

$$\mathop{\mathsf{E}}_{x}[L(f_{G,\oplus}(x)) = 0],$$

where we think of $\{0, 1\}$ as $\{\pm 1\}$, and let $L : \{\pm 1\}^m \to \{\pm 1\}$ be a light linear test. Our key insight is that the case of a general predicate P can be reduced to the case of linear predicates.

More precisely, let ξ denote the outcome of the test $L(f_{G,P}(x))$. Then, by looking at the Fourier expansion of the predicate P, we can write ξ as a convex combination over the reals of exponentially many summands of the form $\xi_i = L(f_{G_i,\oplus}(x))$ where the graphs G_i are subgraphs of G. (The exact structure of G_i is determined by the Fourier representation of P.) When x is uniformly chosen, the random variable ξ is a weighted sum (over the reals) of many dependent random variables ξ_i's. We show that if G has sufficiently high vertex expansion (every not too large set of hyperedges covers many vertices) then the expectation of each summand ξ_i is zero, and so, by the linearity of expectation, the expectation of ξ is also zero.

When the predicate is 2-resilient the size of each hyperedge of G_i is at least 3, and therefore if every 3-uniform subgraph of G is a good expander $f_{G,P}$ (perfectly) passes all light linear tests. Most graphs G satisfy this property. We emphasize that the argument crucially relies on the *perfect* bias of XOR predicates, as there are exponentially many summands. (See Section 3.1 for full details.)

2.2 Fooling heavy tests

Consider a heavy test which involves $t \geq k$ outputs. Switching back to zero-one notation, assume that the test outputs the value $\xi = P(x_{S_1}) + \ldots + P(x_{S_t}) \pmod 2$ where $x \stackrel{R}{\leftarrow} \mathcal{U}_n$. Our goal is to show that ξ is close to a fair coin. For this it suffices to show that the sum ξ can be rewritten as the sum (over $\mathbb{F}_2$) of ℓ random variables

$$\xi = \xi_1 + \ldots + \xi_\ell \pmod 2, \tag{1}$$

where each random variable ξ_i is an *independent* non-constant coin, i.e., $\Pr[\xi_i = 1] \in [2^{-d}, 1 - 2^{-d}]$. In this case, the statistical distance between ξ and a fair coin is exponentially small (in ℓ), and we are done as long as ℓ is large enough.

In order to partition ξ, let us look at the hyperedges $S_1, \ldots, S_t$ which are involved in the test. As a first attempt, let us collect ℓ distinct "independent" hyperedges that do not share a single common variable. Renaming the edges, we can write ξ as

$$(P(x_{T_1}) + \ldots + P(x_{T_\ell})) + \big(P(x_{S_{\ell+1}}) + \ldots + P(x_{S_t})\big) \pmod 2,$$

where the first ℓ random variables are indeed statistically independent. However, the last $t - \ell$ hyperedges violate statistical-independence as they may be correlated with more than one of the first ℓ hyperdges. This is the case, for example, if S_j has a non-empty intersection with both T_i and T_r. This problem is fixed by collecting ℓ "strongly-independent" hyperedges $T_1, \ldots, T_\ell$ for which every S_j intersects at most a single T_i. (Such a big set is likely to exist since t is sufficiently large.) In this case, for any fixing of the variables outside the T_i's, the random variable ξ can be partitioned into ℓ independent random variables of the form $\xi_i = P(x_{T_i}) + \sum P(x_{S_j})$, where the sum ranges over the S_j's which intersects T_i. This property (which is a relaxation of Eq. 1) still suffices to achieve our goal, as long as the ξ_i's are non-constant.

To prove the latter, we rely on the fact that P has algebraic degree 2. Specifically, let us assume that S_i and T_j have no more than a single common input node. (This condition can be typically met at the expense of throwing a small number of the T_i's.) In this case, the random variable $\xi_i = P(x_{T_i}) + \sum P(x_{S_j})$ cannot be constant, as the first summand is a degree 2 polynomial in x_{T_i} and each of the last summands contain at most a single variable from T_i. Hence, ξ_i is a non-trivial polynomial whose degree is lower-bounded by 2. This completes the argument. Interestingly, non-linearity is used only to prove that the ξ_i's are non-constant. Indeed, linear predicates fail exactly for large tests for which the ξ_i's become fixed due to local cancelations. (See Section 3.2 for details.)

2.3 Proving Theorem 1.2

When P is a degenerate predicate and G is random, the existence of a linear distinguisher follows by standard arguments. The cases of linear or biased P are trivial, and the case of bias towards one input was analyzed by Cryan and Miltersen. When P is biased towards a pair of inputs, say the first two, we think of P as an "approximation" of the parity $x_1 \oplus x_2$ of its first two inputs. If P happened to be the predicate $x_1 \oplus x_2$, one could find a short "cycle" of output bits that, when XORed together, causes the corresponding input bits to cancel out. In general, as long as the outputs along the cycle do not share any additional input bits, the output of the test will be biased, with bias exponential in the length of the cycle. In Section 4 we show that a random G is likely to have such short cycles, and so the corresponding linear test will be biased.

3 Non-Degenerate Predicates are Hard

In this section we prove Theorem 1.1. We follow the outline described in Section 2 and handle light linear tests and heavy linear tests separately.

3.1 Fooling Light Tests

In this section we show that if the predicate P is 2-resilient (see definition below) and the graph G is a good expander, the function $f_{G,P}$ is k-wise independent, and in particular fools linear tests of weight smaller than k. We will need the following definitions.

Lossless expansion. Let G be an (m, n, d)-graph. We will say G is (k, t)-expanding ($1 \leq k \leq m, 1 \leq t \leq d$) if for every $\ell \leq k$, every collection of $\ell \leq k$ distinct hyperedges of G covers more than $t\ell$ distinct vertices. We say G is (k, a)-*linear* ($1 \leq a \leq d$) if for every collection of k distinct hyperedges $S_1, \ldots, S_k$ and every collection of subsets $T_1 \subseteq S_1, \ldots, T_k \subseteq S_k$ where $|T_1|, \ldots, |T_k| \geq a$, the indicator vectors of $T_1, \ldots, T_k$ are linearly independent over $\mathbb{F}_2^n$.

Fourier coefficients. The Fourier expansion of a predicate $P : \{0,1\}^d \to \{\pm 1\}$ is given by $\sum_{T \subseteq [d]} \alpha_T \chi_T$ where $\chi_T(x_1, \ldots, x_d) = (-1)^{\sum_{t \in T} x_i}$ is Parity on the coordinates in the set T. The predicate is *a-resilient* if α_T is zero for every T of size smaller or equal to a.

The following lemma shows that resiliency combined with (k, a)-linearity leads to k-wise independence.

Lemma 3.1. *If P is $(a-1)$-resilient and the (m, n, d)-graph G is (k, a)-linear then $f_{G,P}$ is k-wise independent generator, i.e., the m r.v.'s $(y_1, \ldots, y_m) = f_{G,P}(\mathcal{U}_n)$ are k-wise independent.*

To prove the lemma, we will employ the following fact which follows from Vazirani's XOR lemma (cf. [23]).

Fact 3.2. *A sequence of ± 1 random variables $(y_1, \ldots, y_t)$ is k-wise independent if for every $\ell \leq k$ and every $i_1 < i_2 < \ldots < i_\ell$ it holds that $\mathsf{E}[y_{i_1} \cdot \cdots \cdot y_{i_\ell}] = 0$.*

We can now prove Lemma 3.1.

Proof of Lemma 3.1. We will use the following notation: For a hyperedge $S = (i_1, \ldots, i_d)$ and a set $T \subseteq [d]$, we define the T-projection of S, denoted by S_T, to be the set $\{i_j : j \in T\}$.

Fix an $\ell \leq k$ outputs of $f_{G,P}$, and let $S_1, \ldots, S_\ell$ be the corresponding hyperedges. By Fact 3.2, we should show that $\mathsf{E}_x[\prod_i P(x_{S_i})] = 0$. For every $x \in \{0,1\}^n$ we have:

$$\prod_{i=1}^{\ell} P(x_{S_i}) = \prod_{i=1}^{\ell} \sum_{T \subseteq [d], |T| \geq a} \alpha_T \chi_T(x_{S_i}) = \sum_{\vec{T} = (T_1, \ldots, T_\ell), |T_i| \geq a} \prod_i \alpha_{T_i} \chi_{S_{i,T_i}}(x).$$

Hence, by the linearity of expectation, it suffices to show that

$$\mathop{\mathsf{E}}_x \left[\prod_i \chi_{S_{i,T_i}}(x) \right] = 0,$$

for every $(T_1, \ldots, T_\ell)$ where $T_i \subseteq [d], |T_i| \geq a$. (Recall that the α_{T_i}'s are constants.) Observe that $\prod_i \chi_{S_{i,T_i}}(x)$ is just a parity function, which, by (k, a)-linearity, is nonzero. Since every nonzero parity function has expectation zero, the claim follows. □

Next, we show that (k, a)-linearity is implied by expansion, and a random graph is likely to be expanding.

Lemma 3.3. *Let $d \geq 3$ be a constant. Let $\Delta \leq \sqrt{n}/\log n$ and $3 \leq a \leq d$.*

1. *Every (m, n, d)-graph which is $(k, d - a/2)$-expanding is also (k, a)-linear.*

2. *A random $(\Delta n, n, d)$-graph is $(\alpha n/\Delta^2, d - a/2)$-expanding* whp*, where α is a constant that depends on a and d.*[5]

Proof. If G is not (k, a) linear then there exists a nonempty collection of $\ell \le k$ hyperedges $S_1, \ldots, S_\ell$ of G and subsets $T_1 \subseteq S_1, \ldots, T_\ell \subseteq S_\ell$, $|T_i| \ge a$ such that the indicator vectors of $T_1, \ldots, T_\ell$ sum up to zero over $\mathbb{F}_2^n$. Therefore every vertex covered by one of $T_1, \ldots, T_\ell$ must be covered at least twice, and so $T_1, \ldots, T_\ell$ can cover at most $\frac{1}{2}(|T_1| + \cdots + |T_\ell|)$ vertices. On the other hand, the total number of vertices covered by $S_1 - T_1, \ldots, S_\ell - T_\ell$ can be at most $|S_1 - T_1| + \cdots + |S_\ell - T_\ell|$. Therefore the collection $S_1, \ldots, S_\ell$ covers at most

$$\frac{1}{2}(|T_1| + \cdots + |T_\ell|) + (|S_1 - T_1| + \cdots + |S_\ell - T_\ell|) = d\ell - \frac{1}{2}(|T_1| + \cdots + |T_\ell|) \le (d - a/2)\ell$$

vertices of G. Thus G is not $(k, d - a/2)$-expanding.

The second item follows by a standard probabilistic calculation. Fix some $t \in (d - \frac{a}{2}, d)$, e.g., $t = d - (a+1)/2$. For $\ell \le k$, we upper bound the probability that there exists a non-expanding subset of size ℓ, i.e. the probability that there exists a set of hyperedges A of size ℓ and a set of vertices B of size ℓt such all the vertices in A belong to B by a union bound:

$$\binom{\Delta n}{\ell} \cdot \binom{n}{t\ell} \cdot \left(\frac{\ell t}{n}\right)^{d\ell} \le \left(\frac{e\Delta n}{\ell}\right)^{\ell} \cdot \left(\frac{en}{t\ell}\right)^{t\ell} \cdot \left(\frac{\ell t}{n}\right)^{d\ell} = \left(\frac{e^{t+1}\Delta n}{\ell}\right)^{\ell} \cdot \left(\frac{\ell t}{n}\right)^{(a/2)\ell} = \left(\frac{e^{t+1}t^{a/2}\Delta}{(n/\ell)^{a/2-1}}\right)^{\ell}.$$

where e denotes the base of the natural logarithm and the inequality follows by the well-known upper-bound $\binom{n}{k} \le \left(\frac{en}{k}\right)^k$. Using the assumption $a \ge 3$, we can upper bound the last expression by $p_\ell = (c_{d,a}\Delta\sqrt{\ell/n})^\ell$, where $c_{d,a}$ is a constant that depends on d and a only. Now observe that

- For $\ell = 1, 2, 3$ we have $p_\ell = O(1/\log n)$,
- For $4 \le \ell \le 10 \log n$ using $\Delta \le \sqrt{n}/\log n$ we obtain $p_\ell \le (c_{d,a}\sqrt{\ell}/\log n)^\ell = O(1/(\log n)^2)$, and
- For $10 \log n \le \ell \le \alpha n/\Delta^2$, we have $p_\ell \le (c_{d,a}\sqrt{\alpha})^\ell = O(1/n^{10})$ for $\alpha = 1/(2c_{d,a})^2$.

Summing up the contributions of p_ℓ to the failure probability, we conclude that the probability G is not $(\alpha n/\Delta^2, d - a/2)$ expanding is at most $o(1)$. □

By combining the lemmas, we obtain the following corollary.

Corollary 3.4. *If P is 2-resilient and $m = \Delta n$ for constant Δ, then* whp *over the choice of an (m, n, d)-graph G, the function $f_{G,P}$ is k-wise independent for $k = \Omega(n)$. If $\Delta = n^\varepsilon$, the above holds with $k = \Omega(n^{1-2\varepsilon})$.*

By taking $\varepsilon < 1/4$, we obtain that 2-resiliency suffices for $\omega(\sqrt{n})$-wise independence with high probability.

3.2 Fooling Heavy Tests

In this section we show that if the predicate P is non-linear and the graph G has large sets of "independent" hyperedges, the function $f_{G,P}$ fools linear tests of weight larger than k. Formally, we will need the following notion of independence.

[5] An event occurs with high probability (*whp* in short) if it happens with probability $1 - o(1)$.

(k, ℓ, b)**-independence.** Let $\mathcal{S}$ be a collection of k distinct hyperedges. A subset $\mathcal{T} \subseteq \mathcal{S}$ of ℓ distinct hyperedges is an (ℓ, b)-independent set of $\mathcal{S}$ if the following two properties hold: (1) Every pair of hyperedges $(T_i, T_j) \in \mathcal{T}$ are of distance at least 2, namely, for every pair $T_i \neq T_j \in \mathcal{T}$ and $S \in \mathcal{S}$,

$$T_i \cap S = \emptyset \text{ or } T_j \cap S = \emptyset;$$

and (2) For every $T_i \in \mathcal{T}$ and $S \neq T_i$ in $\mathcal{S}$ we have

$$|T_i \cap S| < b.$$

A graph is (k, ℓ, b)-independent if every set of hyperedges of size larger than k has an (ℓ, b)-independent set.

Our key lemma shows that good independence and large algebraic degree guarantee resistance against heavy linear tests.

Lemma 3.5. *If an (m, n, d)-graph G is (k, ℓ, b)-independent and P has an algebraic degree of at least b, then every linear test of size at least k has bias of at most $\frac{1}{2}e^{-2\ell/2^d}$.*

Proof. Fix some test $\mathcal{S} = (S_1, \ldots, S_k)$ of size k, and let $\mathcal{T} = (T_1, \ldots, T_\ell)$ be an (ℓ, b)-independent set of $\mathcal{S}$. Fix an arbitrary assignment σ for all the input variables which do not participate in any of the T_i's and choose the other variables uniformly at random. In this case, we can partition the output of the test y to ℓ summands over ℓ disjoint blocks of variables, namely

$$y = \sum_{i \in [k]} P(x_{S_i}) = \sum_{i \in [\ell]} z_i(x_{T_i}),$$

where

$$z_i(x_{T_i}) = P(x_{T_i}) + \sum_{S : T_i \neq S \cap T_i \neq \emptyset} P(x_{S \cap T_i}, \sigma_{S \setminus T_i}),$$

and the sums are over $\mathbb{F}_2$. We need two observations: (1) the random variables z_i's are statistically independent (as each of them depends on a disjoint block of inputs); and (2) the r.v. z_i is non-constant and, in fact, it takes each of the two possible values with probability at least 2^{-d}. To prove the latter fact it suffices to show that $z_i(x)$ is a non-zero polynomial (over $\mathbb{F}_2$) of degree at most d. Indeed, recall that z_i is the sum of the polynomial $P(x_{T_i})$ whose degree is in $[b, d]$, and polynomials of the form $P(x_{S \cap T_i}, \sigma_{S \setminus T_i})$ whose degree is smaller than b (as $|S \cap T_i| < b$). Therefore the degree of z_i is in $[1, d]$.

To conclude the proof, we note that the parity of ℓ independent coins, each with expectation in $(\delta, 1 - \delta)$, has bias of at most $\frac{1}{2}(1 - 2\delta)^\ell$. (See, e.g., [28]). □

We want to show that a random graph is likely to be $(k, \ell, 2)$-independent.

Lemma 3.6. *For every positive ε and δ. A random $(n^{1+\varepsilon}, n, d)$-graph is,* whp*, $(n^{2\varepsilon+\delta}, n^{\delta/2}, 2)$ independent.*

Proof. We will need the following claim. Call a hyperedge S b-intersecting if there exists another hyperedge S' in the graph for which $|S' \cap S| \geq b$. We first bound the number of b-intersecting hyperedges.

Claim 3.7. *Let b be a constant. Then, in a random $(m = n^{1+\varepsilon}, n, d)$-graph,* whp*, the number of b-intersecting hyperedges is at most $n^{2(1+\varepsilon)-b} \log n$.*

Hence, *whp*, at most $O(n^{2\varepsilon} \log n)$ of the hyperedges are 2-intersecting, and for $\varepsilon < 1/4$ there are at most $o(\sqrt{n})$ such hyperedges.

Proof (of Claim 3.7). Let X be the random variable which counts the number of b-intersecting hyperedges. First, we bound the expectation of X by $m^2 d^{2b}/n^b = d^{2b} \cdot n^{2(1+\varepsilon)-b}$. To prove this, it suffices to bound the expected number of pairs S_i, S_j which b-intersects. Each such pair b-intersects with probability at most d^{2b}/n^b, and so, by linearity of expectation, the expected number of of intersecting pairs is at most $m^2 d^{2b}/n^b$. Now, by applying Markov's inequality, we have that $\Pr[X > \frac{\log n}{d^{2b}} \mathsf{E}[X]] < d^{2b}/\log n = o(1)$, and the claim follows. (A stronger concentration can be obtained via a martingale argument.) □

We can now prove Lemma 3.6. Assume, without loss of generality, that $\varepsilon > 1$ (as if the claim holds for some value of ε it also holds for smaller values). First observe that, *whp*, all the input nodes in G have degree at most $2n^{\varepsilon}$. As by a multiplicative chernoff bound, the probability that a single node has larger degree is exponentially small in n^{ε}. We condition on this event and the event that there are no more than $r = n^{2\varepsilon} \log n$ 2-expanding edges. Fix a set of $k = n^{2\varepsilon+\delta}$ hyperedges. We extract an $(\ell, 2)$-independent set by throwing away the 2-expanding edges, and then by iteratively inserting an hyperedge T into the independent set and removing all the hyperedges S that share with T a common node, and the hyperedges which share a node with an edge, that shares a node with T. At the beginning we removed at most r edges, and in each iteration we remove at most $(d2n^{\varepsilon})^2$ edges, hence there are at least $\ell \geq \frac{k-r}{4d^2 n^{2\varepsilon}} > n^{\delta/2}$ hyperedges in the independent set. □

Combining the lemmas together we get:

Corollary 3.8. *Fix some positive ε and δ. If P has an algebraic degree of at least 2 and $m = n^{1+\varepsilon}$, then,* whp *over the choice of a random (m, n, d)-graph, the function $f_{G,P}$ has at most sub-exponential bias (i.e., $\exp(-\Omega(n^{\delta}))$) against linear tests of size at least $n^{2\varepsilon+2\delta}$.*

By combining Corollaries 3.4 and 3.8, we obtain Theorem 1.1.

4 Linear Tests Break Degenerate Predicates

In this section we prove Theorem 1.2; That is, we show that the assumptions that P is non-linear and 2-resilient are necessary for P to be a hard predicate. Clearly the assumption that P is non-linear is necessary even when $m = n + 1$.

When $m \geq Kn$ for a sufficiently large constant K (depending on d), it follows from work of Cryan and Miltersen [17] that if P is not 1-resilient, then for any $f\colon \{\pm 1\}^n \to \{\pm 1\}^m$, the output of f is distinguishable from uniform with constant advantage by some linear test. When P is 1-resilient but not 2-resilient, Mossel, Shpilka, and Trevisan show that f is distinguishable from uniform by a polynomial-time algorithm, but not by one that implements a linear test.

Here we show that if P is not 2-resilient, then the output of $f_{G,P}$ is distinguishable by linear tests with non-negligible advantage with high probability over the choice of G.

Claim 4.1. *Let $K > 4$ and $d \in \mathbb{N}$ be constants. Assume that the predicate $P : \{0,1\}^d \to \{0,1\}$ is unbiased and 1-resilient but not 2-resilient, i.e., $|\mathsf{E}[P(z)z_1z_2]| = \alpha > 0$. Then for every $\ell = o(\log n)$, with probability $1 - (2^{-\Omega(\ell)} + d\ell/n)$ over the choice of a random (Kn, n, d)-graph G, there exists a linear test that distinguishes the output of $f_{G,P}$ from random with advantage α^ℓ.*

Proof. Let H be the directed graph with vertices $\{1, \ldots, n\}$ where every hyperedge $(i_1, i_2, \ldots, i_d)$ in G induces the edge (i_1, i_2) in H.

Let ℓ be the length of the shortest directed cycle in H and without loss of generality assume that this cycle consists of the inputs $1, 2, \ldots, \ell$ in that order. Let z_i be the name of the output that involves inputs i and $i+1$ for i ranging from 1 to ℓ (where i is taken modulo ℓ) and S_i the corresponding hyperedge. With probability at least $1 - d\ell/n$, input i does not participate in any hyperedge besides S_i and S_{i+1} and all other inputs participate in at most one of the hyperedges $S_1, \ldots, S_\ell$.

We now calculate the bias of the linear test that computes $z_1 \oplus \ldots \oplus z_\ell$. For simplicity, we will assume that $d = 3$; larger values of d can be handled analogously but the notation is more cumbersome. We will denote the entries in S_i by i, $i+1$ and i'. Then the Fourier expansion of $z_i(x_{S_i})$ has the form

$$z_i(x_{S_i}) = \alpha x_i x_{i+1} + \beta x_i x_{i'} + \gamma x_{i+1} x_{i'} + \delta x_i x_{i+1} x_{i'}$$

The Fourier expansion of the expression $\mathsf{E}[z_1(x_{S_1}) \ldots z_\ell(x_{S_\ell})]$ can be written as a sum of 4^ℓ products of different monomials participating in the above terms. The only monomial that does not vanish is the one containing all the α-terms, namely

$$\mathsf{E}\Big[\prod_{i=1}^{n} \alpha x_i x_{i+1}\Big] = \alpha^\ell.$$

All the other products of monomials contain at least one unique term of the form $x_{i'}$, and this causes the expectation to vanish.

It remains to argue that with high probability ℓ is not too large. We show that with probability $1 - O((4/K)^\ell)$, H has a directed cycle of length ℓ, as long as $\ell < \log_{2K}(n/4)$. Let X denote the number of directed cycles of length ℓ in H. The number of potential directed cycles of length in H is $n(n-1)\ldots(n-\ell+1) \geq (n-\ell)^\ell$. Each of these occurs uniquely in H with probability of at least

$$(Kn)(Kn-1)\ldots(Kn-\ell+1)\Big(\frac{1}{n(n-1)}\Big)^\ell\Big(1 - \frac{1}{n(n-1)}\Big)^{Kn-\ell} \geq \Big(\frac{Kn-\ell}{n^2}\Big)^\ell.$$

Therefore $\mathsf{E}[X] \geq (K/4)^\ell$. The variance can be upper bounded as follows. The number of *pairs* of cycles of length ℓ that intersect in i edges is at most $\binom{\ell}{i} n^{2\ell-i-1}$, and the covariance of the indicators for these cycles is at most $(K/n)^{2\ell-i}$. Adding all the covariances up as i ranges from 1 to ℓ, it follows that

$$\mathsf{Var}[X] \leq \mathsf{E}[X] + \sum_{i=1}^{\ell} \binom{\ell}{i} n^{2\ell-i-1}\Big(\frac{K}{n}\Big)^{2\ell-i} \leq \mathsf{E}[X] + \frac{2^\ell K^{2\ell}}{n}.$$

By Chebyshev's inequality,

$$\Pr[X = 0] \leq \frac{\mathsf{Var}[X]}{\mathsf{E}[X]^2} < \frac{2}{\mathsf{E}[X]}$$

as long as $\ell < \log_{2K}(n/4)$. □

5 Implications of Small-Bias

For local functions with large stretch, small bias seems like a good approximation for cryptographic pseudorandomness. Specifically, we are not aware of any local function $f_{G,P}$ with linear stretch that fools linear distinguishers but can be distinguished by some polynomial-time adversary.[6] One may conjecture that if $f_{G,P}$ fools linear adversaries for most graphs, then it also fools polynomial-time adversaries. In other words, local functions are too simple to "separate" between the two different notions. We attempt to support this view by showing that resilience against small-bias, by itself, leads to robustness against other classes of attacks.

First, we observe that, for local functions, k-wise independence follows directly from ε-bias. (This is not the case for non-local functions.)

Lemma 5.1. *Let $f : \{0,1\}^n \to \{0,1\}^m$ be a d-local function which is 2^{-kd}-biased. Then, it is also k-wise independent.*

Proof. Assume towards a contradiction that f is not k-wise independent. Then, there exists a set of k outputs T and a linear distinguisher L for which

$$\varepsilon = \left| \Pr_{y \xleftarrow{R} f(\mathcal{U}_n)} [L(y_T) = 1] - \Pr[L(\mathcal{U}_k) = 1] \right| > 0,$$

where y_T denotes the restriction of the string y to the indices in T. Since f is d-local, y_T is sampled by using less than kd bits and therefore $\varepsilon \geq 2^{-kd}$. □

Note that the proof of our main theorem establishes k-wise independent as an intermediate step (Section 3.1). However, the above lemma is stronger in the sense that it holds for every fixed graph and every output length including ones that are not covered by the main theorem.

By plugging in known results about k-wise independent distributions, it immediately follows that if a local function is sufficiently small-biased, then it is pseudorandom against $\mathbf{AC^0}$ circuits [15], linear threshold functions over the reals [18], and degree-2 threshold functions over the reals [19].

Moreover, attacks on local functions, which are actively studied at the context of algorithms for constraint-satisfaction problems, appear to be based mainly on "local" heuristics (DPLL, message-passing algorithms, random-walk based algorithms) or linearization [9]. Hence, it appears that in the context of local functions, the small-bias property already covers all "standard" attacks. We support this intuition by showing that small-biased local functions (on a random-looking input-output graph) are not merely min-wise independent, but have a stronger property: Even after reading an arbitrary set of t-outputs, the posterior distribution on *every* set of ℓ inputs, while not uniform, still has h bits of min-entropy. We refer to this property as (t, ℓ, h)-*robustness*.

Lemma 5.2. *Suppose that P is a predicate for which $f_{G,P} : \{0,1\}^n \to \{0,1\}^m$ is k-wise independent,* whp *over the choice of a random (m, n, d) graph G. Then,* whp *over the choice of a random $(m' = \Omega(m), n, d)$ graph H, the function $f_{H,P} : \{0,1\}^n \to \{0,1\}^{m'}$ is $(t = \Omega(k), \ell, h)$-robust, where $h = \min\left(\ell, \Omega(m \cdot (\ell/n)^d), \Omega(k)\right)$.*

(See Section A for more details and proof.) Robustness holds with inverse-polynomial parameters ($t = n^\alpha, \ell = n^\beta, h = n^\gamma$) when $m = n^{1+\varepsilon}$, and with linear parameters when $m = O(n)$ is

[6] It can be shown that this is false when the stretch is sub-linear.

linear. The notion of robustness is the main technical tool used by Cook et al. [16] to prove that myopic backtracking algorithms cannot invert $f_{G,P}$ in polynomial time (for the case $m = n$).[7] By Lemma 5.2, robustness follows directly "for free" from small-bias, and thus we can derive a similar lower-bound for larger output lengths (but for a smaller class of predicates). (See Section A for details.)

References

[1] M. Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307. IEEE Computer Society, 2003.

[2] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov, and A. Wigderson. Pseudorandom generators in propositional proof complexity. *SIAM Journal of Computation*, 34(1):67–88, 2004.

[3] M. Alekhnovich, E. A. Hirsch, and D. Itsykson. Exponential lower bounds for the running time of DPLL algorithms on satisfiable formulas. *J. Autom. Reasoning*, 35(1-3):51–72, 2005.

[4] B. Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. *Electronic Colloquium on Computational Complexity (ECCC)*, 18, 2011.

[5] B. Applebaum, B. Barak, and A. Wigderson. Public-key cryptography from different assumptions. In *42nd ACM Symposium on Theory of Computing, (STOC 2010)*, pages 171–180, 2010.

[6] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM Journal on Computing*, 36(4):845–888, 2006.

[7] B. Applebaum, Y. Ishai, and E. Kushilevitz. On pseudorandom generators with linear stretch in NC^0. *Journal of Computational Complexity*, 17(1):38–69, 2008.

[8] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography with constant input locality. *Journal of Cryptology*, 22(4):429–469, 2009.

[9] S. Arora and R. Ge. New algorithms for learning in presence of errors. In *ICALP (1)*, pages 403–415, 2011.

[10] E. Ben-Sasson and A. Wigderson. Short proofs are narrow - resolution made simple. In *STOC*, pages 517–526, 1999.

[11] A. Bogdanov, P. Papakonstantinou, and A. Wan. Pseudorandomness for read-once formulas. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, 2011. To appear.

[12] A. Bogdanov and Y. Qiao. On the security of goldreich's one-way function. In *APPROX-RANDOM*, pages 392–405, 2009.

[7] The two other high-level ingredients are an upper-bound on the number of siblings of a random input, and standard lower-bound on the resolution size of unsatisfiable formulas (cf. [10, 2]).

[13] A. Bogdanov and A. Rosen. Input locality and hardness amplification. In *Proc. of 8th TCC*, pages 1–18, 2011.

[14] A. Bogdanov and E. Viola. Pseudorandom bits for polynomials. *SIAM J. Comput.*, 39(6):2464–2486, 2010.

[15] M. Braverman. Poly-logarithmic independence fools AC^0 circuits. *Computational Complexity, Annual IEEE Conference on*, 0:3–8, 2009.

[16] J. Cook, O. Etesami, R. Miller, and L. Trevisan. Goldreich's one-way function candidate and myopic backtracking algorithms. In O. Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 521–538. Springer, 2009.

[17] M. Cryan and P. B. Miltersen. On pseudorandom generators in NC^0. In *Proc. 26th MFCS*, 2001.

[18] I. Diakonikolas, P. Gopalan, R. Jaiswal, R. A. Servedio, and E. Viola. Bounded independence fools halfspaces. *SIAM Journal of Computation*, 39(8):3441–3462, 2010.

[19] I. Diakonikolas, D. M. Kane, and J. Nelson. Bounded independence fools degree-2 threshold functions. In *FOCS*, pages 11–20, 2010.

[20] S. O. Etesami. Pseudorandomness against depth-2 circuits and analysis of goldreich's candidate one-way function. Technical Report EECS-2010-180, UC Berkeley, 2010.

[21] M. X Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *JACM: Journal of the ACM*, 42, 1995.

[22] O. Goldreich. Candidate one-way functions based on expander graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(090), 2000.

[23] O. Goldreich. Three XOR-lemmas - an exposition. In O. Goldreich, editor, *Studies in Complexity and Cryptography*, volume 6650 of *Lecture Notes in Computer Science*, pages 248–272. Springer, 2011.

[24] R. Impagliazzo, N. Nisan, and A. Wigderson. Pseudorandomness for network algorithms. In *In Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 356–364, 1994.

[25] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography with constant computational overhead. In R. E. Ladner and C. Dwork, editors, *STOC*, pages 433–442. ACM, 2008.

[26] D. Itsykson. Lower bound on average-case complexity of inversion of goldreich's function by drunken backtracking algorithms. In *Computer Science - Theory and Applications, 5th International Computer Science Symposium in Russia*, pages 204–215, 2010.

[27] R. Miller. Goldreich's one-way function candidate and drunken backtracking algorithms. Distinguished major thesis, University of Virginia, 2009.

[28] E. Mossel, A. Shpilka, and L. Trevisan. On ϵ-biased generators in NC^0. In *Proc. 44th FOCS*, pages 136–145, 2003.

[29] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993. Preliminary version in Proc. 22th STOC, 1990.

[30] S. K. Panjwani. An experimental evaluation of goldreich's one-way function. Technical report, IIT, Bombay, 2001.

[31] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, 30(5):776–778, 1984.

A Robustness and Myopic Backtracking Algorithms

Robustness. Let $f : \{0,1\}^n \to \{0,1\}^m$. Let $L \subset [n]$ be a set of inputs, and $t, h \in [m]$. We say that f is (t, L, h)-robust if for every set of outputs $T \subset [m]$ of size t and every string $z \in \{0,1\}^t$ the following holds. Let $x \in \{0,1\}^n$ be a uniformly chosen string conditioned on the event $f(x)_T = z$, i.e., the outputs which are indexed by T equal to z. Then the random variable $x_L = (x_i)_{i \in L}$ has min-entropy of h, namely, for every fixed $w \in \{0,1\}^{|L|}$, $\Pr[x_L = w] \leq 2^{-h}$. The function is (t, ℓ, h)-robust if it is (t, L, h)-robust for every ℓ-size input set L.

We show that if $f_{G,P}$ is k-wise independent with respect to random graph, then it is also robust for shorter output length. (The proof is deferred to Section A.1).

Lemma A.1 (Lemma 5.2 restated)**.** *Suppose that P is a predicate for which $f_{G,P} : \{0,1\}^n \to \{0,1\}^m$ is k-wise independent,* whp *over the choice of a random (m, n, d) graph G. Then,* whp *over the choice of a random $(m - r, n, d)$ graph H, the function $f_{H,P} : \{0,1\}^n \to \{0,1\}^{m-r}$ is (t, ℓ, h)-robust, where $h = \min\left(\ell, r \cdot (\ell/n)^d/2, k - t\right)$.*

In the case of linear stretch, $m = n + O(n)$, where k is linear as well (Corollary 3.4), one can get (t, ℓ, h)-robustness with linear parameters at the expense of linear decrease in the output length (e.g., $r = m/2$). When the output is polynomial $m = n^{1+\varepsilon}$ (for $\varepsilon < 1/4$), we get (t, ℓ, h)-robustness for inverse-polynomial parameters, again at the expense of a linear decrease in the output length (e.g., $r = m/2$).

Robustness is especially useful if the actual number of preimages of $y = f_{G,P}(x)$ is relatively small compared to 2^h. In this case, an algorithm which attempts to guess ℓ bits of a preimage x based on t outputs is likely to be wrong (obtain a partial assignment that does not correspond to any preimage of y.) We show that in our setting of parameters (when the output length is large) most inputs have a small number of siblings under $f_{G,P}$ (where G is random). The proof of the following lemma is deferred to Section A.2.

Lemma A.2. *Let P be any nonconstant predicate. For $m > \Omega_d(n \log n)$,*

$$\Pr_{G,x}[\left|\left\{x' | x' \text{ is a preimage of } f_{G,P}(x)\right\}\right| < M(n)] > 1 - o(1),$$

where $M(n)$ is any (arbitrary slow) increasing function $M(n) = \omega(1)$.

Myopic DPLL algorithms. We now show how the simple statistical properties proved in the above lemmas yield lower-bounds for DPLL algorithms who attack $f_{G,P}$. The high-level argument is similar to the one used in [3, 16] and it is only sketched here. Consider the following myopic backtracking DPLL algorithm, whose input consists of $y = f_{G,P}(x)$ where x is uniformly chosen. The algorithm is allowed to read the entire graph G, but it reads the values of y in an incremental way. Specifically, in each iteration the algorithm adaptively choose an input variable x_i and asks to reveal r new output bits of y. Then it guesses the value of x_i based on its current state and on the output bits that were already revealed (including the ones that were revealed in previous iterations). If the algorithm reaches a contradiction, i.e., its partial assignment to x is consistent with some output it backtracks.

Suppose that $f_{G,P}$ satisfies the above lemmas. (Think of $M = O(\log n)$ and k, t, ℓ, h as polynomial in n, or even linear in n when $m = O(n)$.) Since $f_{G,P}$ is k-wise independent the algorithm does not backtrack in the first k/r steps (as some patrial assignment is consistent with every value of k outputs). Since f is $(r \cdot \ell, \ell, h)$-robust and the number of siblings of a random x is (*whp*) M, the partial assignment chosen by the algorithm after $\ell < k$ steps is likely to be globally inconsistent (there are 2^h locally consistent assignments while there are only $M \ll 2^h$ globally consistent assignments). Hence, with all but negligible probability, the algorithm will err during the first ℓ steps, and therefore will backtrack at some point after more than k steps. It can be shown (by standard lower-bound on resolution [10, 2]) that, for a random graph, the backtracking phase takes super-polynomial time. (By plugging in the exact parameters the lower-bound is exponential $2^{\Omega(n)}$ when $m = O(n)$ or sub-exponential $\exp(n^{\delta})$ when $m = n^{1+\varepsilon}$.)

A.1 Proving Robustness (Lemma A.1)

Proof. Observe that an $(m - r, n, d)$ graph H with hyperedges $(S_1, \ldots, S_{m-r})$ can be extended to an (m, n, d) graph G by adding r hyperedges $(S_{r+1}, \ldots, S_m)$. The additional edges can be packed together in (r, n, d) graph H' to which we refer as an *extension* graph. Call H *good* if $f_{H \cup H',P}$ is k-wise independent *whp* over the choice of the (r, n, d) graph H'. Since $f_{G,P}$ is k-wise independent, *whp* over a random (m, n, d) graph G, it follows from Markov's inequality that all but $o(1)$ of the $(m - r, n, d)$ graphs H are good. We show that if H is a good graph the function $f_{H,P}$ is (t, ℓ, h)-robust.

Fix some good H. Let L be an arbitrary ℓ-size subset of the inputs. We say that an extension graph $H' = (S_{r+1}, \ldots, S_m)$ is *good* (for L and G) if:

1. There is a set $M \subset \{r + 1, \ldots, m\}$ of at least h hyperedges of H' which fall completely into L, i.e., $\bigcup_{i \in M} S_i \subseteq L$.
2. $f_{H \cup H',P}$ is k-wise independent.

Claim A.3. *If there exists a good extension H', then $f_{H,P} : \{0,1\}^n \to \{0,1\}^{m-r}$ is (t, L, h)-robust.*

Proof. Fix some output set $T \subset [m - r]$ of size t. Let $x \overset{R}{\leftarrow} \{0,1\}^n$ and let $y = f_{H \cup H',P}(x)$. Fix the value of y_T to some string z. Then, since the random variables $y = (y_1, \ldots, y_m)$ are k-wise independent, the distribution of y_M conditioned on $y_T = z$ is uniform over $\{0,1\}^h$. (Note that $h + t \leq k$.) Since y_M depends only on x_L it follows that the conditional distribution of x_L has min-entropy at least $|M| = h$. (Otherwise, x_L takes some value w with probability larger than 2^{-M} and the string $f_{M,P}(w)$ occurs in y_M with probability larger than 2^{-M} contradicting uniformity.) $\square$

Finally, a simple calculation shows that there exists a good extension graph (in fact, many of them are good for L). Indeed, a random H' is expected to have $r \cdot (\ell/n)^d \geq 2h$ hyperedges in L, and therefore by Markov's inequality at most $\frac{1}{2}$ of the extension graphs violate (1). Also, since H is assumed to be good, at most $o(1)$ of the extension graphs violate (2), and therefore at least $\frac{1}{2} - o(1)$ of the extension graphs are good, and the lemma follows. □

A.2 Bounding the number of siblings (Lemma A.2)

We prove Lemma A.2 via the following claim.

Claim A.4. *Let P be any nonconstant predicate. For $m > 2^{Kd} n \log n$, $\Pr_{G,x,y}[f_{G,P}(x) = f_{G,P}(y)] < K2^{-n}$, where K is some constant and n is sufficiently large.*

We show that the claim implies Lemma A.2. For every fixed G and x define $\xi_{x,G} = \Pr_y[f_{G,P}(x) = f_{G,P}(y)]$. The claim shows that $\mathsf{E}_{G,x}[\xi_{G,x}] < K2^{-n}$. Hence, by Markov's inequality, for every M, $\Pr_{G,x}[\xi_{G,x} < MK2^{-n}] > 1 - 1/M$, by taking $M = \omega(1)$, we get that, *whp* over the choice of G and x, there are at most $MK2^{-n}2^n = O(MK)$ siblings for x under $f_{G,P}$. We now prove the claim.

proof of Claim A.4. We write

$$\Pr_{G,x,y}[f_{G,P}(x) = f_{G,P}(y)] = \mathop{\mathsf{E}}_{x,y} \Pr_G[f_{G,P}(x) = f_{G,P}(y)] = \mathop{\mathsf{E}}_{x,y}[\Pr_I[P(x|_I) = P(y|_I)]^m]$$

where I is a random sequence of d indices from $[n]$. The value of the inner probability only depends on x and y through the number of pairs $x_i y_i$ of types 00, 01, 10, and 11. Let n_{ab} be the number of pairs $x_i y_i$ where $x_i = a$ and $y_i = b$. Then $\mathcal{D} = (n_{00}/n, n_{01}/n, n_{10}/n, n_{11}/n)$ is a probability distribution over $\{0,1\}^2$ and we can write

$$\mathop{\mathsf{E}}_{x,y}[\Pr_I[P(x|_I) = P(y|_I)]^m] = \frac{1}{2^{2n}} \sum_{n_{00}+n_{01}+n_{10}+n_{11}=n} \binom{n}{n\mathcal{D}} \Pr_{uv \sim \mathcal{D}^d}[P(u) = P(v)]^m$$

where u, v are d-bit strings, $\binom{n}{n\mathcal{D}}$ is shorthand for $\binom{n}{n_{00}, n_{01}, n_{10}, n_{11}}$, and $\mathcal{D}^d$ is the distribution on uv obtained by choosing each pair $u_i v_i$ independently from the joint distribution $\mathcal{D}$.

We divide the sum into four parts depending on the values $n_{00}, n_{01}, n_{10}, n_{11}$) as follows:

- Let $UNBAL$ be those (x, y) such that $n_{ab} \geq 5/6$ for some $a, b \in \{0, 1\}$.
- Let $EQ - UNEQ$ be those (x, y) such that $n_{aa} \geq 1/36$ and $n_{b\bar{b}} \geq 1/36$ for some $a, b \in B$.
- Let $EQ+$ be those (x, y) outside $UNBAL$ and $EQ - UNEQ$ such that $n_{00} + n_{11} \geq 1/2$.
- Let $UNEQ+$ be those (x, y) outside $UNBAL$ and $EQ - UNEQ$ such that $n_{01} + n_{10} \geq 1/2$.

We bound the contribution of each of these sets to the sum.

An entropy calculation (see Section 6 in [13]) shows that the number of (x, y) in $UNBAL$ is at most $2^{0.92n}$ and so the contribution of $UNBAL$ is at most $2^{-1.08n}$.

Let us now look at $(x, y) \in EQ - UNEQ$. For simplicity let's take the case $n_{00}, n_{01} \geq 1/36$, the other cases being similar. Then for any $a \in \{0,1\}^d$, the event "$u = 0^d$ and $v = a$" occurs with probability at least 36^{-d}, and so $\Pr[P(u) = P(v)] \leq 1 - 36^{-d}$, and

$$\sum_{(x,y) \in EQ-UNEQ} \Pr_{uv \sim \mathcal{D}^d}[P(u) = P(v)]^m \leq 2^{2n}(1 - 36^{-d})^m \leq 2^{-2n}$$

by our choice of m.

We now consider the pairs $(x, y) \in EQ+$. By definition of $EQ+$, for all such pairs we have $n_{00}, n_{11} \geq 1/36$. Let $E = \{i \colon x_i = y_i\}$. Since P is not constant, there must exist a pair $u, v \in \{0,1\}^d$ that differ in exactly one coordinate such that $P(u) \neq P(v)$. Therefore

$$\Pr[P(u) \neq P(v)] \geq 1 - 36^{-(d+1)} \cdot \frac{|E|}{n}.$$

We can now write

$$\begin{aligned}
\sum_{(x,y)\in EQ+} \Pr_{uv\sim\mathcal{D}^d}[P(u) = P(v)]^m &\leq \sum_{k=0}^{n/2} \sum_{|E|=k} \sum_{(x,y) \text{ agree on } E} \Big(1 - 36^{-(d+1)} \cdot \frac{k}{n}\Big)^m \\
&= 2^n \sum_{k=0}^{n/2} \binom{n}{k} \Big(1 - 36^{-(d+1)} \cdot \frac{k}{n}\Big)^m \\
&\leq 2^n + 2^n \sum_{k=1}^{n/2} \Big(\frac{en}{k}\Big)^k \Big(1 - 36^{-(d+1)} \cdot \frac{k}{n}\Big)^m \\
&= 2^n + 2^n \sum_{k=1}^{n/2} \exp(k \ln(en/k) - k \ln(2en)) = O(2^n)
\end{aligned}$$

by our choice of m.

Finally we consider those $(x, y) \in UNEQ+$. Then we have $n_{01}, n_{10} \geq 1/36$. Say P is *symmetric* if $P(\overline{w}) = P(w)$ for every w. If P is symmetric, we can bound the contribution of $UNEQ+$ by $O(2^n)$ by a calculation analogous to the one for $EQ+$. If P is not symmetric, then $P(w) \neq P(\overline{w})$ for some w. The event event "$u = w, v = \overline{w}$" then happens with probability at least 36^{-d} and we can bound the contribution of $UNEQ+$ by 2^{-2n} by a calculation analogous to the one for $EQ - UNEQ$. □