Robust Pseudorandom Generators[*]

Yuval Ishai[†] Eyal Kushilevitz[‡] Xin Li[§] Rafail Ostrovsky[¶]
Manoj Prabhakaran[‖] Amit Sahai[**] David Zuckerman[††]

Abstract

Let $G : \{0,1\}^n \to \{0,1\}^m$ be a pseudorandom generator. We say that a circuit implementation of G is (k, q)*-robust* if for every set S of at most k wires anywhere in the circuit, there is a set T of at most $q|S|$ outputs, such that conditioned on the values of S and T the remaining outputs are pseudorandom. We initiate the study of robust PRGs, presenting explicit and non-explicit constructions in which k is close to n, q is constant, and $m >> n$. These include unconditional constructions of robust r-wise independent PRGs and small-bias PRGs, as well as conditional constructions of robust cryptographic PRGs.

In addition to their general usefulness as a more resilient form of PRGs, our study of robust PRGs is motivated by cryptographic applications in which an adversary has a local view of a large source of secret randomness. We apply robust r-wise independent PRGs towards reducing the randomness complexity of private circuits and protocols for secure multiparty computation, as well as improving the "black-box complexity" of constant-round secure two-party computation.

[*] This is a revised full version of [32]. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

[†] Computer Science Department, Technion (yuvali@cs.technion.il). Supported by the European Research Council as part of the ERC project CaC (grant 259426).

[‡] Computer Science Department, Technion (eyalk@cs.technion.il). Supported by ISF grant 1361/10 and BSF grant 2008411.

[§] Computer Science Department, University of Washington (lixints@cs.washington.edu). Supported by a Simons postdoctoral fellowship.

[¶] Computer Science Department and Mathematics Department, University of California, Los Angeles (rafail@cs.ucla.edu). Research supported in part by NSF grants CNS-0830803; CCF-0916574; IIS-1065276; CCF-1016540; CNS-1118126; CNS-1136174, 1619384; US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is also based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392.

[‖] Computer Science Department, University of Illinois at Urbana-Champaign (mmp@uiuc.edu). Research supported in part by NSF grants 1228856 and 0747027 and a Ramanujan Fellowship.

[**] Computer Science Department, University of California, Los Angeles (sahai@cs.ucla.edu). Research supported in part from a DARPA/ONR PROCEED award, NSF grants 1228984, 1136174, 1118096, 1065276, 0916574 and 0830803, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0389.

[††] Computer Science Department, University of Texas at Austin (diz@cs.utexas.edu) Research supported by NSF Grants CCF-0916160, CCF-1218723, and DMS-0835373.

1 Introduction

Pseudorandomness is a central tool in complexity theory and cryptography. A *pseudorandom generator* (PRG) is a deterministic function $G : \{0,1\}^n \to \{0,1\}^m$ which stretches a short random seed into a longer output which looks random to any computationally bounded distinguisher. The question we ask in this work can be pictorially described as follows. Consider an implementation of G by a boolean circuit, and suppose that an attacker can observe a set S of k wires anywhere in the circuit. Since S may contain output wires, the output conditioned on S may no longer look random. But how big is the "shadow" S can cast on the output? Can we design PRG implementations in which the effect of observing any such S is localized to roughly k bits of the output?

We formalize the above question via the notion of *robust pseudorandom generators.* We say that a circuit implementation of G is (k, q)-robust if for every set S of at most k wires anywhere in the circuit there is a set T ("shadow") of at most $q|S|$ outputs such that conditioned on the values of S, the outputs outside T are pseudorandom. We will be mainly interested in a stronger notion of robustness in which the conditioning is on both S and T; if such a stronger requirement is met we say that G is *strongly* (k, q)-robust. We consider the robustness of three distinct types of PRG: *r-wise independent* PRGs, where the distinguisher can observe any r bits of the output, *small-bias* PRGs [43], where the distinguisher can compute the parity of any subset of the outputs, and *cryptographic* PRGs [15, 49], where the distinguisher can perform arbitrary polynomial-time computations.

To motivate the notion of robust PRGs, consider a simple application of cryptographic PRGs for one-time symmetric encryption. To encrypt a long message $M \in \{0,1\}^m$ with a short secret key $K \in \{0,1\}^n$, it suffices to compute $C = M \oplus G(K)$. Since $G(K)$ is indistinguishable from random, so is C. Now, suppose that $k << n$ intermediate values in the computation of C are leaked. What do these values together with C reveal about M? The dense model theorem [48, 46, 23] assures us that if G is sufficiently strong, then M is indistinguishable from some source whose min-entropy is roughly $m-k$. However, even a single lost bit of entropy can correspond to *global* information about m. For instance, if an intermediate value reveals the parity of $G(K)$, this information together with C reveals the parity of M. Our goal is to provide the guarantee that if arbitrary k physical bits are leaked during the process of computing C, this is no worse than leaking (roughly) k physical bits of M.

We turn to the question of constructing robust PRGs, starting with some simple observations. First, if $k = n$, the set S can include the entire PRG seed, conditioned on which the entire output is fixed. We thus restrict the attention to the case where $k < n$. Second, allowing n to be much bigger than k, we can use the following naïve construction: if $G' : \{0,1\}^{n'} \to \{0,1\}^m$ is a PRG then $G : \{0,1\}^{n'(k+1)} \to \{0,1\}^m$ defined by $G(x_1, \ldots, x_{k+1}) = G'(x_1) \oplus G'(x_2) \oplus \cdots \oplus G'(x_{k+1})$ (computed in the natural way) is a strong $(k, 1)$ robust PRG. The main weakness of this construction is that its seed length is far from optimal. A secondary weakness, which turns out to be crucial for one of our motivating applications, is that the *circuit size* of G is much bigger than its output length. Our main goal in this work is to construct robust PRGs in which n is very close to k, while keeping q constant and maximizing the stretch function $m(n)$. As a secondary goal, we would like to minimize the circuit size of robust PRGs. These goals are nontrivial to meet also when considering non-explicit constructions.

1.1 Our Results

We present several constructions of robust PRGs with near-optimal parameters. These include:

- **Robust r-wise independent PRGs:** Using explicit constructions of unbalanced lossless expanders [19, 28], we get constructions of strong (k, q)-robust r-wise PRGs with $q = O(1)$ and either $r, k = \Omega(n)$ and linear stretch, or $r, k = n^{1-\eta}$ and arbitrary polynomial (or even $\exp(n^\delta)$) stretch, for an arbitrary constant $\eta > 0$. For randomized (non-explicit) constructions, we can get an arbitrary polynomial stretch with $r, k = \Omega(n)$.

- **Robust ϵ-biased PRGs:** We get an explicit construction of a strong (k, q)-robust ϵ-biased PRG with $q = O(1)$, $k = \Omega(n)$, linear stretch, and exponentially small bias. We also get a randomized construction with a small polynomial stretch which satisfies the weaker notion of robustness.

- **Robust cryptographic PRGs:** PRG constructions with constant output and input locality [9, 10, 8, 7] yield (n, q)-robust PRGs with linear stretch and $q = O(1)$. Concretely, any PRG is (n, q)-robust where q is the product of the input and output locality. We show that a cryptographic PRG from [9], which has linear stretch, is $(\Omega(n), q)$-robust with q which is smaller than the input locality (under a similar intractability assumption).

The output locality of the above PRGs (i.e., the number of inputs on which each output depends) is at most polylogarithmic in the seed length, and their circuit size is at most quasilinear in the output length.

As discussed above, robust PRGs can be directly motivated by their usefulness as a more resilient alternative to traditional PRGs. We present several other applications of (strong) robust r-wise independent PRGs in cryptography. The high level idea behind these applications is as follows. Suppose that a cryptographic computation, which has secret inputs w and secret randomness ρ, is attacked by an adversary who can observe intermediate values in the computation. Whenever it is guaranteed that the adversary's view depends only on a small number of bits from ρ (but can arbitrarily depend on w), we can replace the true randomness ρ by pseudorandomness generated using a robust r-wise independent PRG without degrading the security of the implementation. Note that robustness is necessary here because the PRG computation becomes a part of the new implementation and hence it is also subject to attacks. We apply this idea in the following domains.

Private circuits. A *t-private circuit* [35] is a randomized circuit which transforms a randomly encoded input into an encoded output while providing the guarantee that the joint values of any t wires reveal nothing about the input. We show that any t-private circuit in which each wire depends on at most ℓ bits of randomness can be converted into a t private circuit that uses roughly $t\ell$ bits of randomness via the use of robust r-wise PRGs. Applying this to a variant of the construction from [35], we get a general construction of t-private circuits which can use $O(t^3)$ bits of randomness to protect an arbitrary poly(t)-time computation.

Secure multiparty computation. We show a similar application of robust r-wise PRGs in the related context of unconditionally secure multiparty computation. Here we improve on the randomness complexity of a previous randomness-efficient protocol from [18], which implicitly relies on the naïve robust PRG construction described above.

Secure two-party computation. We obtain a constant-round two-party computation protocol secure against malicious parties in which evaluating a circuit of size s with security $2^{-\kappa}$ requires only a polylogarithmic (in κ, s) number of calls to a cryptographic PRG for each gate of the circuit, where κ is a security parameter, and a small number of oblivious transfers. In fact, our protocol is *non-interactive* in a model that allows parallel oblivious transfers. This improves over previous constant-round protocols which combine Yao's garbled circuit construction with a "cut-and-choose" technique (e.g., [39]), where the number of PRG calls per gate is $O(\kappa)$. This also improves over a previous protocol from [33] in which the number of PRG calls is similar to our protocol but the number of oblivious transfers is very large (comparable to the number of PRG calls). The improvement over [33] results from implementing randomized circuits of a near-optimal size which use a small amount of randomness to make any disjunction of circuit wires or their negation essentially independent of the input. For this application, the crucial feature of our robust PRG constructions is their near-optimal *circuit size* rather than seed length. We note that our secure two-party protocol is essentially subsumed by a recent protocol from [30] that does not require robust PRGs. See Section 1.2 below.

1.2 Related Work

It is instructive to view the question we study in the broader context of leakage-resilient cryptography. The general goal in this area is to get an implementation of a cryptographic function (say, a PRG or an encryption scheme) which remains "as secure as possible" in the presence of information leakage. One way of classifying works in this area is according to the following criteria:

- What is the class of leakage functions? One may consider either **(A)** *local leakage,* where the adversary can probe k physical bits in the implementation, or **(B)** *global leakage,* where the adversary can learn arbitrary k bits of information.

- Which parts of the system leak? Here one can consider either **(1)** *confined leakage,* which applies only to part of the implementation (e.g., a secret key, a seed, or an online phase), or **(2)** *unconfined leakage,* where the leakage applies to the entire implementation.

  In case (1) one can hope to offer full protection against leakage, whereas in case (2) one needs to settle for allowing a similar type of leakage in the "ideal model." That is, if arbitrary k bits of information can be leaked, the best we can hope for is that the adversary will learn k bits of information *of the same type* about the secrets.

Most work on leakage resilient cryptography falls either into category (B1) (e.g., [12, 23, 2, 44, 27]), (A1) (e.g., [47, 17, 35]), or (B2) (e.g., [29, 25, 13]). Our work may be the first to study nontrivial questions of type (A2).

We conclude by comparing our notion of robust PRGs with other notions of robustness for PRGs considered in the literature. An *exposure resilient function* (ERF) [17] is a PRG whose output remains pseudorandom even if k physical bits of the *seed* (and the seed alone) are leaked. Thus, ERFs can be classified into category (A1). An ERF can be obtained by applying a standard PRG on top of an extractor for bit-fixing sources [20]. A natural approach for constructing a robust PRG from an ERF is to apply a private circuit compiler (such as [35]) to the ERF. This approach fails because of the high randomness complexity of private circuits. Even if one uses the randomness-efficient private circuits mentioned above, the parameters of the resulting robust PRG will end up being worse than those obtained by taking the exclusive-or of $k+1$ standard PRGs. Finally, the dense model theorem (already mentioned above) implies that *any* sufficiently strong cryptographic PRG offers leakage resilience of type (B2). That is, leaking arbitrary k bits of information about the seed is not much worse than leaking roughly k bits of information about the output.

Subsequent work. A recent secure two-party protocol from [30] only requires a constant number of rounds and a constant number of PRG calls per gate. (This assumes an explicit construction of unbalanced expander graph; without this assumption, one needs to settle for an arbitrarily small super-constant overhead.) The protocol from [30] relies on a different randomization technique that eliminates the need for robust PRGs, at the cost of a slightly larger number of oblivious transfers. The construction of randomness efficient leakage-resilient circuits of near-optimal size, on which our two-party protocol is based, may be of independent interest.

Organization. In Section 2 we give an informal overview of our techniques. In Section 3 we give formal definitions for different variants of robust PRGs. Section 4 describes our constructions of robust PRGs and Section 5 describes applications of robust r-wise PRGs in cryptography.

2 Overview of the Techniques

In this section we sketch the techniques used in our constructions. At a very high level, we achieve robustness in all of our PRGs by constructing *local* PRGs. A local PRG is a PRG such that each output bit depends on only a small number of input bits (say d bits). Therefore, we can implement our PRGs using circuits with small locality (such as NC^0 circuits). In such an implementation, if the value of a wire is leaked to an adversary, it is no worse to assume that instead of the wire, the adversary learns the values of some d input bits. Thus, if k wires are leaked, then it is no worse to assume instead that kd input bits are leaked. We then show that our constructions are robust against the leakage of input bits. We now give more details below.

2.1 Robust r-wise independent PRG

Limited independence has found many applications in computer science, which motivated several efficient constructions of PRGs whose outputs have limited independence. A construction from [42], based on expander graphs, has the small locality property that we need. Specifically, assume that we have a bipartite expander graph with left degree d, such that any subset of $v \leq \ell$ vertices on the left has at least $\frac{3}{4}vd$ neighbors on the right. Such an expander with vertex expansion $> d/2$ is called a *unique neighbor* expander. Now associate the left vertices with the output bits and the right vertices with the input bits. It is well known that the PRG obtained by computing each output bit as the xor of its neighbors is an r_1-wise independent PRG for $r_1 = \ell$.

We show that the above construction is also a robust r-wise independent PRG for some $r = \Omega(r_1)$. Specifically, we show that whenever we fix a subset of vertices S on the right with $|S|$ not too big, there exists a subset of vertices T on the left with $|T|$ not too big, such that the induced graph on vertices outside of $S \cup T$ is still a unique neighbor expander. Therefore, even conditioned on any fixing of the input bits in S, the output bits outside of T are still r-wise independent for some r.

To show this, the rough idea is as follows. We let T be the set of vertices which have a large fraction of neighbors in S. Then $|T|$ is small because otherwise T will not have sufficient expansion. Now for any subset of vertices V on the left with $V \cap T = \emptyset$ and $|V| \leq \ell - |T|$, V must have many neighbors in $\bar{S}$ (the vertices on the right not in S), since by expansion $V \cup T$ has many neighbors, while on the other hand the neighbors of T are fairly concentrated in S. This gives our robust r-wise independent PRG.

Next, we show that the above construction is in fact a strong r-wise independent PRG. The rough idea is as follows. Ideally, we want to show that the output bits outside of T are still r-wise independent even conditioned on any fixing of the input bits in S, and any fixing of the output bits in T. However, this may not be true. If this is not true, then by the XOR lemma there must exist a subset W of output bits with $W \cap T = \emptyset$ and $|W| \leq r$, such that the xor of the bits in W is not uniform. Since all output bits are linear functions of the input bits, this implies that there exists a subset $U \subset T$ such that the xor of the bits in U is always equal to the xor of the bits in W (or always equal to the complement of the xor). This can only happen if the unique neighbors of W form a subset of the union of S and the neighbors of T. Now we can do the following process. Initialize T to be the empty set and if there is a non-empty subset T' of vertices with $T \cap T' = \emptyset$ and $|T'| \leq r$ that has a large fraction of unique neighbors in $S \cup \Gamma(T)$, we let $T = T \cup T'$. We repeat the above process until there is no such T' left. Since the graph is finite the process always terminates in finite steps. We then use techniques similar to those in [4] to show that the size of T is not too big. Then, any subset W outside of T with $|W| \leq r$ will have many unique neighbors outside of $S \cup \Gamma(T)$, and therefore even conditioned on the fixing of the input bits in S and the output bits in T, the xor of the output bits in W is still uniform.

2.2 Robust ϵ-biased PRG

Our robust ϵ-biased PRGs are obtained by xoring the output of our robust r-wise independent PRG with the output of another PRG (which uses an independent seed). We give two constructions. The first is an explicit strong robust PRG with linear stretch, or larger stretch with smaller robustness, and the second is a randomized robust PRG with polynomial stretch. Note that an ϵ-biased PRG must fool a linear test with any size. Our first observation is that the robust r-wise independent PRG described above already fools linear test with size at most r. Thus, the other PRG only needs to fool linear tests with large size.

For the strong robust ϵ-biased PRG, we construct the other PRG as follows. We take a seed with n bits and divide it evenly into n/c blocks, where each block contains c bits, for some constant $c > 1$. Now we use each block as a seed and apply a known construction of ϵ-biased PRG, such as those in [43, 5]. The PRG can stretch c bits into $2^{\Omega(c)}$ bits with bias $2^{-\Omega(c)}$. Now we take the concatenation of the outputs for all blocks as the output. This construction indeed also has small locality as fixing k wires is no worse than fixing k blocks of input bits. Moreover, even conditioned on the fixed input bits and the corresponding output bits, the xor of any $> r$ of the remaining output bits has bias of at most $2^{-\Omega(r)}$. In other words, this PRG is itself a strong robust ϵ-biased PRG for linear tests with large size. Thus when xored with our strong robust r-wise independent PRG, we obtain a strong robust ϵ-biased PRG.

For our randomized robust PRG with polynomial stretch, we take the other PRG as the randomized construction in [8]. Specifically, there the authors showed that one can take a randomized bipartite graph with $n^{1+\delta}$ left vertices, n right vertices and left degree d for some constant $d > 0$, and associate the left and right vertices with output bits and input bits respectively. Now, if each output bit is computed by applying a non-degenerate predicate P to its neighbors, then the graph gives an ϵ-biased PRG with $\epsilon = \exp(-n^{\Omega(1)})$. We show that their construction can be adapted to our case. More specifically, if we again take the randomized graph, and now let each output bit be computed by applying an appropriate polynomial P to its neighbors, then the graph gives a robust ϵ-biased PRG with $\epsilon = \exp(-n^{\Omega(1)})$ for linear tests with large size. Thus when xored with our robust r-wise independent PRG, we obtain a robust ϵ-biased PRG.

2.3 Robust cryptographic PRG

Here we give a high level sketch of our robust cryptographic PRG, under a computational assumption.

The assumption is one made in [3, 9], and the construction is essentially the same construction as the linear stretch cryptographic PRG in [9]. Specifically, take a bipartite expander graph with n right vertices such that any subset of left vertices with size at most ℓ, for $\ell = \omega(\log n)$, has vertex expansion c for some constant $c > 1$. Associate the left and right vertices with output bits and input bits respectively. Compute each output bit as the xor of its neighbors. The assumption is that if we xor the output string with a noise vector such that each bit in the vector is the product of a constant number of independent random bits, then the output is indistinguishable from uniform. Note that the noise vector has small locality and moreover, it is (strongly) robust. Thus, if we take the expander graph to be robust as in our construction of robust r-wise independent PRG, then the resulting construction is a robust cryptographic PRG.

However, this PRG does not give us any stretch, since the number of uniform random bits used to produce the noise vector is larger than the output size. To fix this, the authors in [9] use another fresh random seed to apply an *extractor* to the random bits used to generate the noise vector, and concatenate the output of the extractor to the output of the above PRG. Their extractor is obtained by xoring an ϵ-biased distribution with the the random bits used to generate the noise vector, and the output of the extractor can be shown to be statistically close to being uniform and independent of the output of the PRG. Note that the random bits that are used to generate the noise vector are themselves (strongly) robust. Furthermore, we can show that conditioned on the leakage of a small number of wires and the fixing of the noise vector, these bits still have a lot of entropy. Thus if we take the ϵ-biased distribution to be the output of our robust ϵ-biased PRG, we get a robust cryptographic PRG.

We leave open the question of constructing *strong* robust cryptographic PRGs with good parameters. In particular, such constructions need to resist the type of reconstruction algorithms considered in [16].

On the distinction between robustness and strong robustness. One might think that a robust PRG is always also a strong robust PRG with some modest loss in parameters. This is not the case. The following simple example provides a separation between the two notion in the cases of small bias and cryptographic PRGs. Take a (weakly) robust PRG G and add a dummy gadget computing the xor of all output bits.

The resulting PRG is still weakly robust with $k = 1$: if we let S include any single wire of the dummy gadget and T be one of the output bits on which C_S depends, then $Y_{\bar{T}}$ is indistinguishable from uniform conditioned on C_S. However, letting S contain the output of the dummy gadget, there is no output bit T such that $Y_{\bar{T}}$ conditioned on S and T is pseudorandom.

3 Definitions

In this section we define the different notions of robust PRGs we will be interested in. First, U_n denotes the uniform distribution on n bits; if n is understood we sometimes use U. We will need the following notion of "fooling" with respect to functions of varying input lengths.

Definition 1. *Let $\mathcal{F} = \bigcup_n \mathcal{F}_n$ be a class of functions, where the functions in $\mathcal{F}_n$ are from $\{0,1\}^n$ to $\{0,1\}$. A probability distribution D on $\{0,1\}^n$ is said to ϵ-*fool *$\mathcal{F}$ if for any $f \in \mathcal{F}_n$,*

$$|\Pr[f(D) = 1] - \Pr[f(U) = 1]| \leq \epsilon.$$

Definition 2. *A circuit implementation C of a function $G : \{0,1\}^n \to \{0,1\}^m$ is a (k,q)-robust pesudorandom generator (PRG) for a class $\mathcal{F}$ of functions with error ϵ if the following holds. Let X be the uniform distribution over $\{0,1\}^n$ and $Y = G(X)$. For any set S of at most k wires in G, there is a set T of at most $q|S|$ output bits such that conditioned on any fixing of the values C_S of the wires in S, the values $Y_{\bar{T}}$ of the output bits not in T ϵ-fools $\mathcal{F}$. We say that G is a* strong *(k,q)-robust PRG for $\mathcal{F}$ if conditioned on any fixing of the values C_S and Y_T, we have that $Y_{\bar{T}}$ ϵ-fools $\mathcal{F}$.*

If the implementation is understood or unimportant, we may simply say that a function is a robust PRG. This may happen if each output bit depends on only few input bits, and any implementation that does not involve using extraneous bits is robust.

When each $\mathcal{F}_n$ consists of all tests on r bits, we call a robust PRG for $\mathcal{F}$ with 0 error a robust r-wise independent PRG. When each $\mathcal{F}_n$ consists of all parities on subsets of the n bits, we call a robust PRG for $\mathcal{F}$ a robust ϵ-biased PRG. A robust *cryptographic* PRG is one which is robust for each $\mathcal{F}$ that can be computed by circuits of poly(n) size with negligible error $\epsilon(n)$.

We handle leakage of arbitrary wire values by constructing a *local* PRG which can handle leakage of inputs. In particular, we have the following definitions and simple lemma.

Definition 3. *A function $f : \{0,1\}^n \to \{0,1\}^m$ is d-local if each output bit depends on at most d input bits.*

Definition 4. *A function $G : \{0,1\}^n \to \{0,1\}^m$ is a (k,q)-*input *robust PRG for a class $\mathcal{F}$ of functions with error ϵ if the following holds. Let X be the uniform distribution over $\{0,1\}^n$ and $Y = G(X)$. For any set S of at most k input bits, there is a set T of at most $q|S|$ output bits such that conditioned on any fixing of the values X_S of the inputs S, the values $Y_{\bar{T}}$ of the output bits not in T ϵ-fools $\mathcal{F}$. We say that G is a strong (k,q)-input robust PRG for $\mathcal{F}$ if conditioned on any fixing of the values X_S and Y_T, we have that $Y_{\bar{T}}$ ϵ-fools $\mathcal{F}$.*

Lemma 1. *A d-local (strong) (dk,q)-input robust PRG is a (strong) (k,dq)-robust PRG with the same error.*

4 Constructions

In this section, we describe our constructions of robust r-wise PRGs.

Our construction of a robust r-wise independent PRG is simple. A bipartite graph $H = ([m],[n],E)$ induces a function $G_H : \{0,1\}^n \to \{0,1\}^m$ where the ith output bit is the parity of the input bits corresponding to the neighbors of i. That is, $G_H(x_1,\ldots,x_n) = (y_1,\ldots,y_m)$ where $y_i = \oplus_{j\in\Gamma(i)} x_j$.

We will take H to be a bipartite expander with expansion bigger than half the degree. This is defined as follows.

Definition 5. *A bipartite graph* $([m],[n],E)$ *with left vertices* $[m]$ *and right vertices* $[n]$ *is an* (ℓ, b)-expander *if for any subset* $V \subseteq [m]$ *on the left with* $|V| \le \ell$*, we have that* $|\Gamma(V)| \ge b|V|$*.*

We can now state our theorem.

Theorem 2. *Suppose* H *is a* d*-left-regular* $(\ell, (1/2+\gamma)d)$*-expander. Then for any constant* $0 < \alpha < 1$*, we have that* G_H *is a strong* $(\alpha\gamma\ell, 1/\gamma)$*-robust* r*-wise independent PRG, with* $r = (1-\alpha)\ell$*.*

One surprising feature of this theorem is that the degree d of the expander does not appear anywhere explicitly. Besides expansion bigger than $d/2$, the important parameter of the expander is ℓ, the maximum size of subsets which expand. The parameter ℓ determines the robustness of the PRG. We have $\ell \le 2n/d$, so the degree appears implicitly there.

First we state a result that a random d-left-regular graph H is a good expander with high probability.[1]

Theorem 3. *For any constant* $0 < \gamma < 1/2$ *there exists a constant* $0 < \alpha < 1$ *such that for any* $m \le nd/\alpha^{d-1}$*, with probability* $1 - n^{-\Omega(d)}$*, a random* d*-left-regular graph* H *with* m *left vertices and* n *right vertices is an* $(\Omega(n/d), (1/2+\gamma)d)$*-expander. Alternatively, for any* $\eta > 0$ *the following holds. For any* $m \le \alpha n^{1-\eta}(\alpha n^{\eta})^{(1/2-\gamma)d}/d$*, with probability* $1 - n^{-\Omega(d)}$*, a random* d*-left-regular graph* H *with* m *left vertices and* n *right vertices is an* $(n^{1-\eta}/d, (1/2+\gamma)d)$*-expander.*

Proof. This is a direct application of the probabilistic method. Choose some $k = \Omega(n/d)$, we just need to bound the following probability.

$$\sum_{j=2}^{k} \binom{m}{j} \binom{n}{(\frac{1}{2}+\gamma)jd} \left(\frac{(\frac{1}{2}+\gamma)jd}{n}\right)^{jd}.$$

Note that for every fixed j, the summand is at most

$$\left(\frac{em}{j}\right)^{j} \left(\frac{en}{(\frac{1}{2}+\gamma)jd}\right)^{(\frac{1}{2}+\gamma)jd} \left(\frac{(\frac{1}{2}+\gamma)jd}{n}\right)^{jd}$$
$$\le \left(\frac{em}{j}\left(\frac{e^c(\frac{1}{2}+\gamma)jd}{n}\right)^{(\frac{1}{2}-\gamma)d}\right)^{j},$$

where $c = \frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}$. We can now choose $k = \Omega(n/d)$ with a small enough constant in $\Omega()$, so that $\frac{em}{j}\left(\frac{e^c(\frac{1}{2}+\gamma)jd}{n}\right)^{(\frac{1}{2}-\gamma)d} < 1/4$. This will guarantee that the whole probability is less than 1. Thus we get

$$m < \frac{j}{4e}\left(\frac{n}{e^c(\frac{1}{2}+\gamma)jd}\right)^{(\frac{1}{2}-\gamma)d}.$$

[1] In an earlier version we mistakenly state that a random d-left-regular graph with $m = n^d$ is an $(\Omega(n/d), (1/2+\gamma)d)$-expander, which is not true. To achieve such expansion m can only be as large as nd/α^{d-1}, or to achieve output length $m = n^{\Omega(d)}$ one can only guarantee the expansion of subsets up to size $n^{1-\eta}/d$ for an arbitrary $\eta > 0$.

With $k = \Omega(n/d)$ and $d \leq \sqrt{n}$ the minimum of the above quantity happens at $j = k$ and thus we only need to ensure that $m \leq \alpha nd/\alpha^d = nd/\alpha^{d-1}$ for some constant $0 < \alpha < 1$. The largest term in the sum on the other hand happens at $j = 2$ which contributes $n^{-\Omega(d)}$ to the error.

Alternatively, if we just need the expansion property for all subsets up to $k = n^{1-\eta}/d$ for some constant $\eta > 0$, then again the minimum of the above quantity happens at $j = k$ and thus we only need to ensure that $m \leq \alpha n^{1-\eta}(\alpha n^{\eta})^{(1/2-\gamma)d}/d$. The largest term in the sum on the other hand happens at $j = 2$ which contributes $n^{-\Omega(d)}$ to the error. □

The above theorem gives the following non-explicit robust r-wise independent PRG.

Theorem 4. *There exist constants $0 < \alpha, \beta < 1$ such that for any $d \leq \sqrt{n}$, for a random d-left-regular H with $m \leq nd/\alpha^{d-1}$, with probability $1 - n^{-\Omega(d)}$, the function $G_H : \{0,1\}^n \to \{0,1\}^m$ is a d-local strong $(\beta n/d, 21)$-robust r-wise independent PRG for $r = \beta n/d$. Alternatively, for $\eta > 0$ the following holds. For any $m \leq \alpha n^{1-\eta}(\alpha n^{\eta})^{\beta d}/d$, with probability $1 - n^{-\Omega(d)}$, the function $G_H : \{0,1\}^n \to \{0,1\}^m$ is a d-local strong $(n^{1-\eta}/d, 21)$-robust r-wise independent PRG for $r = n^{1-\eta}/d$.*

We now instantiate Theorem 2 with known expander constructions. Capalbo et al. [19] achieved expansion bigger than $d/2$ for constant degree graphs, with $\ell = \Omega(n/d)$. This yields:

Theorem 5. *For any constant $C > 0$ there is a constant $\beta > 0$ such that there is an explicit $O(1)$-local strong $(\beta n, 21)$-robust r-wise independent PRG $G : \{0,1\}^n \to \{0,1\}^{Cn}$ for $r = \beta n$.*

For larger stretch but lower robustness we use the expanders of Guruswami, Umans, and Vadhan [28]. They achieve $\ell = n^{1-\eta}$ for any $\eta > 0$, which gives:

Theorem 6. *For any $\eta > 0$ there exists $\delta, C > 0$ such that for any $m \leq \exp(n^{\delta})$, there is an explicit d-local strong $(n^{1-\eta}, 21)$-robust r-wise independent PRG $G : \{0,1\}^n \to \{0,1\}^m$ for $r = n^{1-\eta}$ and $d \leq \log^C m$.*

To prove the theorem we first prove a lemma about expanders.

Lemma 7. *Suppose $H = ([m], [n], E)$ is a d-left-regular $(\ell, d/2+c)$-expander. Then for any $S \subseteq [n]$ on the right of size $|S| \leq c\ell/2$, there exists $T \subseteq [m]$ on the left with $|T| \leq 2|S|/c$ such that the induced graph on left vertices $[m] \setminus T$ and right vertices $[n] \setminus S$ is an $(\ell - |T|, d/2 + c/2)$-expander.*

Proof. Let T be the subset on the left that has the maximum size among all subsets $\{T'\}$ on the left with size at most ℓ such that $|\Gamma(T') \setminus S| \leq (d/2 + c/2)|T'|$ (note that T is well defined since $T' = \phi$ is such a subset). Then we must have $|T| \leq 2|S|/c$ because otherwise we have

$$|\Gamma(T)| \leq |\Gamma(T) \setminus S| + |S| \leq (d/2 + c/2)|T| + |S| < (d/2 + c)|T|,$$

which contradicts the expansion property of H.

Next, for any non-empty subset V on the left with $V \cap T = \phi$ and $|V| \leq \ell - |T|$, we have

$$|\Gamma(V) \setminus S| + |\Gamma(T) \setminus S| \geq |\Gamma(V \cup T) \setminus S| \geq (d/2 + c/2)|V \cup T| = (d/2 + c/2)(|V| + |T|),$$

since V and T are disjoint and $|V| + |T| \leq \ell$.

Thus we must have

$$|\Gamma(V) \setminus S| \geq (d/2 + c/2)|V|$$

since $|\Gamma(T) \setminus S| \leq (d/2 + c/2)|T|$. □

Note that this theorem already shows that our construction is a (weak) robust r-wise independent PRG for some r. This is because we can view S in the lemma as the set of input bits that are potentially leaked to the adversary, and the lemma says that there exists a subset T of the output bits such that the induced graph of H outside of S and T is still a unique-neighbor expander. Therefore, the output bits outside of T are still r-wise independent for some r. In the following we prove that the same construction is actually strongly robust, proving Theorem 2.

We show that G_H is strongly $(d\alpha\gamma\ell, 1/(\gamma d))$-input-robust. Let S be the set of input bits that are fixed with $|S| = s \leq kd \leq \alpha\gamma d\ell = \alpha c\ell$.

We now consider the input bits in $\bar{S}$, which are the bits not in S. Let $x \in \{0,1\}^n$ be the input string, and $y = G(x) \in \{0,1\}^m$ be the output string. For any output bit y_i we associate with it a vector $V_i \in \{0,1\}^n$. The vector V_i has exactly d 1s at the d positions of y_i's neighbors (more precisely, y_i's corresponding vertex's neighbors), and has 0s everywhere else, i.e., V_i is the indicator vector of whether an x_j influences y_i. We then let $\bar{V}_i \in \{0,1\}^{n-s}$ be the vector that is obtained by projecting V_i into the bits that are in $\bar{S}$. Let X be the uniform distribution over $\{0,1\}^n$, and $Y = G(X)$ be the output bits of the PRG. The following fact is immediate.

Fact 8. *For any subset W of the output bits, we have*

$$\bigoplus_{Y_i \in W} Y_i \text{ is a constant } \iff \sum_{Y_i \in W} \bar{V}_i = 0.$$

Lemma 9. *For any subset W of the output bits, let $\bar{W}$ be the output bits not in W. Assume that conditioned on some fixing of the input bits $\{X_i \in S\}$ and the output bits $\{Y_h \in W\}$, there exist some bits $Y_{j1}, \cdots, Y_{jl} \in \bar{W}$ such that $Y' = \bigoplus_{i=1}^{l} Y_{ji}$ is not uniform. Then Y' is a constant. Moreover, $\bar{V}' = \sum_{i=1}^{l} \bar{V}_{ji}$ is in $\mathsf{Span}(\{\bar{V}_h : Y_h \in W\})$, and vice versa.*

Proof. First, note that if $\bar{V}' = \sum_{i=1}^{l} \bar{V}_{ji}$ is in $\mathsf{Span}(\{\bar{V}_h : Y_h \in W\})$ then $Y' = \bigoplus_{i=1}^{l} Y_{ji}$ is the sum of some Y_h's in W and some X_i's in S. Thus Y' is a constant.

We now prove the other direction. For each X_i we associate with it a vector $U_i \in \{0,1\}^n$ that has exactly one 1 at the i'th position, and has 0's everywhere else. Let b be the dimension of $\mathsf{Span}(\{\bar{V}_h : Y_h \in W\})$. Since X is originally the uniform distribution, and every fixing of X_i or Y_h is a linear constraint, we have that conditioned on all these fixings, X is now an affine source of dimension $n - s - b$. In other words, there exist $n - s - b$ linearly independent vectors $A_1, \cdots, A_{n-s-b} \in \{0,1\}^n$ and another vector $A_0 \in \{0,1\}^n$ such that

$$X = \sum_{i=1}^{n-s-b} Z_i A_i + A_0,$$

where $\{Z_i\}$ are uniform independent random bits. Moreover, each A_i is orthogonal to each $(U_j : X_j \in S)$ and each $(V_h : Y_h \in W)$. Let $V' = \sum_{i=1}^{l} V_{ji}$. Then

$$Y' = \bigoplus_{i=1}^{l} Y_{ji} = \langle V', X \rangle = \sum_{i=1}^{n-s-b} Z_i \langle V', A_i \rangle + \langle V', A_0 \rangle.$$

Thus, if Y' is not uniform, then for all i, we have $\langle V', A_i \rangle = 0$ and $Y' = \langle V', A_0 \rangle$ is a constant. Thus $V' \in \mathsf{Span}(\{U_i : X_i \in S\}, \{V_h : Y_h \in W\})$. Note that the X_i's only have 0 outside of the bit positions in S, thus when projected into the bit positions in $\bar{S}$, we have that $\bar{V}' = \sum_{i=1}^{l} \bar{V}_{ji} \in \mathsf{Span}(\{\bar{V}_h : Y_h \in W\})$. □

We now slightly abuse notation and use S to also denote the set of right vertices in H that correspond to the fixed bits. Below we borrow some techniques from [4]. We have the following definition.

Definition 6. *Suppose $H = ([m], [n], E)$ is a d-left-regular $(\ell, d/2 + c)$-expander where $c = \gamma d$. For any subset $T \subset [m]$ we let $\Delta(T) \subset [n]$ be the set of unique neighbors of T, i.e. the set of right vertices that are adjacent to only one vertex in T. For any subset $S \subset [n]$ with $|S| < c\ell$, we define an inference relation $\vdash_S$ on subsets of the left vertices as follows.*

$$T_1 \vdash_S T_2 \Longleftrightarrow |T_2| \leq \ell - |S|/c \wedge |\Delta(T_2) \setminus [\Gamma(T_1) \cup S]| < c|T_2|.$$

We now set $T = \emptyset$ and repeat the following step as long as it is possible: if there exists a non-empty subset $T_1 \subset [m] \setminus T$ such that $T \vdash_S T_1$ then let $T = T \cup T_1$. Since the graph is finite the above procedure terminates in finite steps. We denote the final T by $Cl(S)$. We now have the following lemma.

Lemma 10. *If $|S| < c\ell$ then $|Cl(S)| \leq |S|/c$.*

Proof. Assume for the sake of contradiction that $|Cl(S)| > |S|/c$. Consider the sequence of subsets of left vertices $T_1, T_2, \cdots, T_v$ that we add to the set T. Note that all these T_i's are disjoint. Let $C_v = \cup_{i=1}^{v} T_i$ be the set of left vertices derived in v steps. Thus $|C_v| = \sum_{i=1}^{v} |T_i|$.

Let v_0 be the first v such that $|C_v| > |S|/c$. Thus $|C_{v_0-1}| \leq |S|/c$ and $|C_{v_0}| \leq |C_{v_0-1}| + |T_{v_0}| \leq \ell$. By the expansion property, C_{v_0} has at least $(d/2 + c)|C_{v_0}|$ neighbors and thus $\Delta(C_{v_0}) \geq 2(d/2 + c)|C_{v_0}| - d|C_{v_0}| = 2c|C_{v_0}|$. Therefore

$$|\Delta(C_{v_0}) \setminus S| \geq 2c|C_{v_0}| - |S| > c|C_{v_0}|.$$

On the other hand, since each time when we add T_i to T, the number of unique neighbors in $\Delta(T) \setminus S$ increases by at most $|\Delta(T_i) \setminus [\Gamma(T) \cup S]\,|$, we have

$$|\Delta(C_{v_0}) \setminus S| < \sum_{i=1}^{v_0} c|T_i| = c|C_{v_0}|,$$

which is a contradiction. □

Lemma 11. *Let $T = Cl(S)$. Then conditioned on any fixing of the input bits in S and any fixing of the output bits in T, the output bits that are not in T are r-wise independent, where $r = \ell - |S|/c \geq (1 - \alpha)\ell$.*

Proof. Let $\bar{T}$ be the set of output bits that are not in T. Assume that the lemma is not true. Then, for some fixing of the input bits in S and some fixing of the output bits in T, there exist $1 \leq l \leq r = \ell - |S|/c$ output bits $\{Y_{j1}, \cdots, Y_{jl} \in \bar{T}\}$ such that $\bigoplus_{i=1}^{l} Y_{ji}$ is not uniform.

By Lemma 9, $\bar{V}' = \sum_{i=1}^{l} \bar{V}_{ji}$ is in $\mathsf{Span}(\{\bar{V}_h : Y_h \in T\})$. Let T_1 be the set of left vertices corresponding to $\{Y_{j1}, \cdots, Y_{jl}\}$. Then we have that $\Delta(T_1) \setminus S \subset \Gamma(T)$. Thus $|\Delta(T_1) \setminus (\Gamma(T) \cup S)| = 0 < c|T_1|$. This means that we can add T_1 to T in the procedure where we obtain $Cl(S)$, which contradicts the fact that $Cl(S)$ is obtained when the procedure stops. □

Note that $|T| \leq |S|/c \leq kd/c = k/\gamma$, thus the theorem is proved.

4.1 Robust ϵ-biased generators

We give two constructions of robust ϵ-biased PRGs. The first construction is strong and explicit. It gives linear stretch for linear robustness and bigger stretch for smaller robustness. The second PRG outputs close to $n^{3/2}$ bits even for linear robustness. However, it is not explicit (because we don't have an explicit construction of the underlying expander) or strong.

Both constructions start from the observation that a strong robust r-wise independent PRG can handle parities of size at most r. We will xor this PRG with another PRG that handles all larger parities. For robust PRGs, we combine them with the following simple lemma.

Lemma 12. *Suppose $G_1 : \{0,1\}^{n_1} \to \{0,1\}^m$ is a (k, q_1)-robust PRG for parities of size at most r with error ϵ and $G_2 : \{0,1\}^{n_2} \to \{0,1\}^m$ is a (k, q_2)-robust PRG for parities of size more than r with error ϵ. Then $G : \{0,1\}^{n_1+n_2} \to \{0,1\}^m$ given by $G(X) = G(X_1) \oplus G(X_2)$ is a $(k, q_1 + q_2)$-robust ϵ-biased PRG.*

For strong robust PRGs, we need the following definition.

Definition 7. *A circuit implementation C of a function $G : \{0,1\}^n \to \{0,1\}^m$ is a doubly strong (k, q, t)-robust pesudorandom generator (PRG) for a class $\mathcal{F}$ of functions with error ϵ if the following holds. Let X be the uniform distribution over $\{0,1\}^n$ and $Y = G(X)$. For any set S of at most k wires in G, there is a set T of at most $q|S|$ output bits such that conditioned on any fixing of the values C_S of the wires and the values of Y_T, and the further fixing of t arbitrary output bits in $\bar{T}$, we have that the rest of the bits in Y ϵ-fools $\mathcal{F}$.*

For example, we have the following lemma.

Lemma 13. *A strong (k, q)-robust r-wise independent PRG is also a doubly strong (k, q, t)-robust $(r - t)$-wise independent PRG for any $t < r$.*

Now, if we take the xor of two doubly strong robust PRGs, then we get a strong robust PRG. Specifically, we have

Lemma 14. *Suppose $G_1 : \{0,1\}^{n_1} \to \{0,1\}^m$ is a doubly strong (k, q_1, kq_2)-robust PRG for parities of size at most r with error ϵ and $G_2 : \{0,1\}^{n_2} \to \{0,1\}^m$ is a doubly strong (k, q_2, kq_1)-robust PRG for parities of size more than r with error ϵ. Then $G : \{0,1\}^{n_1+n_2} \to \{0,1\}^m$ given by $G(X) = G(X_1) \oplus G(X_2)$ is a strong $(k, q_1 + q_2)$-robust ϵ-biased PRG.*

We will set $n_1 = n_2 = n/2$ in both constructions.

4.1.1 First Construction

To handle larger parities, our first construction uses the following well-known constructions of ϵ-biased spaces. Recall that a probability space is ϵ-biased if, for any X the parity of a non-empty subset of variables, $|\Pr[X = 1] - \Pr[X = 0]| \le \epsilon$, i.e., the deviation from $1/2$ is at most $\epsilon/2$.

Theorem 15. *[43, 5] For any $\epsilon > 0$ there is an explicit construction of PRG $G_0 : \{0,1\}^n \to \{0,1\}^m$ such that the output is ϵ-biased and $n \le 3(\log m + \log(1/\epsilon))$.*

Define $G_{u,d} : \{0,1\}^{ud} \to \{0,1\}^m$ by $G_{u,d}(x_1, \ldots, x_u) = G_0(x_1) \circ \ldots \circ G_0(x_u)$, where each $x_i \in \{0,1\}^d$ and $\circ$ denotes concatenation. Here $G_0 : \{0,1\}^d \to \{0,1\}^{m_0}$ has error $\epsilon_0 \le 2^{-d/6}$ and output length $m_0 = 2^{d/6}$, so $m = um_0 = 2^{d/6}u$.

Now we have the following lemma.

Lemma 16. *For any $k, t \in \mathbb{N}$ with $k + t \leq u$ and any $r > tm_0$, the function $G_{u,d}$ defined above is a d-local doubly strong (k, m_0, t)-robust PRG for parities of size at least r with error $\epsilon \leq \epsilon_0^{r/m_0 - t}$.*

Proof. For any fixing of $s \leq k$ wires S, let T be the set of at most sm_0 output bits generated by the s or fewer input blocks x_i which lead to wires in S. Let T_1 be the set of the other fixed t output bits. Let T_2 be the set of at most tm_0 output bits generated by the t or fewer input blocks x_i which lead to the bits in T_1. Consider any fixing of $T \cup T_1$. Now we analyze any parity on a set R of size at least r not involving the bits of $T \cup T_1$. Some of the bits in R may be in T_2. However, the number of bits in R that are not in T_2 is at least $r - |T_2| = r - tm_0$. These bits depend on at least $v = (r - tm_0)/m_0 = r/m_0 - t$ input blocks x_i, which are independent of the bits in $T \cup T_2$. Partition these bits into $R_1 \cup \ldots \cup R_v$ accordingly. Then each parity R_i has bias (deviation from $1/2$) at most $\epsilon_0/2$, so the bias of v independent such parities has bias at most $\epsilon_0^v/2$. Since the R_i's are independent of the bits in T_2, the parity on R also has bias at most $\epsilon_0^v/2$. □

Theorem 17. *For a small enough constant $\beta > 0$, for any constant $C > 1$ there is an explicit strong $(\beta n, O(1))$-robust ϵ-biased PRG $G : \{0,1\}^n \to \{0,1\}^{Cn}$, where $\epsilon = 2^{-\Omega(n/C)}$.*

Proof. We let $G_2 : \{0,1\}^{n/2} \to \{0,1\}^{Cn}$ be $G_{u,d}$ with $2^{d/6}/d = 2C$ and $u = n/(2d)$. Thus $m_0 = 2^{d/6}$ is a constant. We let $G_1 : \{0,1\}^{n/2} \to \{0,1\}^{Cn}$ be the PRG from Theorem 2 and choose α, γ such that G_1 is a strong $(\beta n, 21)$-robust r-wise independent PRG for $r = \tau n$ with $\tau > 25 m_0 \beta$ (since we only need linear stretch we can achieve $\ell = \Omega(n)$ in that theorem). Thus by Lemma 13 G_1 is also a doubly strong $(\beta n, 21, m_0 \beta n)$-robust r'-wise independent PRG for $r' = \tau n - m_0 \beta n$. By Lemma 20 G_2 is a doubly strong $(\beta n, m_0, 21\beta n)$-robust PRG for parities of size at least r' with error

$$\epsilon \leq \epsilon_0^{r'/m_0 - 21\beta n} = \epsilon_0^{\tau n/m_0 - 22\beta n} \leq (2^{-d/6})^{(3\tau n)/(25 m_0)} = (2^{-d/6})^{(3\tau n)/(25 \cdot 2^{d/6})} = 2^{-\Omega(n/C)}.$$

Thus by Lemma 14, G is a strong $(\beta n, m_0 + 21 = O(1))$-robust ϵ-biased PRG with $\epsilon = 2^{-\Omega(n/C)}$. □

We can also achieve larger stretch with smaller robustness.

Theorem 18. *For any $\eta > 0$ and any $s = s(n) < n^{1-\eta}$, there is an explicit strong $(n^{1-\eta}/s, s)$-robust ϵ-biased PRG $G : \{0,1\}^n \to \{0,1\}^{sn}$, where $\epsilon = \exp(-n^{1-\eta}/s)$.*

Proof. We repeat the above proof and let $G_1 : \{0,1\}^{n/2} \to \{0,1\}^{sn}$ be the PRG from Theorem 2 with the expander graph in [28] that achieves $\ell = n^{1-\eta}$, and $G_2 : \{0,1\}^{n/2} \to \{0,1\}^{sn}$ be $G_{u,d}$ with $2^{d/6}/d = 2s$ and $u = n/(2d)$. □

4.1.2 Second Construction

Next, we construct a robust ϵ-biased PRG $G : \{0,1\}^n \to \{0,1\}^m$ with $m = n^{1+\delta}$ for some constant $0 < \delta < 1$. Some of the techniques are borrowed from Applebaum et al [8]. Here we will apply Lemma 12. We now describe G_1 and G_2.

Let H be a d-left-regular $(\ell, 11d/12)$-expander with $m = n^{1+\delta}$ left vertices, $n/2$ right vertices with $d > 8, \ell > 2$, and every vertex on the right has degree at most $4dn^{\delta}$. Associate the right vertices with the $n/2$ input bits, and the left vertices with the m output bits. For G_1, each output bit is computed as the xor of its neighbors, i.e., $G_1 = G_H$. For G_2, each output bit is a function P of its neighbors, where P is defined as

$$P(x_1, \cdots, x_d) = \sum_{S \subset [d], |S| = d/6+1} \Pi_{i \in S} x_i.$$

Here all operations are in $\mathbb{F}_2$. Since the graph has left degree d, we can implement the PRG naturally as a circuit G with locality d. We now view each left vertex in the bipartite graph as a hyperedge that contains d vertices (i.e., its neighbors). First we have the following definition.

Definition 8. *[8] ((r, l, b)-**independence**). Let $\mathcal{W}$ be a collection of distinct hyperedges. A subset $\mathcal{V} \subseteq \mathcal{W}$ of l distinct hyperedges is an (l, b)-independent set of $\mathcal{W}$ if the following two properties hold.*

1. *Every pair of hyperedges $(V_i, V_j) \in \mathcal{V}$ has distance at least 2, namely, for every pair $V_i \neq V_j \in \mathcal{V}$ and $W \in \mathcal{W}$,*
$$V_i \cap W = \phi \text{ or } V_j \cap W = \phi.$$
2. *For every $V_i \in \mathcal{V}$ and $W \neq V_i$ in $\mathcal{W}$ we have*
$$|V_i \cap W| \leq b.$$

A graph is (r, l, b)-independent if every set of hyperedges with size at least r has an (l, b)-independent set.

We have the following lemma.

Lemma 19. *The bipartite graph described above is $(r, r/(16d^4n^{2\delta}), d/6)$ independent.*

Proof. By the expansion property, every two vertices on the left have at least $11d/6$ neighbors on the right. Thus the number of common neighbors is at most $d/6$. In other words, the intersection of any two different hyperedges has size at most $d/6$. Thus the second property holds. Now fix any set of r hyperedges. We greedily pick hyperedges into the independent set. Specifically, we insert a hyperedge V into the independent set and remove all the hyperedges W that share some common node with V, and all the hyperedges that share some common node with W. We then iterate the above process. Since each time we remove at most $(4d^2n^\delta)^2$ hyperedges, we are left with at least $r/(16d^4n^{2\delta})$ hyperedges in the independent set. □

We now have the following lemma.

Lemma 20. *Consider the PRG G_2. For any subset S of the input bits with size $s \leq 5d\ell/24$, there exists a subset T of the output bits with size $t \leq 24s/(5d)$ such that the following holds. Conditioned on any fixing of the bits in S, the xor of any r output bits in $\bar{T}$ has bias of at most $2^{-\Omega(r/(2^{d/6}d^4n^{2\delta}))}$.*

Proof. Let T be the set of output bits which correspond to vertices that have at least $17d/24$ neighbors in S. By Lemma 7 (note that $c = 5d/12$) we have $|T| \leq 2s/c = 24s/(5d)$. Now any vertex in $\bar{T}$ has at least $d - 17d/24 = 7d/24$ neighbors in $\bar{S}$.

Consider the xor of any r output bits in $\bar{T}$. These bits correspond to r hyperedges $(W_1, \cdots, W_r)$. By Lemma 19 there exists an (l, b) independent set $(V_1, \cdots, V_l)$ with $l = r/(16d^4n^{2\delta})$ and $b = d/6$. Now fix an arbitrary assignment for the input bits in $\bar{S}$ that are not in any of the V_i's and choose the other input bits in $\bar{S}$ uniformly randomly. Let σ be the assignment of all fixed input bits (including those in S). Now the xor of the r bits is

$$Y = \sum_{i \in [r]} P(X_{W_i}) = \sum_{i \in [l]} Z_i(X_{V_i}),$$

where the sum is over $\mathbb{F}_2$ and

$$Z_i(X_{V_i}) = P(X_{V_i}) + \sum_{W : W \neq V_i, W \cap V_i \neq \phi} P(X_{W \cap V_i}, \sigma_{W \setminus V_i}).$$

Note that the Z_i's are independent, since by the property of the independent set $(V_i, \{W : W \neq V_i, W \cap V_i \neq \phi\})_{i \in [l]}$ is a partition of $(W_1, \cdots, W_r)$. Next, note that for $W \neq V_i$, we have $|V_i \cap W| \leq b$. Thus $P(X_{W \cap V_i}, \sigma_{W \setminus V_i})$ is a polynomial of the bits of V_i with degree at most $b = d/6$. On the other hand, since V_i has at least $7d/24 > d/6 + 1$ bits in $\bar{S}$, we have that $P(X_{V_i})$ is a polynomial of the bits of V_i with degree $d/6+1$. Thus, Z_i is a non-constant polynomial with degree $d/6+1$. Therefore, Z_i takes each value of $\{0,1\}$ with probability at least $2^{-(d/6+1)}$. Since Y is the sum of l independent Z_i's, we have that Y has bias of at most $2^{-\Omega(l/2^{d/6})} = 2^{-\Omega(r/(2^{d/6}d^4n^{2\delta}))}$. □

We now have the following theorem.

Theorem 21. *For any constant $0 < \alpha < 1/2$, the circuit G described above is a robust ϵ-biased $(5\alpha\ell/12, 48/5)$ PRG, where $\epsilon = 2^{-\Omega(\ell/(2^{d/6}d^4n^{2\delta}))}$.*

Proof. Note that in the expander graph H we have $\gamma = c/d = 5/12$. By Theorem 2, G_1 is a $(5\alpha\ell/12, 24/5)$ robust r-wise independent PRG with $r = (1-2\alpha)\ell$. By Lemma 20, G_2 is a $(5\alpha\ell/12, 24/5)$-robust PRG for parities of size more than r with error $2^{-\Omega(r/(2^{d/6}d^4n^{2\delta}))}$. Thus by Lemma 12, G is a a robust ϵ-biased $(5\alpha\ell/12, 48/5)$ PRG, where $\epsilon = 2^{-\Omega(\ell/(2^{d/6}d^4n^{2\delta}))}$. □

When d is a constant, a random bipartite graph with the above size satisfies the properties with high probability. Thus we have the following theorem.

Theorem 22. *There exists a constant integer d and a constant $\beta > 0$ such that for any $0 < \delta < 1/2$, given a random d-left-regular H with $m = n^{1+\delta}$, with probability $1 - n^{-\Omega(1)}$, the construction described above using H as the expander is a $(\beta n, 48/5)$-robust ϵ-baised PRG with $\epsilon = 2^{-n^{\Omega(1)}}$.*

4.2 Robust cryptographic generators

In this section we show that a variant of an expander-based construction of cryptographic PRGs from [9], which has constant locality and linear stretch, is also robust under a similar intractability assumption. We now describe the construction.

Let $m = m(n) > n$ be a parameter. Let d_0, d_1, d_2 be three integer constants. Take an $(\ell_0, 3d_0/4)$ expander with left vertices $[m]$, right vertices $[n]$, left degree d_0 with $m = c_1 n$ for some constant $c_1 > 1$ and let $M = M_n$ be the adjacency matrix of the graph. Let X be the uniform distribution over n bits. Let $\mu = 2^{-l}$ for some constant $l \in \mathbb{N}$. We sample a noise vector of length m such that each bit of the vector is 1 with probability μ. Namely, let Y be the uniform distribution over $l \cdot m$ bits. We compute

$$E(Y) = \left(\Pi_{j=1}^{l} Y_{l(i-1)+j}\right)_{i=1}^{m}.$$

Now take a constant $c_2 > 1$ and let $g : \{0,1\}^{lm/c_2} \to \{0,1\}^{lm}$ be the ϵ-biased generator in Theorem 17, where the expander graph in that construction is a $(\ell_1, 3d_1/4)$ expander with left degree d_1 and the block size of the second PRG in that construction is d_2. Since we only need a linear stretch, we can achieve constant degrees d_0, d_1 and $\ell_0 = \Omega(n), \ell_1 = \Omega(lm/c_2) = \Omega(n)$. Let R be the uniform distribution over lm/c_2 bits. Our generator is now defined as

$$G(X, Y, R) = (M_n X + E(Y), g(R) + Y),$$

where all operations are in $\mathbb{F}_2$. Thus $G : \{0,1\}^{n+lm+lm/c_2} \to \{0,1\}^{m+lm}$.

We will use the following variant of an assumption from [3, 9].

Assumption 23. *[3, 9] The assumption has a parameter $c \geq 1$. For any constant $0 < \mu < 1/2$ and $m = O(n)$, let $D_\mu(M_n)$ denote the distribution of $M_n X + e$, where M_n is the $(m \times n)$ adjacency matrix defined above, X is a uniform random vector in $\{0,1\}^n$ and e is a noise vector in $\{0,1\}^m$ such that each bit is an independent Bernoulli distribution that takes 1 with probability μ. If the bipartite graph is a $(\omega(\log n), c)$ expander, then*

$$D_\mu(M_n) \approx D_{\mu+m^{-1}}(M_n),$$

where $\approx$ means that any circuit with size $\mathrm{poly}(n)$ cannot distinguish the two distributions with non-negligible probability.

We rely on the following lemmas.

Lemma 24. *[3, 9] For any polynomial $m = m(n)$ and constant $0 < \mu < 1/2$, and any family $\{M_n\}$ of $m \times n$ matrices over $\mathbb{F}_2$, if $D_\mu(M_n) \approx D_{\mu+m^{-1}}(M_n)$ then $D_\mu(M_n) \approx U_m$.*

Lemma 25. *[9] Let $Y \leftarrow U_{lm}$ and $E(Y)$ be defined as above. Then*

$$\Pr_{z \leftarrow E(Y)}[H_\infty(Y | E(Y) = z) \geq (1 - \delta(l)) \cdot lm] \geq 1 - e^{-(2^{-l}m)/3},$$

where $\delta(l) = 2^{-\Omega(l)}$.

Lemma 26. *[6, 26, 22] Let $g : \{0,1\}^n \to \{0,1\}^s$ be an ϵ-biased generator, and let X_s be a random variable on $\{0,1\}^s$ with min-entropy at least $(1-\delta)s$ for some $\delta > 0$. Then,*

$$|(g(U_n) + X_s) - U_s| \leq \epsilon 2^{\delta \cdot s/2 - 1/2},$$

where the vector addition is in $\mathbb{F}_2$.

We now have the following theorem.

Theorem 27. *Suppose Assumption 23 holds for some constant $c \geq 1$. Then there exist three constants $0 < \beta < 1$ and $d > c_0 > 1$ such that the circuit G described above with $d_0 > 8c/5$ is a $(\beta n, c_0)$-robust cryptographic PRG with linear stretch and input locality d.*

Proof. Assume that we fix k_1 wires in $M_n X + E(Y)$ and k_2 wires in $g(R) + Y$, such that $k_1 + k_2 = k$. This will influence at most $k_1 + k_2 = k$ output bits in $E(Y)$. Let $T_1 \subseteq [m]$ be the positions of these output bits in $E(Y)$ and $S_1 \subseteq [lm]$ be the positions of the bits in Y that generate these bits. Thus $|T_1| \leq k$ and $|S_1| \leq lk$.

We first consider the bits in $M_nX + E(Y)$. Fixing k wires is no worse than fixing kd bits in X. Let $S \subseteq [n]$ be the positions of these bits in X. Thus $|S| \leq kd$. By Lemma 7, there exists a set $T \subseteq [m]$ with $|T| \leq 8k$ such that the induced graph on left vertices $[m] \setminus T$ and right vertices $[n] \setminus S$ is an $(\ell_0 - |T|, 5d_0/8)$-expander. Thus the induced graph on left vertices $[m] \setminus (T \cup T_1)$ and right vertices $[n] \setminus S$ is also an $(\ell_0 - |T|, 5d_0/8)$-expander. Note that $\ell_0 - |T| \geq \ell_0 - 8k = \Omega(n)$ and $5d_0/8 > c$ by assumption. Also, note that the bits in X outside of S are uniform, and the bits in $E(Y)$ outside of T_1 are i.i.d bits which are equal to 1 with probability 2^{-l}. Thus by Assumption 23 and Lemma 24 we have that the bits in $M_nX + E(Y)$ outside of $T \cup T_1$ are indistinguishable from uniform. Note that $|T \cup T_1| \leq 9k$.

Next, consider the bits in $g(R) + Y$. We first fix all the bits in $M_nX + E(Y)$. Fixing X has no effect on Y or R. Now let Y' be the bits of Y outside of S_1. Thus Y' has at least $l(m-k)$ bits. Note that fixing the bits in T_1 has no effect on Y'. By Lemma 25, with probability $1 - e^{-(2^{-l}(m-k))/3} = 1 - 2^{-\Omega(n)}$ over the fixing of the other bits in $E(Y)$, Y' has min-entropy at least $(1 - \delta(l)) \cdot l(m-k)$.

Next, by Theorem 17, when we fix k wires in $g(R)$, there is a set $T_2 \subseteq [lm]$ with $|T_2| = O(k)$ such that the bits in $g(R)$ outside of T_2 are ϵ-biased, with $\epsilon = 2^{-\Omega(lm/c_2^2)}$. Let $\bar{Y}$ be the bits of Y outside of $S_1 \cup T_2$. When Y' has min-entropy at least $(1-\delta(l)) \cdot l(m-k)$, $\bar{Y}$ has min-entropy at least $(1-\delta(l)) \cdot l(m-k) - |T_2| = (1-\delta(l)) \cdot l(m-k) - O(k) \geq (1 - 2\delta(l)) \cdot l(m-k)$ for sufficiently small $k = \Omega(n)$. Thus by Lemma 26, the bits in $g(R) + Y$ outside of $S_1 \cup T_2$ are ϵ'-close to uniform, where

$$\epsilon' = \epsilon 2^{2\delta(l) \cdot l(m-k)/2 - 1/2} = 2^{\delta(l) \cdot l(m-k) - \Omega(lm/c_2^2) - 1/2}.$$

Let $\Omega(lm/c_2^2) = blm/c_2^2$ for some constant $b > 0$. Now let

$$c_2 = 2l/(1 - 1/c_1)$$

such that

$$\Delta = l\left(\frac{b}{c_2^2} - \delta(l)\right) > 0.$$

Such constants c_2 and l do exist since $\delta(l) = 2^{-\Omega(l)}$ while $b/c_2^2 = \Theta(1/l^2)$. Thus we have that

$$\epsilon' = 2^{\delta(l) \cdot l(m-k) - blm/c_2^2 - 1/2} \leq 2^{-\Delta m} = 2^{-\Omega(n)}.$$

Therefore, the output bits outside of $T \cup T_1 \cup S_1 \cup T_2$ are indistinguishable from uniform. Note that $|T \cup T_1 \cup S_1 \cup T_2| = O(k) = c_0 k$. At the same time, the input length of the G is $n + lm + lm/c_2 = (lc_1 + c_1/2 + 1/2)n$ while the output length is $(l+1)m = (lc_1 + c_1)n$. Since $c_1 > 1$ the generator has a linear stretch. Note that c_0 does not depend on the input locality d so we can make $c_0 < d$.

Finally, note that $d = O(1)$, since d is the maximum of $d_0 + l$ and 1 plus the input locality of g. □

Remark 28. *We note that any cryptographic PRG where every output bit depends on only d input bits and every input bit influences only c output bits can be easily shown to be (k, cd)-robust. The cryptographic PRG in [9] can be made to have constant c and d, thus it also has a constant q. However, our construction has two advantages. First, by using a strong robust ϵ-biased PRG as* *in Theorem 17, the $g(R) + Y$ part of our construction can already be shown to be strongly robust. Since the $M_nX + E(Y)$ part also has some strongly robust property (e.g., M_nX actually gives a strong robust r-wise independent PRG), our construction serves as a good candidate for a strong robust cryptographic PRG. Second, if we only wish to achieve a (weak) robust cryptographic PRG, then in the $g(R) + Y$ part we can use the (weak) robust ϵ-biased PRG as in Theorem 21 with a linear stretch. In this way we obtain a (k, q)-robust cryptographic PRG with $q = c_0 < d$, while the trivial argument gives $q = cd = O(d^2)$.*

5 Applications

As noted in the introduction, robust cryptographic PRGs can be directly motivated by standard cryptographic applications of PRGs such as symmetric encryption. In this section we present several applications of robust *r-wise* PRGs in cryptography. First, we show how to apply robust PRGs towards reducing the *randomness complexity* of private circuits and protocols for information-theoretic secure multiparty computation. We then rely on the low circuit size of robust PRGs towards improving the "black-box complexity" of constant-round secure two-party computation.

The following technical lemma captures a typical application scenario in which a strong robust r-wise PRG is used to replace a true source of randomness in cryptographic implementations in which the adversary has a local view of the randomness. The security of this approach will follow by showing that the view of any wire-probing adversary who attacks the "real world" implementation, in which a robust PRG is used to generate randomness, can be simulated given the view of a wire-probing adversary who attacks an ideal implementation which uses a true source of randomness. To simplify notation, we will use G to denote both the function computed by a robust PRG and its circuit implementation.

Lemma 29. *Let λ be a positive integer and $G : \{0,1\}^n \to \{0,1\}^m$ be a strong (k, q) robust r-wise PRG with $r \geq \max(\lambda, kq)$. Then, for any set S of at most k wires in G, there is a set $T \subseteq [m]$, $|T| \leq q|S|$, and a randomized algorithm Sim (a simulator) such that the following holds. For every $Q : \{0,1\}^m \to V$ which depends on at most λ bits of its input, the distributions Real and $\mathsf{Sim}(\mathsf{Ideal})$ are identical, where $\mathsf{Real} = (Q(G(X)), G_S(X))$, $\mathsf{Ideal} = (Q(R), R_T)$, X is uniformly distributed over $\{0,1\}^n$, and R is uniformly distributed over $\{0,1\}^m$. Moreover, if G is linear over the binary field then Sim can be implemented in probabilistic polynomial time.*

Proof. We let T be the set corresponding to S in Definition 2. The simulator Sim, on input (v, ρ_T), samples a uniformly random x' such that $G_T(x') = \rho_T$ (where G_T denotes the T-outputs of G) and outputs $(v, G_S(x'))$. Note that Sim can be implemented efficiently when G is linear. We prove that, when (v, ρ_T) are distributed according to $(Q(R), R_T)$, the output of Sim is distributed as Real.

Since $r \geq kq \geq |T|$, the above sampling is well defined and $(X, G_T(X)) \equiv (x', R_T)$. Hence also

$$(G_S(X), G_T(X)) \equiv (G_S(x'), R_T). \tag{1}$$

Let $L \subseteq [m]$ denote the subset of bits of R on which $Q(R)$ depends. Since $r \geq \lambda \geq |L|$, it follows from the strong robustness of G that $G_{L \setminus T}(X)$ is uniform and independent of $(G_S(X), G_T(X))$. Hence,

$$[Q(G(X)) \,|\, G_S(X) = g_S, G_T(X) = g_T] \equiv [Q(R) \,|\, R_T = g_T] \equiv [Q(R)|G_S(x') = g_S, R_T = g_T] \tag{2}$$

where $[Z \mid E]$ denotes the distribution of Z conditioned on the event E. Combining (1) and (2), we get

$$(G_S(X), G_T(X), Q(G(X))) \equiv (G_S(x'), R_T, Q(R))$$

and hence

$$\mathsf{Real} \equiv (Q(G(X)), G_S(X)) \equiv (Q(R), G_S(x')) \equiv \mathsf{Sim}(\mathsf{Ideal})$$

as required. □

Lemma 29 provides a general recipe for reducing the randomness complexity of cryptographic implementations in which the adversary's view depends on at most λ bits of randomness (but potentially also on many bits of secret data). In the next sections, we give several examples for such applications.

5.1 Private circuits

A *t-private circuit* is a randomized circuit which transforms a randomly encoded input into a randomly encoded output while providing the guarantee that the joint values of any t wires reveal nothing about the input. (For simplicity we address here the stateless variant of private circuits with encoded inputs and outputs, see [35, Section 3] and [1, Section 2.1]; a similar result applies to other variants as well.) We will show that robust r-wise PRGs can be used to reduce the randomness complexity of private circuits in which each wire depends on few bits of the randomness. The latter feature can be enforced by adding a simple "rerandomization gadget" to existing constructions.

Definition 9. (Private circuit) *A private circuit for $f : \{0,1\}^{n_i} \to \{0,1\}^{n_o}$ is defined by a triple (I, C, O), where*

- *$I : \{0,1\}^{n_i} \to \{0,1\}^{\hat{n}_i}$ is a randomized input encoder;*
- *C is a randomized boolean circuit with input $\hat{w} \in \{0,1\}^{\hat{n}_i}$, output $\hat{y} \in \{0,1\}^{\hat{n}_o}$, and randomness $\rho \in \{0,1\}^m$;*
- *$O : \{0,1\}^{\hat{n}_o} \to \{0,1\}^{n_o}$ is an output decoder.*

We say that C is a t-private implementation of f with encoder I and decoder O if the following requirements hold:

- Correctness: *For any input $w \in \{0,1\}^{n_i}$ we have $\Pr[O(C(I(w), \rho)) = f(w)] = 1]$, where the probability is over the randomness of I and ρ.*
- Privacy: *For any $w, w' \in \{0,1\}^{n_i}$ and any set P of t wires in C, the distributions $C_P(I(w), \rho)$ and $C_P(I(w'), \rho)$ are identical.*

We say that C makes an ℓ-local use of its randomness if the value of each of its wires is determined by its input $\hat{w}$ and at most ℓ bits of the randomness ρ (where the identity of these bits may depend on the wire). Unless noted otherwise, we assume I and O to be the following canonical encoder and decoder: I encodes each input bit w_i by a block of $t+1$ random bits with parity w_i, and O takes the parity of each block of $t+1$ bits.

Note that without any requirement on I and O the above definition is trivially satisfied by having I compute a secret sharing of $f(w)$ which is passed by C to the decoder. However, applications of private circuits require the use of encoder and decoder which are independent of f (and are typically smaller than the circuit size of f).

The following theorem applies robust r-wise PRGs towards reducing the randomness complexity of private circuits.

Theorem 30. *Suppose C is a qt-private implementation of f with encoder I and decoder O, where C uses m random bits and makes an ℓ-local use of its randomness. Let $G : \{0,1\}^n \to \{0,1\}^m$ be a strong (t, q) robust r-wise PRG with $r \geq t \cdot \max(\ell, q)$. Then, the circuit C' defined by $C'(\hat{w}, \rho') = C(\hat{w}, G(\rho'))$ is a t-private implementation of f with encoder I and decoder O which uses n random bits.*

Proof. We show that the view of an adversary A' who attacks $C'(\hat{w}, \rho')$ by probing a set S of $t' \leq t$ wires in G and a set P of $t - t'$ additional wires in C is independent of the input w. Since C is qt-private, it suffices to show that the view of A' can be simulated given the view of an adversary A who probes at most qt wires in $C(\hat{w}, \rho)$.

Let T and Sim be as promised by Lemma 29 for $\lambda = t\ell$. For any $\hat{w}$, let $Q_{\hat{w}}(\rho) = C_P(\hat{w}, \rho)$. Since C makes an ℓ-local use of its randomness, $Q_{\hat{w}}$ depends on at most λ bits of ρ. Thus, for any fixed $\hat{w}$, we have $\mathsf{Sim}(Q_{\hat{w}}(R), R_T) \equiv (Q_{\hat{w}}(G(X)), G_S(X))$, where R is uniform on $\{0,1\}^m$ and X is uniform on $\{0,1\}^n$. It follows that $\mathsf{Sim}(Q_{I(w)}(R), R_T) \equiv (Q_{I(w)}(G(X)), G_S(X))$. Since the distribution to which Sim is applied captures the view of an adversary A who corrupts a set $P \cup T$ of at most $qt' + (t - t') \leq qt$ wires in C and the distribution on the right hand side captures the view of A', it follows that the view of A' is independent of w as required. □

Theorem 30 implies that robust PRGs can be used to reduce the question of improving the randomness complexity of private circuits to that of improving their randomness locality. Luckily, known constructions such as the one from [35], while technically not satisfying the randomness locality condition, can be easily modified to have good randomness locality.

The t-private circuit construction from [35] emulates each gate g of a circuit implementation of f by a gadget which uses $O(t^2)$ fresh random bits to compute, given random shares $(a_0, \ldots, a_t)$ of an input $a \in \{0,1\}$ and $(b_0, \ldots, b_t)$ of an input $b \in \{0,1\}$, random shares $(c_0, \ldots, c_t)$ of $c = g(a, b)$. To prevent the randomness used for generating the shares c_i from influencing gadgets which use c as an input, one can use the following simple refreshing approach.[2] Generate fresh random bits $(r_1, \ldots, r_t)$ and let $c'_0 = c_0 \oplus r_1 \oplus c_1 \oplus r_2 \ldots \oplus r_t \oplus c_t$ (where the latter expression is computed from left to right), let $c'_i = r_i$ for $i = 1, \ldots, t$, and output $(c'_0, \ldots, c'_t)$ as the new shares of c.

Note that the outputs c'_i are completely determined by c and the fresh random bits r_i, thus ensuring that the randomness used to generate the c_i does not influence other gadgets. Moreover, the above gadget does not violate the t-privacy requirement. Indeed, the values of every $t' \leq t$ wires in the refreshing gadget can be simulated given the values of t' c_i wires in the original circuit independently of all other values. Applying the above modification to the construction of [35] we get:

Claim 31. *Any function f with circuit size s admits a t-private implementation (I, C, O) with the canonical encoder I and decoder O, where C uses $O(t^2 s)$ random bits and makes an $O(t^2)$-local use of its randomness.*

[2] This version of the refreshing gadget, suggested by Jean-Sebastien Coron, replaces the one described in [32] which failed to respect t-privacy.

Combining Claim 31 with Theorem 6 we get the following corollary.

Corollary 32. *For any polynomial $s(\cdot)$ and $\epsilon > 0$, any function f of circuit size $s(t)$ admits a t-private implementation (I, C, O) with the canonic encoder I and decoder O, where C uses $O(t^{3+\epsilon})$ random bits. Moreover, C can be efficiently constructed given a circuit for f.*

Using a naive implementation of robust PRGs obtained by taking the exclusive-or of $k+1$ independent $(r+k)$-wise PRGs would require $O(t^4)$ random bits. Improving the randomness locality of private circuits would immediate yield a corresponding improvement to Corollary 32.

An alternative model. While we considered in this section a model of private circuits which receive encoded inputs and produce encoded outputs, the results apply also to an alternative variant in which the inputs and outputs are not protected by an encoder and decoder. In this case one should settle for the following relaxed t-privacy requirement: the distribution of any set of at most t wires in C can be simulated given t bits from the input and output. To apply robust PRGs and get efficient simulation in this context, one needs to use the efficient simulation variant of Lemma 29.

5.2 Secure multiparty computation

Private circuits can be viewed as a restricted form of secure multiparty computation, with a large number of parties, each of which is assigned to evaluate a single gate of the circuit. In this section, we consider the more general case where, typically, the number of parties is significantly smaller than the circuit size. The randomness complexity of general t-private computations was studied by Canetti et al. [18] (other works, such as [38, 37, 36, 24, 14], concentrated on special values of t or on special tasks).

For concreteness, consider the case where k parties, each holding an input bit x_i, wish to compute the value $f(x_1, ..., x_k)$, for some function f. Let C_f be a circuit that computes f. The question studied in [18] is the randomness complexity of such protocols. They prove the following:

Proposition 33. *Let $f : \{0,1\}^k \to \{0,1\}$ be as above, and $t \leq \sqrt{k}$. Then, f can be computed by a t-private protocol P_f, using a trusted party which generates $\tilde{O}(t^4 \cdot |C_f|/k)$ random bits and with no additional randomness. More concretely, the trusted party picks a random string α from a space of r-wise independent strings of length $m = \tilde{O}(|C_f| \cdot t^2)$, with $r = \tilde{O}(t^4 \cdot |C_f|/k)$, and then distributes the entries of α between the k parties.*

Next, [18] switch to the standard model of secure computation, where no trusted party is available. This is done by letting $t+1$ parties each emulate the role of the trusted party by picking a random string from the above space and distributing the bits of the string among the k parties. Then, each of the k parties takes the exclusive-or of the $t+1$ strings it received and uses the result as its randomness for the protocol P_f. This can be viewed as a naive implementation of a robust PRG that results in a protocol with total randomness complexity $\tilde{O}(t^5 \cdot |C_f|/k)$. Our goal is to show that, using a better robust r-wise PRG, we can save a factor of t in the randomness, hence achieving in the standard model the same asymptotic randomness complexity as in the case of a trusted party.

The above result of [18] is meaningful only when $t \leq \sqrt{k}$ (for larger values of t, the basic BGW protocol is superior). Hence, for convenience, we will focus on the case where $k^{c_1} \leq t \leq k^{c_2}$, for some constants $c_1 < c_2 < 0.5$ and where $|C_f| \leq k^{O(1)}$.

For a robust r-wise PRG $G : \{0,1\}^n \to \{0,1\}^m$, let C_G be a circuit evaluating G. In our construction, the size of the circuits is $|C_G| = O(md) = \tilde{O}(m)$, where d is the locality of G (we have $d =$polylog(m) in our explicit construction; in the non-explicit case, we can have $d = O(1)$). We set m, r to be as in Proposition 33 and $n = r^{1/(1-\eta)}$, as in Theorem 6, with $\eta > 0$ being a small constant. We modify the protocol P_f of Proposition 33 into a protocol P'_f as follows: we let the k parties pick $\approx n/k$ random bits each (if $n < k$ then only n of the k parties pick one uniformly random bit each). Then, the k parties evaluate the circuit C_G, where each party is in charge of $\approx |C_G|/k$ of the wires. This means that the party gets the inputs of the corresponding gate and evaluates the wire. Finally, after the m-bit output of G is computed (and each output bit is delivered to the designated party), the k parties execute the protocol P_f, using the m-bit output of G as its randomness. During the evaluation of G, each party gets to see the values of $O(|C_G|/k)$ of the wires of C_G. This means that an adversary who corrupts t of the k parties gets to see a set S of $O(t \cdot |C_G|/k)$ wires of C_G (this includes the $O(t \cdot n/k)$ input wires that these t parties pick). Using the values of $|C_G|$ and m, this is $\tilde{O}(mt/k) = \tilde{O}(t^3 \cdot |C_f|/k)$ wires. The randomness complexity of P'_f is $n = r^{1/(1-\eta)} = \tilde{O}(t^4 \cdot |C_f|/k)^{1+\epsilon}$, for some $\epsilon = \epsilon(\eta)$. Since t and $|C_f|$ are both polynomial in k, this is just $\tilde{O}(t^{4+\epsilon'} \cdot |C_f|/k)$, for some ϵ'. The above construction yields:

Proposition 34. *Let $k^{c_1} \leq t(k) \leq k^{c_2}$, for some constants $c_1 < c_2 < 0.5$. Suppose P_f is a t-private protocol for computing $f : \{0,1\}^k \to \{0,1\}$, which uses m random bits and makes an ℓ-local use of its randomness, in the sense that the view of any coalition of t parties depends on at most ℓ bits of the randomness. Let $G : \{0,1\}^n \to \{0,1\}^m$ be a strong (s,q)-robust r-wise PRG with $s = \tilde{O}(t^3 \cdot |C_f|/k))$ and $r = \max(q \cdot s, \ell)$. Then, the protocol P'_f described above, is a t-private protocol for f which uses n random bits.*

The formal proof of security relies on Lemma 29, with X being the uniformly random bits picked in the first stage of P'_f by n of the parties (assumed to be honest), S is the set of wires in C_G that the adversary gets to see, and Q is the view that the adversary sees while executing P_f (this information depends on $\lambda = \tilde{O}(t^4 \cdot |C_f|/k)$ bits).

As described above, we instantiate Proposition 34 with the protocol P_f of Proposition 33 and with the PRG G of Theorem 6 to get the following theorem.

Theorem 35. *For any constants $0 < \tau < 0.5$, $c \geq 1$, and $\epsilon > 0$ the following holds. If $f : \{0,1\}^k \to \{0,1\}$ admits a circuit C_f of size $|C_f| = k^c$ then, for $t = k^\tau$, f can be computed by a t-private k-party protocol which uses a total of $O(t^{4+\epsilon} \cdot |C_f|/k)$ random bits.*

5.3 Secure two-party computation

In the previous sections, we demonstrated the usefulness of robust r-wise PRGs for reducing the *randomness complexity* of several cryptographic tasks. In this section we present a somewhat unexpected application of robust r-wise PRGs to the "black-box complexity" of constant-round secure two-party computation. Unlike the previous applications, this application is not very sensitive to the seed length (e.g., the seed length of the naive construction would suffice), but is rather sensitive to the *circuit size* of the robust PRG, requiring it to be very close to the output length. Since the end result is quite involved and relies on results from previous works, we start by describing the core technical question to which we apply robust PRGs.

Suppose that a function f admits a t-private circuit (I, C, O) where C has size s, uses $\Omega(s)$ random bits, and makes a $t^{O(1)}$-local use of its randomness. We think of s as being a large polynomial in t, much larger than the randomness locality parameter. Our goal here is to reduce the randomness complexity of C without significantly increasing its *size*. Concretely, we would like to obtain a t'-private circuit (I, C', O) for f where $t' = \Omega(t)$, C' uses $t^{O(1)}$ bits of randomness and has size at most $s \cdot \mathrm{polylog}(t)$. For this we can combine Theorem 30 with constructions of strong robust t-wise PRGs with polylogarithmic locality. (If we settle for a non-explicit[3] construction, we can make the size of C' a constant multiple of s by using a constant-locality robust PRG.) Note that the naive construction of robust t-wise PRGs (taking the exclusive-or of $t+1$ independent t-wise PRGs) is unsuitable for this application not because of its randomness complexity but rather because of its circuit size. Using the naive construction would inherently incur an $\Omega(t)$ overhead to the circuit size, compared to the desired $\mathrm{polylog}(t)$ overhead.

Disjunction-resilient circuits. The application we consider here does not require C (or C') to be t-private but rather to satisfy a different property: it should be the case that for any subset Z of wires or their negations, the *disjunction* of the (possibly negated) wires in Z is essentially independent of the input for I. More concretely, define $\Delta_{\mathsf{Disj}}(Z)$, the disjunctive distinguishing advantage of Z, by

$$\Delta_{\mathsf{Disj}}(Z) \overset{def}{=} \max_{w,w'} \left|\Pr[\mathsf{Disj}_Z(C(I(w), \rho)) = 1] - \Pr[\mathsf{Disj}_Z(C(I(w'), \rho)) = 1]\right|$$

where Disj_Z denotes the disjunction of (possibly negated) wires Z. We say that C is *disjunction resilient* if for any set of (possibly negated) wires Z, we have $\Delta_{\mathsf{Disj}}(Z) = 2^{-\Omega(t)}$. For $|Z| \le t$, the t-privacy of C ensures that $\Delta_{\mathsf{Disj}}(Z) = 0$. Furthermore, one can ensure via a local randomization gadget from [33] that for any Z we have $\Pr[\mathsf{Disj}_Z(C(I(w), \rho)) = 1] \ge 1 - 2^{-\Omega(|Z|)}$. We refer to the latter property as *local entropy.* Note that the combination of t-privacy and local entropy implies disjunction resilience.

The randomization gadget from [33] increases the size of C by a constant factor. Our goal is, as before, to get a randomness efficient C' for f whose size is close to that of C. However, now C' should be disjunction resilient rather than private. We show that if C is t-private and has the local entropy property, then the construction of Theorem 30 yields a disjunction resilient C' of low randomness complexity and roughly the same size as C.

Lemma 36. *Let $(I, C, O), G, C'$ be as in Theorem 30, where G consists only of XOR gates. Furthermore, suppose that for any set Z of size $\tau = \lceil t/2 \rceil$ and w we have* $\Pr[\mathsf{Disj}_Z(C(I(w), \rho)) = 1] \ge 1 - \epsilon$. *Then, for any set Z' in C', we have $\Delta_{\mathsf{Disj}}(Z') \le \max(\epsilon, 2^{-\tau})$.*

Proof. We use the following case analysis.

- If $|Z'| \le t$, then the t-privacy of C' implies that $\Delta_{\mathsf{Disj}}(Z') = 0$.

- If Z' involves at least τ wires in the C-part of C', then $\Pr[\mathsf{Disj}_{Z'}(C'(I(w), \rho')) = 1] \ge 1 - \epsilon$ and hence $\Delta_{\mathsf{Disj}}(Z') \le \epsilon$. This follows from the fact that for any such Z' containing exactly τ wires in the C part, the joint distribution of $C'_{Z'}(I(w), \rho')$ is identical to that of $C_{Z'}(I(w), \rho)$, and adding more wires (from both the C and G parts) can only increase the probability of the disjunction being equal to 1.

- If Z' involves at least τ wires in the G-part of C' and at most τ wires in the C part of C', we distinguish between the following two subcases, depending on the dimension dim of the linear space spanned by the parities corresponding to the G-part of Z'.

[3] The application we consider here can actually be securely implemented with a non-explicit construction in which one of the parties picks the randomness used to define the robust PRG; however, if we set the locality parameter to d, the non-explicit construction will fail with $n^{-\Omega(d)}$ probability.

- $\mathsf{dim} \le \tau$: In this subcase the G wires in Z' are determined by at most τ wires from G and at most τ wires from C, implying that $\Delta_{\mathsf{Disj}}(Z') = 0$ as in the first case.
- $\mathsf{dim} > \tau$: In this subcase we have $\Pr[\mathsf{Disj}_{Z'}(C'(I(w), \rho')) = 1] \ge 1 - 2^{-\tau}$ (since the probability is lower bounded by the probability that a random vector in a linear space of dimension dim is nonzero). Thus we have $\Delta_{\mathsf{Disj}}(Z') \le 2^{-\tau}$.

In each of the above cases we have $\Delta_{\mathsf{Disj}}(Z') \le \epsilon$ or $\Delta_{\mathsf{Disj}}(Z') \le 2^{-\tau}$ as required. □

We combine Lemma 36 with results from [21, 33] and with robust r-wise PRGs to compile any circuit C into a $2^{-\Omega(\kappa)}$ disjunction resilient circuit C' with $\mathrm{poly}(\kappa)$ randomness complexity, where the size of C' is within a polylog factor of the size of C whenever C is much larger than its depth, input, and output size.

Lemma 37. *There is a polynomial-time algorithm which given a boolean circuit $C(w)$ and a security parameter κ generates a triple $(I, C_{\mathsf{Disj}}, O)$ such that:*

- *$C_{\mathsf{Disj}}(\hat{w}, \rho)$ is a randomized boolean circuit and I, O are the canonical encoder and decoder with privacy parameter κ.*
- *C_{Disj} has size $s' = s \cdot \mathrm{polylog}(s) + \mathrm{poly}(\kappa, d, n_i, n_o)$, where d, n_i, n_o are the depth, input length, and output length of C, respectively.*
- *C_{Disj} has randomness complexity $\mathrm{poly}(\kappa, \log s)$ and randomness locality $\ell' = \mathrm{poly}(\kappa)$.*
- *For any set Z of wires in C_{Disj} or their negations, $\Delta_{\mathsf{Disj}}(Z) \le 2^{-\kappa}$.*

Proof. We construct C_{Disj} from C and κ via the following steps. First, similarly to [33], we use the efficient MPC protocol from [21] to efficiently generate, given C and κ, a κ-private implementation (I, C', O) of size $s' = s \cdot \mathrm{polylog}(\kappa) + \mathrm{poly}(\kappa, d, n_i, n_o)$ and randomness locality $\mathrm{poly}(\kappa)$. (The randomness locality feature can be obtained via a "refreshing" approach similarly to the one used in Claim 31.) The next step is to inject randomness into C' by converting it into a circuit C'' of size $O(|C'|)$ and randomness locality $\ell'' = O(\ell')$ which satisfies the local entropy property. This is done by applying the randomization gadget from [33] to each gate of C'. More concretely, the circuit C'' inherits the κ-private of C' and has the additional property that for any set Z of (possibly negated) wires and input w, $\Pr[\mathsf{Disj}_Z(C''(I(w), \rho'') = 1] \ge 1 - \Omega(|Z|)$. The final step is to apply Theorem 30 with the strong robust r-wise PRGs of Theorem 6 (with $q = O(1)$ and $k, r = O(\kappa)$) for reducing the randomness complexity of C'' while increasing its size by another polylogarithmic factor. By Lemma 36, the resulting circuit C_{Disj} has the disjunction property stated in the last item of the lemma. □

We note that by replacing the canonical I, O by more efficient ones, it is possible to remove the dependence of s' on n_i, n_o. Moreover, the entire additive term can be eliminated for almost all "natural" circuits, see [21] for discussion.

Non-interactive secure computation. To describe the final application, we recall the model of non-interactive secure two-party computation (NISC) from [33]. In a NISC protocol for a function $f(a, b)$ there is a *receiver* holding an input a and a *sender* holding an input b. The goal is to allow the receiver to learn $f(a, b)$ without learning any additional information about the sender's input b and using as little interaction as possible. We capture the latter requirement by restricting the parties to communicate via a single round of parallel calls to an ideal oblivious transfer (OT) oracle. The sender's input to each OT call is a pair of strings and the receiver's input is a selection bit; the receiver obtains the selected string from each OT call. The joint inputs each party feeds into the OT oracle are obtained by applying a randomized function of its input (depending only on this input and on local, secret randomness), as specified by the protocol.

The security requirements are as follows. First, we require that if both parties follow the protocol, then the receiver learns the correct output $f(a, b)$. Second, we require the protocol to be secure against malicious parties. That is, the protocol should guarantee that for every receiver strategy there is an effective distribution A over inputs a such that the view of the receiver can be simulated (up to computational indistinguishability) given $f(A, b)$ alone, i.e., without knowing the sender's input b. Moreover, for every sender strategy there is an effective distribution B such that for any receiver input a the receiver's output is statistically close to $f(a, B)$. We refer to a protocol as above as a *NISC protocol for f in the OT-hybrid model*.

It was shown in [33] that every f admits a NISC protocol in the OT-hybrid model in which the parties make a black-box use of a cryptographic PRG. This strengthens the classical result of Yao for "honest but curious" parties [50, 40] as well as similar results for constant-round protocols which also make a black-box use of a PRG but require additional interaction [41, 39, 34, 45].

An interesting question left open by the above line of work is that of minimizing the complexity of NISC protocols, measured by the number of PRG calls and the number of OT calls. In Yao's protocol, the parties only need to invoke the PRG $O(1)$ times for each gate of a boolean circuit computing f. In addition, they invoke the OT oracle n_a times, where n_a is the length of the receiver's input. To offer security against malicious parties while still making a black-box access to a PRG, previous constant-round protocols employed a "cut-and-choose" approach which required $O(\kappa)$ PRG calls per gate to guarantee security against a malicious sender[4] with statistical error of $2^{-\kappa}$.

The approach suggested by [33] to lower the number of PRG calls consists of the following steps:

1. Design a protocol Π for an arbitrary circuit $C(a, b)$ which only makes $O(1)$ PRG calls per gate and uses roughly n_a OT calls, but offers limited security. Concretely, a malicious sender can mount a *disjunctive attack* in which the receiver's output takes some special value whenever $\mathsf{Disj}_Z(C(a, b))$ is satisfied for some disjunctive predicate Z picked by the sender. (Recall that a disjunctive predicate is the logical OR of wires and their negations.) If the predicate is not satisfied, then the receiver gets the correct output defined by the sender's effective strategy B. This attack may correlate the receiver's output with its input in a way that violates the standard security definition.

2. To mitigate this attack, the given circuit C is converted into a randomized circuit $(I, C_{\mathsf{Disj}}, O)$ for which $\Delta_{\mathsf{Disj}}(Z) \leq 2^{-\kappa}$, where κ is a security parameter. More precisely, the encoder I is only applied to the first input a of C_{Disj}, and $\Delta_{\mathsf{Disj}}(Z)$ is defined by maximizing the advantage over all pairs or inputs (a, b) and (a', b) (with a common sender input b).

3. The circuit C_{Disj} is viewed as a deterministic circuit taking an encoded input a, an input b, and randomness ρ. It is securely evaluated using Π, where the receiver locally computes its encoded input $I(a)$, picks ρ at random, and uses both as inputs for Π. The sender uses b

[4] The level of computational security against a malicious Receiver offered by the protocol is directly inherited from the strength of the PRG and does not affect the number of PRG calls; we ignore this measure in the following.

as its input for Π. Finally, the receiver locally applies the decoder O to the output of Π to compute the output of C.

This approach is instantiated in [33] by using (essentially) the circuit C'' from the proof of Lemma 37 as C_{Disj}. This circuit has size quasilinear in the size of C, but its randomness complexity is comparable to its size. Thus, while the protocol from [33] makes a polylogarithmic number of PRG calls for each gate of C, it makes a comparable number of calls to the OT oracle. The latter cost may be viewed as prohibitive, as OT is typically much more expensive to implement than a PRG.[5] The randomness efficient version provided by Lemma 37 makes the number of OTs comparable to the receiver's input length (rather than the circuit size) while keeping the number of PRG calls per gate polylogarithmic.

We formalize the above via the following proposition. To simplify notation, we treat f as a concrete function, avoiding an explicit quantification over families of f. When using big-O notation, the hidden constants should be understood to be independent of f.

Proposition 38. *(Implicit in [33]) Suppose $f(a,b)$ admits a randomized circuit implementation $(I, C_{\mathsf{Disj}}, O)$ such that $I = I(a)$ produces n_a encoded receiver input bits and C_{Disj} has s binary (AND, OR, NOT, XOR) gates and uses n_r random bits. Moreover, suppose that $C_{\mathsf{Disj}}((I(a),b),\rho)$ satisfies the following additional condition: for any a, a', b and disjunction Z, we have*

$$\Pr[Z(C_{\mathsf{Disj}}((I(a),b),\rho)) = 1] - \Pr[Z(C_{\mathsf{Disj}}((I(a'),b),\rho)) = 1] \leq 2^{-\kappa}.$$

Then, f admits a NISC protocol Π_f in the OT-hybrid model with the following features:

- *Π_f makes $n_a + n_r + O(\kappa)$ calls to the OT oracle.*
- *Π_f makes $O(s)$ black-box calls to a (length doubling, cryptographic) PRG and no further use of cryptography.*
- *Π_f is statistically $2^{-\Omega(\kappa)}$ secure against a malicious sender.*
- *Π_f is computationally secure against a malicious receiver.*

Combining Proposition 38 with Lemma 37 yields the following.

Theorem 39. *There is a polynomial-time algorithm which, given a boolean circuit $C(a,b)$ and a security parameter κ, generates a NISC protocol Π_C in the OT-hybrid model with the following features:*

- *Π_C makes $n_a + \mathrm{poly}(\kappa, \log s)$ calls to the OT oracle, where $n_a = |a|$ and $s = |C|$.*
- *Π_C makes $s \cdot \mathrm{polylog}(s) + \mathrm{poly}(\kappa, d, n_i, n_o)$ black-box calls to a (length doubling, cryptographic) PRG and no further use of cryptography, where d, n_i, n_o are the depth, total input length, and output length of C, respectively.*
- *Π_C is statistically $2^{-\Omega(\kappa)}$ secure against a malicious sender.*
- *Π_C is computationally secure against a malicious receiver.*

[5] Known techniques for OT extension [11, 31] (namely, producing many OTs from few OTs) cannot use an underlying PRG as a black-box and do not work in the non-interactive setting unless one allows offline interaction before the inputs are known.

Similarly to the discussion after Lemma 37, the additive term in the number of PRG calls can be eliminated when the circuit is sufficiently "regular".

Acknowledgements. We thank Benny Applebaum and and Jean-Sebastien Coron for helpful discussions and comments, and an anonymous reviewer for pointing out the relevance of [4]. We also thank Jean-Sebastien Coron and Sebastian Faust for pointing out the insecurity of the original version of the refreshing gadget from Claim 31.

References

[1] Miklós Ajtai. Secure computation with information leaking to an adversary. In *STOC*, pages 715–724, 2011. Full version on ECCC 18: 82, 2011.

[2] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, pages 474–495, 2009.

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

[4] Michael Alekhnovich, Edward A. Hirsch, and Dmitry Itsyksonz. Exponential lower bounds for the running time of DPLL algorithms on satisfiable formulas. *Journal of Automated Reasoning*, 35:51–72, 2005.

[5] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k–wise independent random variables. *Random Structures and Algorithms*, 3(3):289–303, 1992.

[6] Noga Alon and Yuval Roichman. Random cayley graphs and expanders. *Random Struct. Algorithms*, 5(2):271–285, 1994.

[7] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In *STOC*, pages 805–816, 2012.

[8] Benny Applebaum, Andrej Bogdanov, and Alon Rosen. A dichotomy for local small-bias generators. In *TCC*, pages 600–617, 2012.

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

[10] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant input locality. *J. Cryptology*, 22(4):429–469, 2009.

[11] Donald Beaver. Correlated pseudorandomness and the complexity of private computations. In *STOC*, pages 479–488, 1996.

[12] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988.

[13] Nir Bitansky, Ran Canetti, and Shai Halevi. Leakage-tolerant interactive protocols. In *TCC*, pages 266–284, 2012.

[14] Markus Bläser, Andreas Jakoby, Maciej Liskiewicz, and Bodo Manthey. Private computation: k-connected versus 1-connected networks. *J. Cryptology*, 19(3):341–357, 2006.

[15] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13:850–864, 1984.

[16] Andrej Bogdanov and Youming Qiao. On the security of goldreich's one-way function. *Computational Complexity*, 21(1):83–127, 2012.

[17] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *EUROCRYPT*, pages 453–469, 2000.

[18] Ran Canetti, Eyal Kushilevitz, Rafail Ostrovsky, and Adi Rosén. Randomness versus fault-tolerance. *J. Cryptology*, 13(1):107–142, 2000.

[19] Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant-degree expansion beyond the degree/2 barrier. In *STOC*, pages 659–668, 2002.

[20] Benny Chor, Oded Goldreich, Johan Håstad, Joel Friedman, Steven Rudich, and Roman Smolensky. The bit extraction problem of t-resilient functions. In *FOCS*, pages 396–407, 1985.

[21] Ivan Damgård, Yuval Ishai, and Mikkel Krøigaard. Perfectly secure multiparty computation and the computational overhead of cryptography. In *EUROCRYPT*, pages 445–465, 2010.

[22] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In *STOC*, pages 654–663, 2005.

[23] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302, 2008.

[24] Anna Gál and Adi Rosén. Lower bounds on the amount of randomness in private computation. In *STOC*, pages 659–666, 2003.

[25] Sanjam Garg, Abhishek Jain, and Amit Sahai. Leakage-resilient zero knowledge. In *CRYPTO'11*, pages 297–315, 2011.

[26] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Struct. Algorithms*, 11(4):315–343, 1997.

[27] Shafi Goldwasser and Guy N. Rothblum. How to compute in the presence of leakage. In *FOCS*, pages 31–40, 2012.

[28] Venkatesan Guruswami, Chris Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *Journal of the ACM*, 56:1–34, 2009.

[29] Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In *TCC*, pages 107–124, 2011.

[30] Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Actively secure garbled circuits with constant communication overhead in the plain model. In *Proceedings of TCC 2017 (II)*, pages 3–39, 2017.

[31] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *CRYPTO*, pages 145–161, 2003.

[32] Yuval Ishai, Eyal Kushilevitz, Xin Li, Rafail Ostrovsky, Manoj Prabhakaran, Amit Sahai, and David Zuckerman. Robust pseudorandom generators. In *ICALP*, pages 396–407, 2013.

[33] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, and Amit Sahai. Efficient non-interactive secure computation. In *EUROCRYPT*, pages 406–425, 2011.

[34] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In *CRYPTO*, pages 572–591, 2008.

[35] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO*, pages 463–481, 2003.

[36] Eyal Kushilevitz and Yishay Mansour. Randomness in private computations. In *PODC*, pages 181–190, 1996.

[37] Eyal Kushilevitz, Rafail Ostrovsky, and Adi Rosén. Characterizing linear size circuits in terms of privacy. In *STOC*, pages 541–550, 1996.

[38] Eyal Kushilevitz and Adi Rosén. A randomnesss-rounds tradeoff in private computation. In *CRYPTO*, pages 397–410, 1994.

[39] Yehuda Lindell and Benny Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In *EUROCRYPT*, pages 52–78, 2007.

[40] Yehuda Lindell and Benny Pinkas. A proof of security of yao's protocol for two-party computation. *J. Cryptology*, 22(2):161–188, 2009.

[41] Payman Mohassel and Matthew K. Franklin. Efficiency tradeoffs for malicious two-party computation. In *Public Key Cryptography*, pages 458–473, 2006.

[42] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On epsilon-biased generators in nc^0. *Random Struct. Algorithms*, 29(1):56–81, 2006.

[43] Joseph Naor and Moni Naor. Small–bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993.

[44] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

[45] Benny Pinkas, Thomas Schneider, Nigel P. Smart, and Stephen C. Williams. Secure two-party computation is practical. In *ASIACRYPT*, pages 250–267, 2009.

[46] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Dense subsets of pseudorandom sets. In *FOCS*, pages 76–85, 2008.

[47] Ronald L. Rivest. All-or-nothing encryption and the package transform. In *FSE*, 1997.

[48] Terence Tao and Tamar Ziegler. The primes contain arbitrarily long polynomial progressions. http://arxiv.org/abs/math.NT/0610050, 2006.

[49] A. C. Yao. Theory and application of trapdoor functions. In *Proc. 23rd FOCS*.

[50] Andrew Chi-Chih Yao. How to generate and exchange secrets. pages 162–167, 1986.