The Cryptographic Hardness of Random Local Functions

(Survey)

Benny Applebaum*

Abstract

Constant parallel-time cryptography allows to perform complex cryptographic tasks at an ultimate level of parallelism, namely, by local functions that each of their output bits depend on a constant number of input bits. A natural way to obtain local cryptographic constructions is to use *random local functions* in which each output bit is computed by applying some fixed d-ary predicate P to a randomly chosen d-size subset of the input bits.

In this work, we will study the cryptographic hardness of random local functions. In particular, we will survey known attacks and hardness results, discuss different flavors of hardness (one-wayness, pseudorandomness, collision resistance, public-key encryption), and mention applications to other problems in cryptography and computational complexity. We also present some open questions with the hope to develop a systematic study of the cryptographic hardness of local functions.

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

Contents

1 Introduction

Constant parallel-time cryptography allows to perform complex cryptographic tasks at an ultimate level of parallelism, namely, by local functions that each of their output bits depends on a constant number of input bits. (Local functions also known as $\mathbf{NC}^0$ functions.) The feasibility of locally-computable cryptography was established by Applebaum, Ishai and Kushilevitz [12] in 2004: It was shown that many cryptographic tasks admit a local implementation. Concretely, for the case of one-way functions the following theorem was proven.

Theorem 1.1 (local OWF)**.** *Assuming the existence of one-way functions computable in* $\mathbf{NC}^1$*, there exist locally computable functions that are one-way.*[1]

The existence of $\mathbf{NC}^1$ computable one-way functions is considered to be a solid assumption, since it can be based on a variety of standard cryptographic assumptions (e.g., the intractability of factoring, discrete logarithms, and lattice problems). Hence, Theorem 1.1 essentially shows that local one-way functions are very likely to exist. This theorem is proven by encoding $\mathbf{NC}^1$ computable cryptographic functions into *specially crafted* local functions whose input-output dependency graph has a very specific form. A much stronger conjecture was suggested by Goldreich [40] in 2000:

Conjecture 1.2 (Random Local Functions are OWF)**.** *For* $d \geq 3$*, let* $f : \{0,1\}^n \to \{0,1\}^m$ *be a random* d*-local function that each of its output bits is computed by applying some fixed* d*-ary predicate* P *to a random set of* d *distinct inputs. Then, for a properly chosen predicate* P *and a properly chosen output length* $m = m(n)$*, the function* f *is likely to be one-way.*

The hardness of random local functions is a basic and intriguing question. Compared to the encoding-based theorems of [12], Conjecture 1.2 seems much bolder, but it also has the advantage of being simpler and it may potentially lead to constructions with better efficiency and security trade-offs. As we will also see, this conjecture has several other interesting implications.

Organization. In this survey we explore the cryptographic hardness of random local functions. We will mainly focus on hardness in terms of one-wayness (Section 3) and pseudorandomness (Section 4), but also consider other cryptographic tasks such as hashing (Section 5), and public-key encryption (Section 6). We will review known attacks and hardness results, mention applications to other problems in cryptography and computational complexity, and present some open questions. We hope that this survey will encourage further study of the cryptographic hardness of random local functions.

2 Preliminaries

General. We let $[n]$ denote the set $\{1, \ldots, n\}$. For a string $x \in \{0,1\}^n$ and $i \in [n]$, we let x_i denote the i-th bit of x. For an ordered set $S = (i_1, \ldots, i_d)$, we let $x_S \in \{0,1\}^d$ denote the *ordered* restriction of x, i.e., the string $x_{i_1} \ldots x_{i_d}$. For a distribution or random variable X (resp., set), we write $x \xleftarrow{R} X$ to denote the operation of sampling a random x according to X (resp., uniformly from X). Through this survey the term *efficient* refers to probabilistic polynomial-time algorithms.

[1] Recall that $\mathbf{NC}^1$ is the class of functions computable by Boolean circuits of polynomial size, logarithmic depth, and bounded fan-in. We mention that the theorem extends to the case of polynomial size branching programs which essentially correspond to log-space computation.

Negligible, noticeable, and high probability. We use the standard cryptographic convention by which a function $\varepsilon : \mathbb{N} \to [0,1]$ is *negligible* (resp., *noticeable*) if for all sufficiently large n's, $\varepsilon(n) < n^c$ for every constant $c > 0$ (resp., $\varepsilon(n) > n^c$ for some constant $c > 0$). A sequence of events E_n happens with high probability (*whp*) if $\lim_{n\to\infty} \Pr[E_n] = 1$.

Hypergraphs. An (n, m, d)-*hypergraph* is a hypergraph over n vertices $[n]$ with m hyperedges $S_1, \ldots, S_m$ each of cardinality d. We assume that each hyperedge $S = (i_1, \ldots, i_d)$ is ordered, and that all the d members of an hyperedge are distinct. We let $\mathcal{G}_{n,m,d}$ denote the distribution over such hypergraphs in which a hypergraph is chosen by picking each hyperedge uniformly and independently at random among all the possible $n \cdot (n-1) \cdot \cdots \cdot (n-d+1)$ ordered hyperedges. The reader may envision the hypergraph as a bipartite graph with vertices on the left and hyperedges on the right.

Random Local Function. For a d-ary predicate P, and positive integers n and m, we let $\mathcal{F}_{P,n,m}$ denote the collection of d-local functions such that each member $f_{G,P} : \{0,1\}^n \to \{0,1\}^m$ is specified by an (n, m, d)-hypergraph G, and the i-th output of $f_{G,P}$ is computed by applying the predicate P to the d inputs that are listed in the i-th hyperedge. We sample a function $f_{G,P}$ from $\mathcal{F}_{P,n,m}$ by choosing a random hypergraph G from $\mathcal{G}_{n,m,d}$.

3 Inversion

The most basic form of cryptographic hardness is one-wayness. In this section we study how hard it is to invert a random local function.

The inversion problem. The *inversion* problem for $\mathcal{F}_{P,n,m}$ is defined as follows:

- **Input**: a random hypergraph $G \xleftarrow{R} \mathcal{G}_{n,m,d}$ and an m-bit string $y = f_{G,P}(x)$ where $x \xleftarrow{R} \{0,1\}^n$.
- **Output**: a preimage x' of y under $f_{G,P}$ (i.e., $f_{G,P}(x') = y$).

We say that the collection $\mathcal{F}_{P,n,m}$ is ε *hard to invert* if every efficient adversary $\mathcal{A}$ cannot solve the problem with probability larger than ε. By default, we let ε be some negligible function.

It will be beneficial to view the inversion problem as a *random constraint satisfaction problem* (CSP) over n variables $x = (x_1, \ldots, x_n)$ with d-ary constraints of the form:

$$\begin{cases} y_1 = P(x_{S_1}), \\ \quad \vdots \\ y_n = P(x_{S_m}), \end{cases} \tag{1}$$

where $S_1, \ldots, S_m$ are the hyperedges of G. In other words, we would like to find a satisfying assignment for a random CSP problem with a random planted solution. As we will see, this view of the inversion problem will allow us to adopt algorithmic ideas and hardness results from the rich literature of random constraint satisfaction problems (e.g., Random 3-SAT, see [37, 28, 1] for surveys).

The output length. In this section, we will focus on the regime where the output length m is at least as large as the input length n. It can be shown that, beyond some threshold $m_0 = n + \Omega(n)$, if $\mathcal{F}_{P,n,m}$ is one-way then so is $\mathcal{F}_{P,n,m'}$ for $m_0 < m' < m$. We will therefore measure the quality of an inversion algorithm in terms of the value m for which it inverts $\mathcal{F}_{P,n,m}$.

3.1 Failure of "Myopic" Algorithms

One natural strategy to invert $\mathcal{F}_{P,n,m}$ is by using some form of divide-and-conquer approach. The (naive) hope is that the local structure of the constraints enables a decomposition into small subproblems whose solutions can be later combined to form a global solution. In general, this approach fails. Due to the expansion of the constraint hypergraph, any small subset of the constraints gives very little information on the global solution [40]. This meta-argument was formalized in several ways to rule out different forms of local or myopic algorithms [40, 5, 29]. To illustrate this idea let us consider the following t-myopic algorithm which in each step reads t constraints (i.e., t bits of y) and based on them (and all the previous observations) assigns a value to some input variable x_i. See Figure 1.

- **Input**: an (n, m, d)-hypergraph G, a string $y \in \{0,1\}^m$.
- **Parameter**: Integer $t > 0$.
- Initialize an empty assignment $x' = \star^n$, and an empty set $R = \emptyset$ of revealed outputs. (At this point, all the bits of y are "covered".)
- While there exists an unassigned input variable do:
  - Choose a set A of t output coordinates $A \subset [m]$ and add them to R. (From now on, the bits y_A are "uncovered").
  - Based on (G, R, y_R), choose an unassigned input variable i and assign x'_i. If there is no consistent assignment abort with "Failure". (The choice may be based on any computationally unbounded strategy.)
- **Output**: x'.

Figure 1: The basic t-myopic algorithm.

Analysis. We rely on the following key property. Let us say that a function f is (r, ℓ, h)-*robust* if even after reading an arbitrary set of r outputs, the posterior distribution on *every* set of ℓ inputs has at least h bits of min-entropy.[2] Formally, for every r-subset of outputs $R \subseteq [m]$, ℓ-subset of inputs $L \subseteq [n]$ and string $w \in \{0,1\}^\ell$,

$$\Pr[x'_L = w] \leq 2^{-h}$$

where x' is chosen by first choosing $x \xleftarrow{R} \{0,1\}^n$, letting $y = f(x)$, and then choosing x' at random such that $f(x')_R = y_R$.

[2]This notion of robustness is implicit in [5] and was made explicit in [9].

Lemma 3.1 (implicit in [5]). *Fix some integer t and assume that there exist integers ℓ, h and k such that $f_{G,P}$ is $(\ell \cdot t, \ell, h)$-*robust *and each string y has at most 2^k preimages under $f_{G,P}$. Then, the t-myopic algorithm succeeds with probability smaller than $2^{-(h-k)}$.*

Proof. After ℓ steps the algorithm fixes ℓ input variables based on at most $\ell \cdot t$ output bits of y. The min-entropy of the partial assignment is h while the number of preimages of y is at most 2^k, hence the probability of hitting a partial assignment that can be extended to a preimage is 2^{k-h}. □

It turns out that $\mathcal{F}_{P,n,m}$ is likely to be robust. Specifically, the following lemma was proved in [29] (extending the techniques of [5]).

Lemma 3.2 ([29]). *For most d-ary predicates P, and some $r = O(n/d), \ell = 2^{-o(d)}n, h = 2^{-o(d)}n$, a function $f \xleftarrow{R} \mathcal{F}_{P,n,n}$ is likely to be (r, ℓ, h) robust and likely to have an expected number of $2^{2^{-o(d)}n}$ preimages .*[3]

It is essentially shown that $f_{G,P}$ is robust whenever the hypergraph G enjoys some expansion properties and the predicate P is "sensitive" enough (e.g., for some parameter $c < d$, even after fixing at most c of P's inputs, P remains unfixed). Combining the above lemmas, we derive the following corollary.

Theorem 3.3 ([29]). *For most d-ary predicates P and some $t = 2^{o(d)}/d$, the expected success probability of the basic t-myopic algorithm in inverting $\mathcal{F}_{P,n,n}$ is* $\exp(-\Omega(n))$.

A similar theorem holds for larger output lengths (e.g., every $m = O(n)$). When the output length is polynomial (e.g., $m = n^{1.1}$), the results of [9] imply a subexponential bound $\exp(-n^{\varepsilon})$ on the success probability of $\mathcal{F}_{P,n,m}$, alas for a more restricted family of predicates. (See Section 4.2.)

Stronger algorithms. One can use similar techniques to rule out stronger variants of myopic algorithms [5, 29, 48]. For example, assume that whenever the t-myopic algorithm "gets stuck" with an unsatisfiable system (i.e., the partial assignment x' contradicts some constraints $f_{G,P}(x)_i = y_i$ for $i \in R$), the algorithm is allowed to "regret" and change its last "free" decision. Namely, the algorithm backtracks to the last variable j that could be assigned to both 0 and 1, and flips the assignment from x'_j to $1 - x'_j$. After sufficiently many steps, this *backtracking myopic* algorithm (which falls under the framework of DPLL-type algorithms) will surely find a preimage. However, it can be shown [5, 29] that the expected running time will be exponential in n for $\mathcal{F}_{P,n,O(n)}$ and most d-ary predicates. The high level argument goes as follows. (1) By Lemmas 3.1 and 3.2, after $\ell = \Theta(n)$ steps the algorithm is likely to fix the first ℓ variables L mistakenly to a value that cannot be extended to a satisfying assignment. This means that the residual problem $(f_{G,P}(x) = y, x_L = x'_L)$ is unsatisfiable. (2) As argued next, it takes exponential time for the backtracking myopic algorithm to recover from its mistake, and backtrack to the ℓ-th iteration. This is proved by showing that (a) the trace of the algorithm's execution provides a tree-like resolution proof for the unsatisfiability of the residual problem; and (b) with high probability, any such proof must be of exponential size (cf. [18]).

[3] A special case of this lemma (for 3-XOR predicate) was originally proved by [5].

3.2 Linearization

Algebraic linearization. In light of the failure of local inversion algorithms, let us try to apply a more global approach. First, observe that if the predicate is XOR then one can efficiently invert the function by solving a system of linear equations over the binary field $\mathbb{F}_2$. More generally, assume that the predicate P can be written as a degree c polynomial over $\mathbb{F}_2$. Then we can view the m constraints $\{y_i = P(x_{S_i})\}$ as a linear system over n^c new variables $z_{i_1,\ldots,i_c} = x_{i_1} \cdot \ldots \cdot x_{i_c}$. If the number of constraints is sufficiently large, that is, $m = \Omega(n^c \log n)$, then (*whp*) there exists a unique solution z to the system, which can be found efficiently. At this point, we are left with the simple task of finding a vector x which satisfies a (satisfiable) system of AND constraints of the form $x_{i_1} \cdot \ldots \cdot x_{i_c} = z_{i_1,\ldots,i_c}$. Such a system can be efficiently solved by assigning 1 to each input variable x_i that participates in a constraint whose RHS is 1, and 0 to all other variables. It follows, that for $m = \Omega(n^{\deg(P)} \log n)$, the collection $\mathcal{F}_{P,n,m}$ is invertible.

Theorem 3.4. *If P has degree c over $\mathbb{F}_2$, then the collection $\mathcal{F}_{P,n,\Omega(n^c \log n)}$ is efficiently invertible.*

Since any d-ary predicate P can be written as a degree d-polynomial, we conclude that $\mathcal{F}_{P,n,m}$ is always invertible for $m = \Omega(n^d \log n)$. (A better bound will be proven later.)

Fourier linearization. A different form of linearization arises when the predicate $P(w)$ is (even slightly) correlated with the parity of c of its inputs (c-XOR). We say that P is *c-correlated* if $c > 0$ is the minimal positive integer for which

$$\Pr_{w \xleftarrow{R} \{0,1\}^d} [P(w) = \sum_{i \in T} w_i \pmod 2] \neq \frac{1}{2},$$

for some c-subset $T \subseteq [d]$. Fix such a predicate P, and assume, wlog, that the correlation is positive, i.e., the LHS is larger than $\frac{1}{2} + \varepsilon$. (Otherwise, replace P by its complement.) Since the predicate is d-local, we may further assume that $\varepsilon \geq 2^{-d} = \Omega(1)$. To simplify the following discussion, let us further assume that the predicate is balanced. (The case of an unbalanced predicate can be treated by a variant of the following attack.)

We can now view each of the m original constraints as a noisy linear equation

$$y_i = e_i(x) + \sum_{j \in S'_i} x_j \pmod 2, \tag{2}$$

where the c-tuple S'_i is the T-projection of the original d-tuple S_i, and each of the e_i's is a Bernoulli random variables with expectation of $\frac{1}{2} - \varepsilon$. In fact, even if x_{S_i} is treated as a fixed value, the expectation of the random variable e_i, induced by the choice of $x_{S_i \setminus S'_i}$, is $\frac{1}{2} - \varepsilon$. (This follows from the minimality of c and the fact that P is balanced.)

Say that the hypergraph G contains t hyperedges $S_1, \ldots, S_t$ whose T-projection correspond to the same c-tuple S', but all their other entries are disjoint, i.e., the sets $S_1 \setminus S', \ldots, S_t \setminus S'$ are pairwise disjoint. Then, we can derive t noisy copies of the term $\sum_{j \in S} x_j$ where the noise terms $e_1, \ldots, e_t$ are *statistically independent* Bernoulli random variables with expectation of $\frac{1}{2} - \varepsilon$. In this case, we can take a majority vote over these noisy terms and recover $\sum_{j \in S} x_j$ with probability $1 - 2^{-\Omega(\varepsilon^2 t)}$. After collecting $m = \Omega(n^c \log n)$ random hyperedges, we expect to see $t = \Omega(\log n)$ copies of each equation S'_i, and so we can (with constant probability) simultaneously "purify" the noise in all these equations, and recover x by solving a linear system.

Optimization. In fact, we can do better and reduce the output length required for this attack to $m = \Omega(n^{c/2} + n \cdot \log n)$ or even to $m = \Omega_d(n^{c/2} + n)$. This is done in three steps. In the first step, we use the first $m_1 = \Omega(n^{c/2})$ outputs to obtain a system (2) of m_1 noisy c-LIN equations, and convert this system into a random system of $t = \Omega(n)$ noisy 2-LIN equations. Each 2-LIN equation is generated by XOR-ing together a pair of c-LIN equations which share exactly $c-1$ variables. By standard calculation ("birthday paradox"), a collection of $m_1 = \Omega(\sqrt{tn^{c-1}}) = \Omega(n^{c/2})$ original equations is expected to contain t such pairs.

In the second step, we apply one of the known algorithms (e.g., the SDP of [39] or [27]) to obtain a solution x' that satisfies a large fraction of the 2-LIN constraints. Since the constraint graph of the 2-LIN instance is random, it can be shown that the assignment x' is likely to be correlated with the original (planted) solution x. That is, the relative Hamming distance between x' and x is at most $\frac{1}{2} - \varepsilon$ for some constant $\varepsilon > 0$ (see [8]).

In the final step, we use x' and additional $m_2 = \Omega(n \log n)$ noisy c-linear equations to recover a solution x''. The i-th bit of x'' is recovered as follows: (a) Collect $\Omega(\log n)$ equations in which the i-th variable x_i participates; (b) Compute $\Omega(\log n)$ "votes" for the value of x''_i by substituting all the other variables with their value under the approximate solution x'; and (c) Set x''_i to be the majority among all the votes. It can be shown that (*whp*) this voting procedure recovers the i-th bit of the planted solution. Overall, the algorithm works when the output length m is larger from $m_1 + m_2 = \Omega(n^{c/2} + n \log n)$ outputs. In fact, one can reduce the complexity of the last amplification step to $m_2 = \Omega_d(n)$ outputs (see Section 3.3), which leads to the following theorem.

Theorem 3.5. *If $P : \{0,1\}^d \to \{0,1\}$ is c-correlated, then the collection $\mathcal{F}_{P,n,\Omega_d(n^{c/2}+n)}$ is efficiently invertible.*

Interestingly, algebraic linearization (Theorem 3.4) and Fourier linearization (Theorem 3.5) complement each other. Siegenthaler [60] proved that any (non-linear) d-ary predicate P of algebraic degree c_1 over $\mathbb{F}_2$, must be c_2-correlated for $c_2 \leq d - c_1$. Hence, when m is roughly $n^{d/3}$, it is possible to invert $\mathcal{F}_{P,n,m}$ either via Theorem 3.4 or via Theorem 3.5. A more careful calculation yields the following corollary.

Corollary 3.6. *For any d-ary predicate P and $m = \Omega(n^{\frac{1}{2}\lfloor 2d/3 \rfloor} \log n)$, the collection $\mathcal{F}_{P,n,m}$ is efficiently invertible.*

Proof. First, observe that if $d = 2$ the inversion problem can be reduced to a 2-CNF problem, and can therefore be solved efficiently for any value of m. Inversion is also trivial when P is linear (i.e., XOR or its complement). Hence, we may assume that P is non-linear and that $d > 2$. We distinguish between two cases. If the algebraic degree c_1 of P is smaller or equal to $\lfloor d/3 \rfloor$, then Theorem 3.4 allows to invert $\mathcal{F}_{P,n,m}$ once $m = \Omega(n^{c_1} \log n)$, let alone when $m = \Omega(n^{\frac{1}{2}\lfloor 2d/3 \rfloor} \log n)$. On the other hand, if $c_1 > \lfloor d/3 \rfloor$ then, by Siegenthaler's theorem, P must be c_2-correlated for some $c_2 < d - \lfloor d/3 \rfloor$. Since c_2 is an integer, it must hold that $c_2 \leq \lfloor 2d/3 \rfloor$, and so, by Theorem 3.5, $\mathcal{F}_{P,n,m}$ can be inverted when $m = \Omega_d(n^{\frac{1}{2}\lfloor 2d/3 \rfloor} + n)$. For $d > 2$, the latter simplifies to $\Omega_d(n^{\frac{1}{2}\lfloor 2d/3 \rfloor})$. □

Corollary 3.6 provides the best known attack against a general predicate.

Question 3.7. *Is it possible to efficiently invert the collection $\mathcal{F}_{P,n,m}$ for every predicate P and some $m = n^{\frac{1}{2}\lfloor 2d/3 \rfloor - \varepsilon}$ for some $\varepsilon > 0$?*

A more concrete challenge is to solve the above question for the predicate

$$\mathsf{MST}_{d_1,d_2}(w_1,\ldots,w_{d_1},z_1,\ldots,z_{d_2}) = (w_1 \oplus \cdots \oplus w_{d_1}) \oplus (z_1 \wedge \cdots \wedge z_{d_2}), \tag{3}$$

where $d_1 = 2d_2$. Originally introduced by Mossel, Shpilka and Trevisan [54], this predicate has algebraic degree of d_2 and it is uncorrelated with parities of arity smaller than d_1.

3.3 Self Correction

Suppose that we are given, as an advice, a string x' which is correlated with a true preimage x. Is it possible to efficiently invert $f_{G,P}$? It turns out that the answer is positive as long as the advice is good enough and the output length m is sufficiently large.

Formally, let us define the problem of *inversion with ε-leakage* as follows. The input consists of a triple (G, y, x') where $G \xleftarrow{R} \mathcal{G}_{n,m,d}$, $y = f_{G,P}(x)$ where $x \xleftarrow{R} \{0,1\}^n$, and x' is a random string that is $(\frac{1}{2} - \varepsilon)$-close to x in (relative) Hamming distance. As before, the goal is to recover some preimage of y. It is important to note that the advice string x' is chosen solely based on x, and it is *statistically independent* from the hypergraph G. The following theorem was proven in [23, Theorem 1.3].

Theorem 3.8 (Self-Correction)**.** *For every d-ary predicate P, every $\varepsilon > 0$, and every $m > n \cdot (k/\varepsilon)^{2d}$, where k is some universal constant, it is possible to efficiently invert $\mathcal{F}_{P,n,m}$ with ε-leakage.*

Thinking of ε as a constant, it follows that when $m = \Omega_d(n)$, one can efficiently invert $\mathcal{F}_{P,n,m}$ with ε-leakage.

Proof idea. The algorithm is based on a three-phase approach originally suggested in the context of planted colorability [6] and planted SAT [38]. Ignoring some technical details (and oversimplifying) the algorithm proceeds as follows: First, we pass over all input variables (in parallel) and flip the value of x'_i if it appears in "too many" constraints that are violated by x'. After repeating this basic amplification step $O(\log n)$ times, we will have (*whp*) an assignment x'' that agrees with the planted assignment on all but a small fraction of "untypical" inputs.

In the second phase, we will identify (based on the properties of the hypergraph) a large set of inputs for which our assignment is likely to be correct, and unassign all other input variables. As a result we will obtain a partial assignment with no error.

Finally, we complete the partial assignment to a satisfying assignment via brute-force. The analysis shows that *whp* the last step can be done efficiently as the partial assignment breaks the original problem to small (logarithmic-size) independent sub-problems. □

Applications. Theorem 3.8 shows that, for sufficiently large output length (i.e., $m = \Omega_d(n)$), the task of inverting $\mathcal{F}_{P,n,m}$ reduces to the task of finding an approximate preimage. We can think of this as a hardness result: If inversion is hard, then even *approximate inversion* is hard. In some cases, however, approximate inversion can be done efficiently and the theorem leads to inversion algorithms. For example, when the predicate P is c-correlated, the linearization technique outlined in Section 3.2 allow us to approximately invert $\mathcal{F}_{P,n,\Omega(n^{c/2})}$ up to a distance of, say 0.51. Combined with self-correction, this leads to full inversion for output lengths of $\Omega_d(n^{c/2} + n)$, as claimed in Theorem 3.5. In particular, in the special case of 2-correlated predicate one can use spectral

techniques and find an approximate preimage even when the output length is linear $m = \Omega_d(n)$. (See [23].) Hence, for such predicates, $\mathcal{F}_{P,n,m=\Omega_d(n)}$ is efficiently invertible.

For general predicates, Theorem 3.8 yields the following natural attack: (1) Generate a list of candidates which contains a $(\frac{1}{2} - \varepsilon)$-approximate preimage, and (2) Apply the algorithm of Theorem 3.8 to each of these candidates. To implement the first step, assign random values to the first $(1-2\varepsilon)n$ input variables, and enumerate over all possible assignments to the last $2\varepsilon n$ variables. With probability $\frac{1}{2}$, the random part of the assignment will agree with the planted assignment x on at least half of its coordinates, and so the list will contain an assignment x' which agrees with x on $\frac{1}{2}(1-2\varepsilon)n + 2\varepsilon n = (\frac{1}{2} + \varepsilon)n$ of the coordinates. The complexity of the algorithm is $2^{2\varepsilon n}$. When the output length is super-linear, i.e., $m = n^{1+c}$ for constant $c > 0$, we can apply Theorem 3.8 with inverse-polynomial $\varepsilon = n^{-\delta}$ for some constant $\delta = \delta(c, d) > 0$, and derive a subexponential inversion algorithm of complexity $2^{O(n^{1-\delta})}$.

3.4 Graph-Related Leakage

Theorem 3.8 says that inversion becomes easy if large enough fraction of the bits of the preimage x are leaked. As already noted, the theorem assumes that the leakage is *independent* of the hypergraph $G \xleftarrow{R} \mathcal{G}_{n,m,d}$. Namely, the set of indices on which x' and x agree is chosen uniformly at random independently of the hypergraph G. In contrast, it can be shown that inversion remains (somewhat) hard when the set of agreement $\{i : x'_i = x_i\}$ depends on G. Specifically, suppose that we reveal the substring x_T where T is the union of a random subset of the hyperedges of G. (Equivalently, each hyperedge of G is selected independently at random with probability $\frac{1}{2}$, and all the inputs which participate in the selected hyperedges are leaked.) In [10] it is shown that, for random local functions, inversion with "random-exposure" is not (much) easier than standard inversion with no leakage.[4] The basic idea is to argue that with noticeable probability one can embed a "small" standard inversion problem in an inversion problem with random exposure.

To get some intuition, assume that the exposure is applied to a random function with $m = 2cn$ outputs, n inputs and (expected) locality d. Further, assume that exactly $m/2$ hyperedges were exposed and so there are cn unexposed hyperedges. In this case, an input bit is not exposed (i.e., does *not* participate in an exposed hyperedge) with probability roughly $(1 - d/n)^{cn} \approx e^{-cd}$. Hence, the expected number of unexposed inputs is $n' = n/e^{cd}$, and so the hypergraph G contains a unexposed subgraph G' with n' inputs and $m' = ce^{cd}n'$ outputs. It turns out that a random local function $f : \{0,1\}^{n'} \to \{0,1\}^{m'}$ can be embedded in this subgraph. (See [10].)

For $c = \ln d/d$, it follows that if a random local function which expands n' inputs to $m' = (\ln d)n'$ outputs is hard to invert, then a random local function which shrinks n inputs to $(2 \ln d/d)n$ outputs is hard to invert with "random-exposure". The existence of functions which remain one-way in the presence of random-exposure was used in [10] to construct one-way functions with optimal output locality of 3.

[4] For technical reasons, the result applies to a variant of $\mathcal{F}_{P,n,m}$ in which each output is computed by a *different* predicate $P_i : \{0,1\}^{d_i} \to \{0,1\}$. Each predicate is chosen by first choosing an arity d_i from the binomial distribution $B(n, d/n)$, and then choosing P_i uniformly at random from all predicates of arity d_i. It should be noted a random predicate yields an easy-to-invert collection when the output length $m = cn$ is sufficiently large, i.e., $c = \text{poly}(d)$. Still, when the output length m is close to n, no efficient attacks are known.

3.5 Hardness Amplification

Is it possible to find a small ε-fraction of inputs $X \subseteq \{0,1\}^n$ for which $\mathcal{F}_{P,n,m}$ is efficiently invertible? In [24, Theorem 6.1], it is shown that, for a random local function $f \xleftarrow{R} \mathcal{F}_{P,n,m}$ with sufficiently large output length $m = \Omega_d(n)$, the existence of an easy set of inputs for f of density $\varepsilon > 2^{-n^{O(1)}}$ implies that f is invertible on $1-\varepsilon$ fraction of the inputs in subexponential-time of $\exp(\tilde{O}(\sqrt{n}))$.[5] Hence, one-wayness with respect to sub-exponential time has an all-or-nothing flavor: Either almost all inputs are hard to invert, or almost all inputs are easy to invert.

The proof is based on the following idea. Given a weak inversion algorithm $\mathcal{A}$ that inverts $f_{G,P}$ over a set of inputs X, we construct an inversion algorithm $\mathcal{B}$ that inverts $f_{G,P}$ over the set X' that contains all the inputs which are k-close to X in (absolute) Hamming distance, where $k = \sqrt{2\ln(1/\varepsilon)n}$ and X has density at least ε. The key observation is that if x' is k-close to $x \in X$, then the image $y' = f_{G,P}(x')$ is likely to be $O(k)$ close to the image $y = f_{G,P}(x)$, since each input in $f_{G,P}$ is expected to influence a constant number of outputs. Ignoring some technical details, we can now invert $f_{G,P}$ at y' as follows: (1) search for y by flipping all possible sets of $O(k)$ coordinates; (2) apply $\mathcal{A}$ to invert y and obtain x, and then (3) search for x' by flipping all possible sets of k coordinates. In other words, we first "walk" on the range space of $f_{G,P}$ until we get to an easy instance y which can be inverted to x, and then, we recover x' by "walking" back from x, this time on the domain space. A standard probabilistic argument shows that, by taking $k = \sqrt{2\ln(1/\varepsilon)n}$, at least $1-\varepsilon$ fraction of the inputs x' will be k-close to some instance in the easy set (whose density is ε).

3.6 Inversion: Discussion

We end this section with an attempt to draw the borderline (in terms of locality and output length) between easiness and hardness. Let $T_P(n,m)$ denote the complexity of inverting $\mathcal{F}_{P,n,m}$ with noticeable probability, and let $T(d,n,m) = \max_{P:\{0,1\}^d \to \{0,1\}} T_P(n,m)$. We saw that $T(d,n,n^c)$ is (at most) sub-exponential for any $c > 1$, and polynomial for $c > d/3$ (or more accurately for $c > \frac{1}{2}\lfloor 2d/3 \rfloor$)). In light of this, we put forward the following conjecture.

Conjecture 3.9. *For every constant $c > 1$, there exists a constant d and a d-ary predicate P, for which the collection $\mathcal{F}_{P,n,n^c}$ is hard to invert, i.e., $T(d,n,n^c)$ is super-polynomial.*

It should be mentioned that for linear output lengths $m = O(n)$, the complexity of the best known attacks is $2^{\Omega(n)}$. Hence, one may (strongly) conjecture that for some predicate P, the collection $\mathcal{F}_{P,n,m=O(n)}$ is exponentially-hard to invert. Moreover, for $m = n$, this conjecture may even hold for *most* d-ary predicates and every $d \geq 5$. (In contrast, for every 2-ary predicate P and every m, the collection $\mathcal{F}_{P,n,m}$ is easy to invert.)

We believe that exploring Conjecture 3.9 is an important research direction. It will be also interesting to study a concrete version of the conjecture with an explicit dependency of the locality d in the output length $m = n^c$. All we currently know is that one-wayness requires $d > 3c$ due to Corollary 3.6. We ask: Does it suffice to take $d = \Omega(c)$ or maybe a larger locality of $d > \text{poly}(c)$ or even $d > \exp(c)$ is needed? Some recent results suggest that the bounds obtained in Corollary 3.6 may be tight, at least for a rich family of algorithms [57, 36].

[5] This result holds *whp* over the choice of the hypergraph, and even if the "easy inputs" are invertible in time $\exp(\tilde{O}(\sqrt{n}))$.

It will be also useful to study concrete suggestions for the predicate P. Ideally, one may hope to identify the "hardest" d-ary predicate $P^\star$ and prove a completeness result of the form: "if $\mathcal{F}_{P,n,m}$ is hard for some d-ary predicate P then so is $\mathcal{F}_{P^\star,n,m'}$". While we do have some understanding of potential "easy" and "hard" predicates (see also next section), the existence of complete predicates is currently wide open. We note that the natural choice of using a random predicate as the hardest d-ary predicates fails when the output length is sufficiently large ($m > \text{poly}(d)n$) as with high probability such a predicate will be correlated with one of its inputs.

Bibliographic Note. Linearization attacks were originally considered in [30, 54] in the context of locally-computable *pseudorandom generators* with an *arbitrary* dependencies hypergraph (as opposed to random). Theorems 3.4 and 3.5 and Corollary 3.6 (which have not appeared before) are mainly based on the ideas of [54, 23, 8], and may be considered a folklore.

4 Pseudorandomness

We move on to study the pseudorandomness of $\mathcal{F}_{P,n,m}$. Let us recall the notion of collection of pseudorandom generators. In the following, we let $\mathcal{F}$ denote a collection of efficiently computable functions, in particular, each function $f : \{0,1\}^n \to \{0,1\}^m$ in $\mathcal{F}$ has a succinct representation $\langle f \rangle$ of size $\text{poly}(n)$ (e.g., as a polynomial-size circuit).

Definition 4.1. *A collection of length-increasing functions $\mathcal{F} = \{f : \{0,1\}^n \to \{0,1\}^m\}$ is ε-pseudorandom generator (ε-PRG) if for every efficient algorithm $\mathcal{A}$ the* distinguishing advantage

$$\left| \Pr_{f \xleftarrow{R} \mathcal{F}, x \xleftarrow{R} \{0,1\}^n} [\mathcal{A}(\langle f \rangle, f(x)) = 1] - \Pr_{f \xleftarrow{R} \mathcal{F}, y \xleftarrow{R} \{0,1\}^m} [\mathcal{A}(\langle f \rangle, y) = 1] \right| \tag{4}$$

is at most $\varepsilon(n)$.

We emphasize that since we are dealing with *collection* of functions, the adversary's input consists the description of a random function (in our case a hypergraph chosen from $\mathcal{G}_{n,m,d}$) and an m-bit challenge string. When the collection consists of a single function, we derive the standard notion of ε-pseudorandom generator. (See [41, Section 2.4.2] for an analogous treatment for the case of OWF collections.) Thus, in the case of $\mathcal{F}_{P,n,m}$ the distinguisher is given a random (n, m, d)-hypergraph G and a string $y \in \{0,1\}^m$ and its goal is to distinguish between the case where y is chosen at random and the case where y is a random image of $f_{G,P}$.

The stretch. Pseudorandomness becomes non-trivial when the output length m is larger than n. In fact, we will be interested in the regime where the *stretch* $m-n$ is linear in n, or even polynomially larger than n. The existence of such high-stretch pseudorandom generators with low locality has turned to be an important open question with several applications [30, 54, 4, 13, 47, 8, 7, 46]. For example, as shown in [47], such PRGs allow to improve the *sequential complexity* of cryptography: A linear-stretch PRG with constant locality would lead to implementations of several basic primitives (e.g., public-key encryption, commitment schemes) with *constant* computational overhead, and a polynomial-stretch PRG with constant locality would lead to *secure computation protocols* with *constant* computational overhead – a fascinating possibility which is not known to hold under any other cryptographic assumption. Currently, all known candidates for large-stretch PRGs with low

locality are essentially based on random local functions. In contrast, the PRGs derived by [12] have sub-linear stretch (i.e., $m = n + n^{1-\varepsilon}$ for some fixed $\varepsilon > 0$).

The distinguishing advantage. The standard cryptographic convention requires *negligible* distinguishing advantage of $n^{-\omega(1)}$. Unfortunately, the collection $\mathcal{F}_{P,n,m}$ fails to achieve this regardless of the choice of the predicate P or the output length m. Indeed, observe that if the graph G has two identical hyperedges then $f_{G,P}(x)$ can be distinguished from a truly random string with constant advantage (of one half). Since a $1/\text{poly}(n)$-fraction of (n, m, d)-hypergraphs have two identical hyperedges, the collection $\mathcal{F}_{P,n,m}$ can be distinguished with a noticeable advantage of $1/\text{poly}(n)$. Keeping this limitation in mind, we will strive for $1/\text{poly}(n)$-pseudorandomness, or, better yet, try to prove statements of the form: For all but $1/\text{poly}(n)$ fraction of the hypergraphs (e.g., all hypergraphs which satisfy some expansion properties) the function $f_{G,P}$ is $(n^{-\omega(1)})$-pseudorandom. (We will see examples for both versions in the following sections.)

Pseudorandomness vs. One-wayness. From an algorithmic point of view, the task of distinguishing seems much easier than the task of inversion, and therefore pseudorandomness seems more fragile than one-wayness. This is especially true in the setting of low locality, since in this case even the task of avoiding simple regularities in the output is quite challenging. Indeed, there are some predicates and hypergraphs for which distinguishing (with constant advantage) is easy, but inversion seems hard. This is the case, for example, when the predicate is not perfectly balanced (i.e., $\Pr_w[P(w) = 1] \neq \frac{1}{2}$) or the dependencies hypergraph contains two identical hyperedges. Nevertheless, as we will see (in Sections 4.1 and 4.2), there are good reasons to believe that, for a proper choice of P, the collection $\mathcal{F}_{P,n,m}$ is pseudorandom.

4.1 Pseudorandomness from One-wayness

The work of [7] relates the pseudorandomness of random local functions to the difficulty of inverting them.

4.1.1 Using Weak Unpredictability

As a starting point, it is shown that one-wayness implies some form of (weak) unpredictability as defined below.

Definition 4.2 (Unpredictability)**.** *The collection $\mathcal{F}_{P,n,m}$ is* α unpredictable generator *(UG) if for every efficient adversary $\mathcal{A}$ and every index $i \in [m]$ we have that*

$$\Pr[\mathcal{A}(f, f(x)_{[1,i-1]}) = f(x)_i] < \alpha(n), \qquad \textit{where } f \xleftarrow{R} \mathcal{F}_{P,n,m}, x \xleftarrow{R} \{0,1\}^n.$$

In the following, we say that a predicate P is *sensitive* if at least one its coordinates i has full influence, i.e., flipping the value of the i-th variable always changes the output of P. Equivalently, $P(w)$ can be written as $w_i \oplus Q(w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_d)$ for some $(d-1)$-ary predicate Q.

The following theorem is proven in [7] (see Theorems. 4.1 and 5.1 for a stronger and more detailed statements).

Theorem 4.3 (**one-wayness $\Rightarrow$ unpredictability**)**.** *Let P be a d-ary predicate and assume the collection $\mathcal{F}_{P,n,m}$ is one-way.*

1. *If $m = \Omega_d(n)$, then $\mathcal{F}_{P,n,m}$ is also $(1-\varepsilon)$-UG for some positive constant $\varepsilon = \varepsilon(P) > 0$.*

2. *If P is sensitive then for every polynomial q the collection $\mathcal{F}_{P,n,m/q(n)^2}$ is $(\frac{1}{2} + c/q(n))$-UG, for some constant $c = c(d) > 0$.*

See Section 4.1.3 for a proof sketch. The two parts of the theorem complement each other. Part (1) applies to any predicate but yields only $(1-\varepsilon)$ unpredictability for a constant ε, while Part (2) applies only to sensitive predicates but yields stronger parameters, e.g., $(\frac{1}{2}+\varepsilon)$-unpredictability for inverse polynomial ε. Such a difference is somewhat expected since unbalanced predicates (for which Part (1) applies) can be predicated with success probability that is bounded away from $\frac{1}{2}$.

Theorem 4.3 has several interesting applications. First, it can be used to show that, for some predicates, the one-wayness of $\mathcal{F}$ with output length m, implies pseudorandomness for a related (shorter) output length m'.

Theorem 4.4 (**one-way $\Rightarrow$ PRG**). *For every constants $a > 1, b > 0$ and d, and every sensitive d-ary predicate P, if $\mathcal{F}_{P,n,n^{3a+2b}}$ is one-way then $\mathcal{F}_{P,n,n^a}$ is ε-pseudorandom for $\varepsilon = n^{-b}$.*

Observe that the distinguishing advantage ε is inverse polynomial and not negligible as per the standard cryptographic definitions. As already mentioned this is inherent for the collection $\mathcal{F}_{P,n,m}$. It is plausible to assume that when the hypergraph G is a good expander, ε can be taken to be negligible.

4.1.2 Constructions of local PRGs with large stretch

We can further use the above theorems to construct locally computable PRGs with large stretch. For the regime of linear stretch, it is shown in [7] how to construct locally computable PRG with linear stretch and negligible distinguishing advantage based on a locally computable $(1-\varepsilon)$-UG with constant unpredictability (i.e., $\varepsilon > 0$) and sufficiently large linear stretch.[6] Combined with the first part of Theorem 4.3, we derive the following theorem ([7, Theorem 1.2]).

Theorem 4.5 (**Local PRG with Linear Stretch**). *For every d-ary predicate P and $m = \Omega_d(n)$, if the collection $\mathcal{F}_{P,n,m}$ is one-way, then there exists a collection of locally computable ε-PRGs with linear stretch and negligible distinguishing advantage $\varepsilon = n^{-\omega(1)}$.*

Note that a d-local PRG with linear stretch $m = (1+c)n$ bits can be composed with itself a constant number of times to yield a PRG with an arbitrary linear stretch c' at the expense of increasing the locality to a larger constant d'.

In the regime of polynomial stretch, we do not know how to simultaneously achieve constant locality and negligible distinguishing advantage. Still, we can prove an "almost" tight result based on Theorem 4.4 and standard amplification techniques (see [7, Thm 1.3]).

Theorem 4.6 (**Almost Local PRG with Polynomial Stretch**). *Assume that $\mathcal{F}_{P,n,n^{1+\delta}}$ is one-way for some sensitive d-ary predicate P and some constant $\delta > 0$. Then, for every constant $c > 0$ the following hold:*

[6] Observe that in this regime Yao's transformation from unpredictability to pseudorandomness cannot be used since the unpredictability is at a constant level whereas Yao's transformation requires unpredictability of at least $\frac{1}{2} + o(1/n)$). Instead, the transformation employs locally computable randomness extractors, which are based on locally-computable low-bias generators. The latter will be discussed in Section 4.2.

- *There exists a PRG collection with distinguishing advantage n^{-c}, polynomial stretch n^c, and constant locality $d = d(c)$.*
- *There exists a PRG collection with negligible distinguishing advantage $n^{-\omega(1)}$, polynomial stretch n^c, and* any *(arbitrarily slowly growing) super-constant locality $d = \omega(1)$, e.g., $\log^\star n$.*

The existence of a locally computable PRG with polynomial stretch and negligible distinguishing advantage remains an interesting open question.

Question 4.7. *Does a PRG with constant locality, polynomial stretch, and negligible distinguishing advantage exist?*

It is known [11] that such a PRG essentially requires an explicit construction of constant-degree bipartite expander graphs which are highly-unbalanced – a problem which is currently open. For the case of PRG collections, it actually suffices to describe an efficiently samplable distribution over highly-unbalanced constant-degree graphs which puts all but a negligible fraction of its mass on good expanders. To the best of our knowledge, this problem is also open. One way to avoid this barrier, is to focus on a *non-uniform* construction.

4.1.3 Proof sketch of Theorem 4.3

Conceptually, we reduce unpredictability to one-wayness via the following approach: Suppose that we have an efficient *predictor* $\mathcal{A}$ for the function $f(x)$ sampled from $f \xleftarrow{R} \mathcal{F}_{P,n,m}$. Then we can collect information about x, and eventually invert the function $f_{G,P}(x)$, by invoking the adversary multiple times with respect to many *different* hypergraphs $G_1, \ldots, G_t$, which are all close variants of the original hypergraph G. We sketch the details for the first part of the theorem.

The basic procedure. Let us assume for now that the first input of P has full influence. In addition, let us assume for simplicity that $\mathcal{A}$ always attempts to predict the last-bit. Namely, the predictor $\mathcal{A}$ is given a random (n, m, d)-hypergraph G and the $(m-1)$-bit long prefix of the string $y = f_{G,P}(x)$, and it predicts with probability $\frac{1}{2} + \delta$ the last output $y_m = P(x_S)$, which corresponds to the (last) hyperedge $S = (i_1, \ldots, i_d)$. Given such a pair (G, y), let us replace the first entry i_1 of S (hereafter referred to as "pivot") with a random index $\ell \in [n]$, and then invoke $\mathcal{A}$ on the modified pair. If the predictor succeeds and outputs $P(x_{S'})$, then, by comparing this value to y_m, we get to learn whether the input bits x_ℓ and x_{i_1} are equal. Since the predictor may err, we can treat this piece of information as a noisy 2-LIN equation of the form $x_\ell \oplus x_{i_1} = b$ where $b \in \{0, 1\}$.

Collecting many 2-LIN equations. In order to recover x, we would like to collect many such equations. To this end, we iterate the basic procedure n times where the hypergraph G, the m-bit string y, and the hyperedge S are all *fixed*, and in each iteration a different index $j \in [n]$ is being planted in S. As a result we obtain a system of noisy linear equations of the form

$$\begin{cases} y_1 = x_1 + x_{i_1}, \\ \quad \vdots \\ y_n = x_n + x_{i_1}, \end{cases}$$

where i_1 is the pivot. By guessing the value of x_{i_1} we derive a solution $x' \in \{0,1\}^n$, which agrees (*whp*) with x on $\frac{1}{2} + \delta'$ of the coordinates for some constant $\delta' > 0$. At this point we employ self-correction (Theorem 3.8) and recover x.

Handling general predicate. We would like to use the "basic procedure" for an arbitrary predicate that is not necessarily sensitive. For concreteness, think of the majority predicate. In this case, when recovering a 2-LIN equation, we are facing two sources of noise: one due to the error of the prediction algorithm, and the other due to the possibility that the current assignment x_S is "stable" (i.e., flipping its first location does not change the value of the predicate, e.g., in the case of majority, any assignment with less than $\lfloor d/2 \rfloor$ ones.) To bypass this problem we strengthen our hypothesis and require a predictor whose success probability is larger than the probability of getting a stable assignment. With some care, the previous argument can be extended to this case, and one can prove $(1-\varepsilon)$-unpredictability for some positive constant $\varepsilon = \varepsilon(P)$.[7]

Finally, we mention that the second part of the theorem requires additional noise-reduction steps, since it is based on a *weaker* $(\frac{1}{2} + 1/\mathrm{poly}(n))$-predictor. See [7] for details.

4.2 Linear Distinguishers

Let us shift gears and examine the security of $\mathcal{F}_{P,n,m}$ against linear distinguishers. Pseudorandomness against linear distinguishers means that there is no subset of output bits whose XOR has noticeable bias [55]. A bit more formally, for a function $f : \{0,1\}^n \to \{0,1\}^m$, we let

$$\mathsf{bias}(f) = \max_{L} \left\{ \left| \Pr_{x \xleftarrow{R} \{0,1\}^n} [L(f(x)) = 1] - \Pr_{y \xleftarrow{R} \{0,1\}^m} [L(y) = 1] \right| \right\}, \tag{5}$$

where the maximum is taken over all non-constant linear functions $L : \mathbb{F}_2^m \to \mathbb{F}_2$. A small-bias generator is a function f for which $\mathsf{bias}(f)$ is small (preferably negligible) as a function of n. Having a small bias is a definite necessary condition for achieving pseudorandomness as in Definition 4.1. Small-bias generators are also motivated by their own right, and they are used as building blocks in constructions that give stronger forms of pseudorandomness. This includes constructions of local cryptographic pseudorandom generators [13, 7], as well as pseudorandom generators that fool low-degree polynomials [25, 52, 62], small-space computations [45], and read-once formulas [22].

We are interested in understanding which d-local functions $f_{G,P} : \{0,1\}^n \to \{0,1\}^m$, described by a hypergraph G and a predicate P, have low-bias. This question was initiated by Cryan and Miltersen [30] and was further studied in a sequence of works [54, 8, 9]. To get some intuition, let us begin with few concrete examples.

4.2.1 Warm-up: Noisy-XOR

Consider the *noisy-XOR predicate* P that computes the parity of d inputs and XORs the result with a fresh noise bit e that is taken to be 1 with some constant probability $\delta < \frac{1}{2}$. This randomized predicate has internal randomness, which is chosen in each invocation independently of the input.

[7] It is shown in [7] that the actual bound on ε depends on a special measure of "matching" sensitivity $\mu(P)$ defined as follows: Look at the subgraph of the d-dimensional hypercube whose nodes are the sensitive assignments of P (i.e., the boundary and its neighbors), let M be a largest matching in the graph, and let $\mu(P) = |M|/2^d$. For example, for majority with an odd arity d, it can be shown that all the assignments of Hamming weight $\lceil d/2 \rceil$ and $\lfloor d/2 \rfloor$ are in the matching and so the matching sensitivity is exactly $2 \cdot \binom{d}{\lfloor d/2 \rfloor}/2^d = \Theta(1/\sqrt{d})$.

No noise. Let us begin with the simple case where there is no noise (i.e., $\delta = 0$). Since $m > n$, there must be a (non-empty) set of linearly-dependent outputs, and so the corresponding linear test always outputs zero. We conclude that $f_{G,P}$ has a high bias (at least $\frac{1}{2}$). Let us therefore assume from now on that the noise rate δ is some positive constant smaller than $\frac{1}{2}$.

$\mathbf{d = 1}$. Another simple case arises when the arity d is 1 (and $\delta \in (0, \frac{1}{2})$). In this case, there must be two outputs y_i and y_j which depend on the same input x_ℓ, and so the corresponding linear test $y_i \oplus y_j$ is 1 with probability $2\delta \cdot (1 - \delta) < \frac{1}{2}$. Since the noise rate is a constant, the function $f_{G,P}$ has a high bias, which is bounded away from 0.

$\mathbf{d = 2}$. We move on to the case of arity $d = 2$. In this case, we can view the dependencies hypergraph G as a *simple* graph over the input variables in which each edge (i, j) corresponds to an output $y_{i,j} = x_i + x_j + e_{i,j}$ and $e_{i,j}$ is a Bernoulli random variable with expectation δ. Let $T = (i_1, i_2, \ldots, i_{t-1}, i_1)$ be a shortest cycle in G, and let t denote its length. Consider the corresponding linear test

$$y_{i_1,i_2} + y_{i_2,i_3} + \cdots + y_{i_{t-1},i_1} = (x_{i_1} + x_{i_2} + e_{i_1,i_2}) + (x_{i_2} + x_{i_3} + e_{i_2,i_3}) + \ldots + (x_{i_{t-1}} + x_{i_1} + e_{i_{t-1},i_1}).$$

Since the input variables cancel out, the outcome is simply the parity of t noise terms, and so the bias is $\exp(-\Omega(t))$. When there are $m = n + \Omega(n)$ edges, a random graph G will have, with constant probability, a cycle of constant length, and so the expected bias of $f_{G,P}$ (over the choice of G) is constant.

$\mathbf{d \geq 3}$. Next, consider the case of $d \geq 3$. We argue that, in the typical case where G is a good expander, the bias of $f_{G,P}$ is negligible. In particular, we say that a set T of hyperedges in G has a uniquely covered node if there exists a node that participates in a single hyperedge $S \in T$. We will show that if any set T of at most k hyperedges in G has a uniquely covered node, then the bias of $f_{G,P}$ is $\exp(-\Omega(k))$. We will later discuss typical values of the parameter k (which is sometimes referred to as the *unique expansion* parameter).

Fix some set of outputs $T \subseteq [m]$. The outcome of the corresponding linear test $\sum_{i \in T} y_i$ (which is a random variable induced by the choice of $x \xleftarrow{R} \{0,1\}^n$ and the outcome of the noise) can be written as

$$\sum_{i \in T} y_i = \sum_{i \in T} \left(e_i + \sum_{j \in S_i} x_j \right), \tag{6}$$

where the e_i's are independent Bernoulli variables with mean δ, and the S_i's are the hyperedges of G. Following [54], we distinguish between *light* linear tests, which depend on less than k outputs, and *heavy* tests, which depend on more than k outputs. (Recall that k is the unique expansion parameter.)

If the test is light, T has a uniquely covered input node j and so we can write (6) as $x_j + z$ where z is statistically independent of x_j. Since x_j is a random bit, the outcome of the test is perfectly balanced. On the other hand, if T is heavy, we can write (6) as $\sum_{i \in T} e_i + z$ where z is statistically independent of the noise terms (the e_i's). Since the e_i's are independent Bernoulli variables, the bias of the sum drops exponentially with $t \geq k$.

Overall, the bias of $f_{G,P}$ is $\exp(-\Omega(k))$. For every $d \geq 3$ and $m = O(n)$, a random (n, m, d)-hypergraph is likely to achieve k-unique expansion for some $k \geq \Omega(n)$, and so, in this regime, $f_{G,P}$

is likely to have exponentially small bias. Similarly, it can be shown that in the polynomial regime $m = \text{poly}(n)$, the bias of $f_{G,P}$ is likely to sub-exponential (i.e., $\exp(-n^{\delta})$ for some constant $\delta > 0$ which depends on m and d).

4.2.2 Other Predicates

Interestingly, the noisy-linear predicate reveals an all-or-nothing behavior. The bias of $f_{G,P}$ is either large for *almost all* hypergraphs (for $d \leq 2$ or $\delta = 0$), or negligible for *almost all* hypergraphs (for $d \geq 3$). It turns out that, for some range of parameters, all predicates can be classified along these lines while revealing a similar all-or-nothing behavior. Formally, we call a predicate P *degenerate* if is either unbalanced, 1-correlated, 2-correlated, or affine over $\mathbb{F}_2$. (Recall that a predicate is c-correlated if it is correlated with the parity of c of its inputs.) The following theorem is proven by extending the ideas presented in Section 4.2.1.

Theorem 4.8 (a dichotomy [9])**.** *Let $m = n^{1+\varepsilon}$ where $\varepsilon \in (0, 1/4)$ is an arbitrary constant.*

1. *If P is degenerate, then,* whp*, the bias of $f \xleftarrow{R} \mathcal{F}_{P,n,m}$ is noticeable.*
2. *If P is non-degenerate, then,* whp*, the bias of $f \xleftarrow{R} \mathcal{F}_{P,n,m}$ is negligible.*

The theorem validates the intuition that, for sufficiently large output length, all hypergraphs are essentially the same, and hardness (or easyness) solely depend on the choice of predicate. The existence of a similar dichotomy for larger output lengths remains an interesting open question. It is known that the above notion of non-degeneracy does not imply negligible bias for output lengths larger than $m = n^{3/2}$. We also mention that there are explicit constructions of ε-biased generators with constant locality, linear output length $m = n + \Omega(n)$, and negligible bias $\varepsilon = n^{-\omega(1)}$. (See [13, Section 5.4].)

Proof idea. The first part is proved similarly to the case of noisy linear predicate of arity $d = 1$ or $d = 2$. For the second part, we follow the same strategy used for the noisy-XOR predicate and $d \geq 3$. That is, we distinguish between light tests and heavy tests. Note that non-degenerate predicates satisfy two forms of "non-linearity": First, it is required that P is uncorrelated with any linear function that involves less than 3 variables (called 2-*resiliency*); and second the algebraic degree of P as a polynomial over $\mathbb{F}_2$ should be at least 2 (called *degree* 2). It turns out that 2-*resiliency* allows to fool light tests and *degree* 2 allows to fool heavy tests.

Light Tests. In the previous section we essentially showed that if the predicate is the parity predicate $\oplus$ and the hypergraph is a good expander, the output of $f_{G,\oplus}(x)$ *perfectly* fools all light linear tests. In terms of expectation, this can be written as

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

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

More precisely, let ξ denote the outcome of the test $L(f_{G,P}(x))$. Then, by looking at the Fourier expansion of the predicate P, we can write ξ as a convex combination over the reals of many summands of the form $\xi_i = L(f_{G_i,\oplus}(x))$ where the hypergraphs G_i are subgraphs of G. (The exact structure of G_i is determined by the Fourier representation of P.) When x is uniformly chosen,

the random variable ξ is a weighted sum (over the reals) of many dependent random variables ξ_i's. It is shown that if G has sufficiently high vertex expansion (every not too large set of hyperedges covers many vertices) then the expectation of each summand ξ_i is zero, and so, by the linearity of expectation, the expectation of ξ is also zero.

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

Heavy Tests. Consider a heavy test that involves $t \geq k$ outputs. Switching back to zero-one notation, assume that the test outputs the value $\xi = P(x_{S_1}) + \ldots + P(x_{S_t}) \pmod 2$ where $x \xleftarrow{R} \{0,1\}^n$. Our goal is to show that ξ is close to a fair coin. For this it suffices to show that, for a sufficiently large ℓ, the sum ξ can be rewritten as the sum (over $\mathbb{F}_2$) of ℓ random variables

$$\xi = \xi_1 + \ldots + \xi_\ell \pmod 2, \tag{7}$$

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

In order to partition ξ, let us look at the hyperedges $S_1, \ldots, S_t$ that are involved in the test. As a first attempt, let us collect ℓ distinct "independent" hyperedges that do not share a single common variable. Renaming the independent hyperedges by $T_1, \ldots, T_\ell$ and letting $S_{\ell+1}, \ldots, S_t$ denote all other hyperedges, we can write ξ as

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

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

To prove the latter, we rely on the fact that P has algebraic degree 2. Specifically, let us assume that S_j and T_i have no more than a single common input node. (This condition can be typically met at the expense of throwing a small number of the T_i's.) In this case, the random variable $\xi_i = P(x_{T_i}) + \sum_j P(x_{S_j})$ cannot be constant, as the first summand is a degree 2 polynomial in x_{T_i} and each of the last summands contain at most a single variable from T_i. Hence, ξ_i is a non-trivial polynomial whose degree is lower-bounded by 2. This completes the argument. □

4.2.3 The Power of Local Low-Bias Generators

Generally speaking, security against linear distinguishers provides no guarantee against more general attackers. Furthermore, it is easy to construct generators which defeat linear distinguishers but completely break down by, say, degree-2 distinguishers. However, the case of local functions with large stretch may be qualitatively different. Specifically, we are not aware of any local function $f_{G,P}$

with *linear* stretch that fools linear distinguishers but can be distinguished by some polynomial-time adversary.[8] One may conjecture that if $\mathcal{F}_{P,n,m}$ fools linear adversaries for $m = n + \Omega(n)$ and for most hypergraphs, then it also fools polynomial-time adversaries. In other words, local functions are too simple to "separate" between these two different notions.

Question 4.9. *Is there a predicate P and output length $m = n + \Omega(n)$ for which $\mathcal{F}_{P,n,m}$ is $o(1)$-biased (as per Eq. (5)) but is not $o(1)$-pseudorandom (as per Def. 4.1)? In stronger form: Is there a predicate P and an (n, m, d) graph G where $m = n + \Omega(n)$, for which $f_{G,P}$ is $o(1)$-biased but is not $o(1)$-pseudorandom?*

While a positive answer only requires a "counter-example", an unconditional negative answer seems to be out of reach. (Indeed, by Theorem 4.8, there are predicates for which $\mathcal{F}_{P,n,n+\Omega(n)}$ has low bias, and so, a negative answer requires proving that $\mathcal{F}_{P,n,m}$ is $o(1)$-pseudorandom.) Nevertheless, one can gather evidence towards a negative answer by showing that if $\mathcal{F}_{P,n,m}$ has low-bias, then it fools wider classes of limited distinguishers. In fact, such a result was already proven for several interesting classes of adversaries. In the following, we say that a function $f : \{0,1\}^n \to \{0,1\}^m$ *ε-fools* a class of adversaries $\mathcal{A} = \{A : \{0,1\}^m \to \{0,1\}\}$ if for every $A \in \mathcal{A}$,

$$\left| \Pr_{x \xleftarrow{R} \{0,1\}^n} [A(f(x)) = 1] - \Pr_{y \xleftarrow{R} \{0,1\}^m} [A(y) = 1] \right| \leq \varepsilon.$$

If $\varepsilon = 0$, we say that *f perfectly fools $\mathcal{A}$*. Observe that ε-pseudorandomness coincides with ε-fooling the class of all efficient algorithms (cf. Def. 4.1).

Proposition 4.10. *Let $f : \{0,1\}^n \to \{0,1\}^m$ be a d-local function of bias 2^{-kd}. Then, f*

1. *perfectly fools the class of k-wise adversaries $A_{g,K} : y \mapsto g(y_K)$, that is all functions that project y to an arbitrary k-subset $K \subset [m]$ and apply an arbitrary function $g : \{0,1\}^k \to \{0,1\}$ to the result.*
2. *$\exp(n^{-\alpha})$-fools the class of functions computed by $\mathbf{AC^0}$ circuits of constant depth r and sub-exponential size $\exp(n^{\beta})$, provided that $(\alpha + \beta)r = O(\frac{\log k}{\log n})$.*
3. *$1/\mathrm{poly}(k)$-fools degree-2 threshold functions $A_p : y \mapsto \mathrm{sgn}(p(y))$ where p is a degree-2 polynomial over the reals and y is viewed as a vector in $\{\pm 1\}^n$.*

Perfect security against k-wise adversaries (also known as k-wise independence) is an important and well-studied notion of pseudorandomness. Constant depth circuits of unbounded fan-in ($\mathbf{AC^0}$) capture a large class of natural computation and is essentially the maximal circuit class against which we currently have (unconditional) pseudorandom generators. Finally, although degree-2 threshold functions (over the reals) may seem somewhat artificial, they actually cover some natural counting tests such as computing the Hamming weight of the given string (or its variance) and accepting if it exceeds some threshold.

Proof. (1) Assume towards a contradiction that there exists a k-wise adversary $A_{g,K} : y \mapsto g(y_K)$ with positive distinguishing advantage. Then, there exists a linear function $L : \mathbb{F}_2^k \to \mathbb{F}_2$ that distinguishes between $f(\mathcal{U}_n)_K$ and $\mathcal{U}_k$, where $\mathcal{U}_n$ denotes the uniform distribution over n-bit strings.

[8] Such functions exist when the stretch is sub-linear, i.e,. $m = n + o(n)$.

Namely, $|\Pr[L(f(\mathcal{U}_n)_K) = 1] - \Pr[L(U_k) = 1]| \neq 0$. Since the function is d-local, the distribution $f(\mathcal{U}_n)_K$ is sampled by less than kd random bits, and so the distinguishing advantage of L is larger than $2^{-kd} \geq \varepsilon$. It follows that f is not ε-biased in contradiction to the hypothesis.

The other two parts follow immediately from Part (1) by plugging in known results about k-wise independent distributions. Specifically, Part (2) follows from Braverman's theorem [26], and Part (3) follows from the works of Diakonikolas et al. [31, 32]. □

Additionally, in [9, Lemma 5.2] it is shown that, for random local functions, low-bias implies robustness (as in Section 3.1). It will be interesting to extend these implications to other classes of adversaries. A natural starting point would be to show that $\mathcal{F}_{P,n,m}$ fools low-degree polynomials over $\mathbb{F}_2$. (Such a result was already proven in [8] for the special case of the noisy-XOR predicate.)

Question 4.11. *For which predicates P and output lengths m, the collection $\mathcal{F}_{P,n,m}$ $o(1)$-fools degree-2 polynomials over $\mathbb{F}_2$?*

Note that the *existence* of locally computable generators which fool polynomials of constant degree $d = O(1)$ already follows from Viola's theorem [62] (i.e., by XOR-ing d independent locally-computable ε-bias generators). Here, we conjecture that a *random local function* is likely to fool low-degree polynomials.

4.3 Distinguishing vs. Approximation/Refutation

We move on to discuss the security of $\mathcal{F}_{P,n,m}$ against distinguishers which are based on CSP approximation and refutation algorithms. Let us begin with some standard terminology. For a d-ary predicate P, let $\mathsf{CSP}(P)$ denote the collection of all CSP instances in which each constraint is of the form $P(x_S) = 1$ where S is a d-tuple of distinct variables.[9] The *value* of a CSP instance Φ is the maximum fraction of satisfiable constraints taken over all possible assignments. There are two well studied probability distributions over $\mathsf{CSP}(P)$ instances: In the *planted distribution* $\mathbb{P}_{n,m}(P)$ an instance is generated by first picking at random a truth assignment $x \xleftarrow{R} \{0,1\}^n$, and then choosing m random d-ary tuples $S_1, \ldots, S_m$ for which $P(x_{S_i}) = 1$; whereas in the *uniform distribution* $\mathbb{U}_{n,m}(P)$ an instance is generated by choosing each of the d-ary tuples $S_1, \ldots, S_m$ uniformly at random.

It turns out that, if $\mathcal{F}_{P,n,m}$ is pseudorandom, the planted and uniform distributions are indistinguishable. (Through this section we write pseudorandom and indistinguishable to denote $o(1)$-pseudorandom and $o(1)$-indistinguishable.)

Proposition 4.12. *If $\mathcal{F}_{P,n,m}$ is pseudorandom, then $\mathbb{P}_{n,m/3}(P)$ and $\mathbb{U}_{n,m/3}(P)$ are indistinguishable.*

Proof sketch. By the hypothesis, it is hard to distinguish between a random pair

$$(G, y) \qquad \text{where } G \xleftarrow{R} \mathcal{G}_{n,m,d}, y \xleftarrow{R} \{0,1\}^m \tag{i}$$

and a pseudorandom pair

$$(G, f_{G,P}(x)) \qquad \text{where } G \xleftarrow{R} \mathcal{G}_{n,m,d}, x \xleftarrow{R} \{0,1\}^n. \tag{ii}$$

[9] Typically, one also allows *negated* variables (literals). The results of this section apply to this variant as well, provided that the collection $\mathcal{F}_{P,n,m}$ is modified accordingly. Namely, P is applied to (randomly chosen) d-tuples of *literals*.

As in Section 3 (Eq. 1), we may view the pair (G, y) as a constraint satisfaction problem $\Phi_{G,y,P}$ over n variables $x = (x_1, \ldots, x_n)$ and m constraints of the form $P(x_{S_i}) = y_i$ where S_i is the i-th hyperedge of G. By keeping the first $m' = m/3$ constraints for which $y_i = 1$ we obtain a $\mathsf{CSP}(P)$ instance with m' constraints. (Such a number of 1-constraints exists, *whp*, as otherwise one can easily distinguish a random instance from a pseudorandom instance.) When the input is pseudorandom the resulting instance is distributed according to the planted distribution, whereas a random input is mapped to the uniform distribution. Therefore, a distinguisher between $\mathbb{P}_{n,m'}(P)$ and $\mathbb{U}_{n,m'}(P)$ contradicts the indistinguishability of (i) and (ii).[10] □

In the superlinear regime $m = \omega(n)$, the distribution $\mathbb{U}_{n,m}(P)$ is concentrated over instances whose value equals to the density of the predicate $\rho(P) \stackrel{\text{def}}{=} |P^{-1}(1)|/2^d$. These are essentially the most unsatisfiable instances since any $\mathsf{CSP}(P)$ instance achieves this value, e.g., by a random assignment. So breaking the pseudorandomness of $\mathcal{F}_{P,n,m}$ boils down to distinguishing, in the average-case, between satisfiable CSP instances and highly unsatisfiable instances. (Compare this to one-wayness where inversion translates to finding a satisfying assignment.)

The CSP literature typically considers two (weaker) related algorithmic tasks: *Approximation* and *Refutation*.

Approximation An α-*approximation* algorithm takes a CSP instance $\Phi \in \mathsf{CSP}(P)$ and outputs a (possibly randomized) assignment x which satisfies (in expectation) at least $\alpha \cdot \mathsf{val}(\Phi)$ fraction of the constraints, where $\mathsf{val}(\Phi)$ is the maximal fraction of the constraints that can be satisfied. So an approximation algorithm provides a *lower-bound* on the value of Φ. A predicate P is *approximation resistant* [44] if all efficient algorithms fail to achieve (in the worst-case) an approximation ratio better than the trivial ratio $\rho(P)$ achieved by the naive random assignment algorithm.

Refutation A *refutation* algorithm $\mathcal{A}$ for $\mathsf{CSP}(\mathcal{P})$ computes an *upper-bound* on the value of an instance $\Phi \in \mathsf{CSP}(P)$, i.e., $\mathcal{A}(\Phi) \geq \mathsf{val}(\Phi)$ for every $\Phi \in \mathsf{CSP}(P)$. When $\mathcal{A}$ outputs a non-trivial value (smaller than 1), it certificates that Φ is unsatisfiable.

So approximation/refutation algorithms can be viewed as one-sided distinguishers between satisfiable instances and highly-unsatisfiable instances. In light of Proposition 4.12, a necessary condition for the pseudorandomness of $\mathcal{F}_{P,n,\omega(n)}$ is that both approximation and refutation are intractable in the *average case*.

Corollary 4.13. *Assume that $\mathcal{F}_{P,n,m}$ is $o(1)$-pseudorandom for $m = \omega(n)$. Then, for every efficient refutation algorithm $\mathcal{A}$ and $m' = m/3$ we have*

$$\Pr_{\Phi \stackrel{R}{\leftarrow} \mathbb{U}_{n,m'}(P)} [\mathcal{A}(\Phi) < 1] < o(1). \tag{8}$$

Also, for every efficient approximation algorithm $\mathcal{B}$ we have

$$\Pr_{\Phi \stackrel{R}{\leftarrow} \mathbb{P}_{n,m'}(P)} [\mathcal{B}(\Phi) > \rho(P)] < o(1), \tag{9}$$

where we abuse notation and let $\mathcal{B}(\Phi)$ denote the fraction of constraints that are satisfied by $\mathcal{B}$.

[10] A tighter analysis allows to reduce the overhead m/m' from 3 to 2.

Proof. By definition, $\mathsf{val}(\mathbb{P}_{n,m'}(P)) = 1$ and so a refutation algorithm $\mathcal{A}$ which does not satisfy (8) can be used to distinguish $\mathbb{P}_{n,m'}(P)$ from $\mathbb{U}_{n,m'}(P)$, contradicting Proposition 4.12. Similarly, since $\mathsf{val}(\mathbb{U}_{n,m'}(P)) = \rho(P)$ (with all but negligible probability), an approximation algorithm $\mathcal{B}$ which does not satisfy (9) allows us to distinguish between $\mathbb{P}_{n,m'}(P)$ and $\mathbb{U}_{n,m'}(P)$. □

So in order to argue that $\mathcal{F}_{P,n,\omega(n)}$ is pseudorandom, one should first show that P is approximation resistant (on random satisfiable instances) as well as hard-to-refute (on random unsatisfiable instances).[11] As we will see next, such hardness results are believed to hold for *non-degenerate predicates* as defined in Section 4.2.2. In fact, such (worst-case) hardness results apply to a larger class of predicates called *pairwise independent*.

A predicate P is pairwise independent if there is a distribution μ over the set of accepting assignments $P^{-1}(1) \subseteq \{0,1\}^d$ such that $(w_1, \ldots, w_d) \stackrel{R}{\leftarrow} \mu$ is pairwise independent, i.e., for every $i \neq j$ the pair (w_i, w_j) is uniform over $\{0,1\}^2$. It is not hard to verify that non-degenerate predicates satisfy this condition with respect to the uniform distribution over satisfying assignment. In [15] it is shown that, under Khot's unique-game conjecture [50], pairwise independent predicates are approximation resistant. Pairwise independent predicates seem also hard-to-refute. Specifically, a notable class of refutation algorithms work by relaxing Φ to a linear program (LP) or semi-definite program (SDP), and use the value of the program as a lower-bound to the value of Φ. It is known that relatively powerful families of LP/SDP-based refutation algorithms fail to refute $\mathbb{U}_{n,m=\omega(n)}(P)$ when P is pairwise independent [19, 61, 16]. In fact, it is conjectured in [17] (following Feige [34]) that no efficient algorithm can beat these SDP-based algorithms over the uniform distribution, and so pairwise predicates are "hard-to-refute" over the uniform distribution.

Pseudorandomness as a working hypothesis . Reversing the viewpoint, we may consider the pseudorandomness of $\mathcal{F}_{P,n,m}$ as a *working hypothesis* which directly implies the intractability of approximation and refutation in the average case. The latter assumption was used by Feige [34] to derive inapproximability results for combinatorial problems. Since pseudorandomness seems strictly stronger, one may hope to further improve these implications. Indeed, it is shown in [7] that, assuming that $\mathcal{F}_{P,n,m}$ is pseudorandom, the densest-subgraph problem in d-uniform hypergraphs cannot be approximated to within a polynomial factor of n^{ε}, improving the constant inapproximability result from [34, 51]. It seems likely that the pseudorandomness of $\mathcal{F}_{P,n,m}$ can lead to further implications.

4.4 Pseudorandomness: Summary

Large output lengths. Theorem 4.4 suggests that pseudorandomness (with inverse polynomial distinguishing advantage) can be achieved with essentially the same parameters as one-wayness. Specifically, assuming that Conjecture 3.9 holds with respect to sensitive predicates, for every polynomial $m(n) = n^c$ and inverse polynomial $\varepsilon(n)$, there exists a predicate P of constant arity d for which $\mathcal{F}_{P,n,m}$ is ε-pseudorandom. As we saw (in Theorems 3.4 and 3.5), such a predicate P must have an algebraic degree of c and must satisfy $\lfloor 2c \rfloor$-resiliency.[12] For small values of $c < 5/4$, this condition seems to be sufficient as hinted by hardness results for linear distinguishers (Theorem 4.8) and approximation/refutation based distinguishers (Section 4.3). Even under Conjecture 3.9, understanding which predicates achieve pseudorandomness for larger output lengths is an interesting

[11] A similar result holds for the complement of P, as $\mathcal{F}_{1-P,n,m=\omega(n)}$ is also pseudorandom.

[12] Recall a-resiliency means that P is not correlated with parity of arity smaller or equal to a

open question. Recent results suggest that $2c$-resiliency defeats powerful families of adversaries including SDP-based refutation algorithms [57] and "statistical query" algorithms [36]. However, it is currently unknown what properties are needed to guarantee low-bias for $\mathcal{F}_{P,n,m=n^c}$ when c is a large constant.[13]

Bibliographic note. Many of the results in this section were first proven for the special case of the noisy-XOR predicate. Feige conjectured that random 3-XOR formulas are hard to refute (and proved equivalence to the irrefutability of other 3-ary predicates). Following Feige, Alekhnovich [4] suggested to assume that the collection $\mathcal{F}_{P,n,\Theta(n)}$, instantiated with noisy 3-XOR, is pseudorandom, and observed that this implies Feige's hypothesis.[14] The extension of this observation to the case of general predicates (Proposition 4.12) is new to this survey.

Alekhnovich's construction does not lead directly to a local PRG (due to the use of noise), however it was derandomized in [13] yielding the first construction of local PRG with linear stretch. The pseudorandomness of the collection $\mathcal{F}_{P,n,m}$ with noisy-XOR was reduced by [8] to the one-wayness of the same collection (with some loss in the parameters) establishing a special case of Theorem 4.4.

The first construction of local low-bias generator was given in [54]. Specifically, it was shown that the predicate $\mathsf{MST}_{3,2}$ combined with a specific hypergraph G, achieves low-bias.[15] In [8] this result was extended to *random* local functions, by obtaining a sufficient condition over the predicate P that guarantees low bias for most functions in $\mathcal{F}_{P,n,n^{1+\varepsilon}}$ for some small constant $\varepsilon > 0$. Later, a full classification of good predicates (Theorem 4.8) was established in [9].

5 Target Collision Resistance

We saw that it is possible to locally stretch a short seed into a long pseudorandom string. Is it also possible to locally hash a long string into a shorter one? Specifically, a collection of *universal one-way hash functions* [56] $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ shrinks a long n-bit string into a shorter string of length $m < n$ such that, given a random function $h \xleftarrow{R} \mathcal{H}$ and a preselected target string x, it is hard to find a sibling $y \neq x$ that collide with x under h. We focus on the regime of linear shrinkage where the function shrinks an n-bit input into $(1-\varepsilon)n$-bit output for some constant $\varepsilon > 0$.

5.1 Hashing via Random Local Functions?

As a starting point, we ask whether the collection $\mathcal{F}_{P,n,(1-\varepsilon)n}$ itself can be used, even heuristically, as a UOWHF. Unfortunately, the answer turns to be negative. Since every output is connected to a random d-tuple of inputs, a function $f \xleftarrow{R} \mathcal{F}_{P,n,(1-\varepsilon)n}$ is likely to have some input i of degree 0,

[13] The results of [54] show that there exists a predicate P of arity $d = O(c^2)$ and a *concrete* $(m = n^c, n, d)$-hypergraph G, for which $f_{G,P}$ has negligible bias. (The resulting construction is not uniform as it relies on bipartite expander graphs with no explicit construction .)

[14] More precisely, Alekhnovich's conjecture asserts that, for a random $f \xleftarrow{R} \mathcal{F}_{\oplus,n,m=\Theta(n)}$, the sequence $f(x)+e$ is indistinguishable from the sequence $f(x)+e+e_i$ where $x \xleftarrow{R} \{0,1\}^n$, e is a random vector of weight pm, and e_i is a random vector of weight 1. This assumption is (trivially) equivalent to the unpredictability of $f(x)$ which, by Yao's theorem, implies that $f(x)+e$ is pseudorandom. The latter assumption was shown to imply pseudorandomness with respect to iid noise in [13, Appendix A].

[15] An explicit version of this construction was given by [13] for the linear output regime $m = n + \Theta(n)$.

with no influence at all. Hence, we can find a collision with a target string x simply by flipping the values of x_i.

To remedy the problem, let us try to modify the distribution of the input-output dependency hypergraph and consider only *regular* (uniform) hypergraphs in which each input affects exactly c outputs and each output depends on d inputs. Unfortunately it turns out that, in this modified ensemble, collision are still easy to find, even with respect to a random target string x. To see this, it is instructive to consider the case where P is the majority predicate. With high probability, there will be an input variable x_i such that all of its neighboring inputs (i.e., the inputs that share an output with x_i) are assigned zero. In this case, we can flip the *insensitive* input x_i without affecting the output of the function, and this way obtain a trivial collision. In fact, since each variable has at most $cd = O(1)$ neighbors and since x is chosen at random, every variable x_i has a constant probability of being insensitive, and one is likely to find an $\Omega(n)$ insensitive inputs. Furthermore, by collecting an independent set I of insensitive inputs (that do not share any common output) one can simultaneously flip any subset of the inputs in I without changing the output. Hence, there exist a ball of radius $\Omega(n)$ around x such that for every x' in this ball $f(x') = f(x)$. Finally, observe that a similar attack can be applied to $\mathcal{F}_{P,n,m}$ for every predicate P except for XOR or its negation. Unfortunately, in the latter case collisions can be found via Gaussian elimination.

5.2 Finding far-collisions

Despite the above failure, let us keep asking: Can $\mathcal{F}_{P,n,m}$ achieve some, possibly weak, form of collision resistance? Note that the previous attack yields collisions that are "not too far" from the target string x. That is, the relative Hamming distance is smaller than β for some constant β, which depends on the predicate P. It is therefore natural to consider the following notion of β *target collision resistance*.

Definition 5.1 ([14])**.** *A collection of length-decreasing functions $\mathcal{H}$ is δ-secure β-target collision resistance (TCR) if for every efficient adversary $\mathcal{A}$*

$$\Pr_{x \xleftarrow{R} \{0,1\}^n, h \xleftarrow{R} \mathcal{H}} [\mathcal{A}(\langle h \rangle, x) = y \text{ s.t. } h(x) = h(y) \text{ and } \Delta(x,y) > \beta] < \delta,$$

where $\Delta(x,y)$ denotes the relative Hamming distance between x and y, i.e., the fraction of coordinates where x and y differ.

The study of the *geometry of the solutions* of random constraint satisfaction problems [2] suggests, at least heuristically, that $\mathcal{F}_{P,n,m}$ may be secure with respect to β-far collisions. Thinking of each output as inducing a local constraint on the inputs, it can be essentially showed that, for under-constraint problems where $m = (1-\varepsilon)n$, the space of solutions (siblings of x) is shattered into far-apart clusters of Hamming-close solutions. It is believed that efficient algorithms cannot move from one cluster to another as such a transition requires to pass through strings x' that violate many constraints (i.e., $f(x')$ is far, in Hamming distance, from $f(x)$). Therefore, it seems plausible to conjecture that the collection $\mathcal{F}_{P,n,m}$ is secure with respect to β-far collisions.

It turns out that this conjecture can be proven assuming the pseudorandomness of $\mathcal{F}_{P,n,m'}$ (where $m' > n > m$).

Theorem 5.2 (Corollary 4.2 [14])**.** *There exists a predicate P, constants $\varepsilon, \beta \in (0, \frac{1}{2})$ and $c > 1$ such that the following holds. For every $\delta > 0$, if $\mathcal{F}_{P,n,cn}$ is $(\delta/5)$-pseudorandom then $\mathcal{F}_{P,n,(1-\varepsilon)n}$ is δ-secure β-TCR.*

The theorem holds even if δ decreases with n (as long as it is inverse polynomial), although we will employ it only with small constant values. We further mention that the theorem is constructive, and can be applied to every predicate that satisfies some explicit sensitivity criteria. In particular, it is shown that the MST predicate $\mathsf{MST}_{d_1,d_2}(x,y) = (y_1 \oplus \ldots \oplus y_{d_1}) \oplus (x_1 \wedge \ldots \wedge x_{d_2})$, satisfies the theorem for every $d_2 \geq 2$ and every sufficiently large odd constant d_1.

Proof idea. Let $m = (1-\varepsilon)n$. Let P be a balanced predicate, which, in addition, enjoys the following *sensitivity* properties:[16]

$$\forall x, x' \in \{0,1\}^n, \Delta(x,x') > \beta \Rightarrow \mathop{\mathbb{E}}_{f \xleftarrow{R} \mathcal{F}_{P,n,m}} [\Delta(f(x), f(x'))] > \gamma \qquad \text{for some constants } \beta, \gamma > 0$$

$$\forall x, x' \in \{0,1\}^n, \Delta(x,x') = \frac{1}{2} \Rightarrow \mathop{\mathbb{E}}_{f \xleftarrow{R} \mathcal{F}_{P,n,m}} [\Delta(f(x), f(x'))] = \frac{1}{2}.$$

An example of such a predicate is parity $\oplus_d$ with an odd arity d. A relaxation of the above properties (e.g., by considering only x of Hamming weight $\frac{1}{2}$) allows us to use MST_{d_1,d_2} for every $d_2 \geq 2$ and every odd constant d_1. (Larger d_1 pushes β towards zero and increases γ towards $\frac{1}{2}$.)

Assume that we have an algorithm $\mathcal{A}$ that, given a random function $h \xleftarrow{R} \mathcal{F}_{P,n,m}$ and a random target w, finds a β-far sibling with probability δ. Let us first try to use $\mathcal{A}$ to invert the collection $\mathcal{F}_{P,n,m'}$ with output length of $m' \approx 2m$. Given a random function $f_G \xleftarrow{R} \mathcal{F}_{P,n,m'}$ specified by a random input-output dependencies hypergraph G, and an image $y = f_G(x)$ of a random point $x \xleftarrow{R} \{0,1\}^n$, we will recover the preimage x as follows.

First, we choose a target w uniformly at random and partition the hypergraph G into two hypergraphs: $G_=$ which contains only the output hyperedges for which $f_G(w)$ agrees with y, and $G_\neq$ which contains the remaining hyperedges. Hence,

$$f_{G_=}(x) = f_{G_=}(w) \qquad \text{and} \qquad f_{G_\neq}(x) = \overline{f_{G_\neq}(w)},$$

where $\overline{z}$ denotes the bit-wise complement of the string z. Since P is balanced, each subgraph contains roughly m' hyperedges. Next, we ask $\mathcal{A}$ for a β-far sibling w' of w under the function $f_{G_=}$. As we will see next w' is likely to be *correlated* with the preimage x, in the sense that for some constant $\alpha > 0$, either w' or its complement $\overline{w'}$ agree with x on $(\frac{1}{2}+\alpha)$-fraction of their coordinates. Therefore, by employing the self-correction theorem (Theorem 3.8), we can use w' to fully recover x with the aid of $O(n)$ additional outputs.

It remains to show that w' is likely to be correlated with the preimage x. Using the sensitivity properties of the predicate P, this boils down to proving that $f_G(w')$ and $f_G(x)$ agree on $\frac{1}{2}+\alpha'$ of their coordinates, for some constant $\alpha' > 0$. Let us first (optimistically) assume that w' is *statistically independent* of the sub-graph $G_\neq$ that was not submitted to the adversary. That is, imagine that this part of the dependencies hypergraph is chosen uniformly at random after w' is obtained. Since w is β-far from w', this pair is expected to disagree on a constant fraction γ of the remaining coordinates of $f_{G_\neq}$. Namely,

$$\Delta(f_{G_\neq}(w), f_{G_\neq}(w')) > \gamma.$$

[16] It can be shown that the above properties are actually independent of the output length m, and corresponds to a generalized notion of noise sensitivity of P. See [14, Section 3].

Since $f_{G_{\neq}}(x) = \overline{f_{G_{\neq}}(w)}$ it follows that

$$\Delta(f_{G_{\neq}}(x), f_{G_{\neq}}(w')) < 1 - \gamma.$$

Furthermore, since w' collides with w under $f_{G_=}$ we have that

$$f_{G_=}(x) = f_{G_=}(w) = f_{G_=}(w').$$

We conclude that x and w agree on a fraction of $1 - \frac{1}{2}(1-\gamma) = \frac{1}{2} + \gamma/2$ of the outputs of f_G (γ-fraction of the coordinates of $f_{G_{\neq}}$ and all the coordinates of $f_{G_=}$).

The above argument is over-optimistic, since it is not clear that w' is statistically independent of the subgraph $G_{\neq}$. (Indeed, the adversary $\mathcal{A}$ chooses w' based on $(w, G_=)$ which contain some information on x and, therefore, also on $G_{\neq}$.) Fortunately, we can show that a failure of the above approach allows to distinguish the string $y = f_G(x)$ from a truly random string. Hence, we are in a win-win situation: we can either invert $\mathcal{F}$ by finding a correlated string, or we can distinguish its output from a random string. So the theorem can be based on the pseudorandomness of $\mathcal{F}_{P,n,n+\Omega(n)}$. □

5.3 Applications

Theorem 5.2 has several interesting applications. First, it is shown in [14] that it is possible to convert δ-secure β-target collision resistance, for constant parameters δ, β, into a standard UOWHF while preserving *constant* locality and *linear* shrinkage. (Interestingly the transformation makes use of two collections of random local linear functions $\mathcal{F}_{\oplus,n,m}$, where one collection is length-shrinking and one collection is length-expanding.) By combining this transformation with Theorems 5.2 and 4.4, we obtain a collection of locally computable UOWHFs $\mathcal{H}$ with linear shrinkage based on the one-wayness of random local functions.

Theorem 5.3 ([14])**.** *There exist a predicate P and a constant c for which the following holds. If $\mathcal{F}_{P,n,cn^3}$ is one-way then there exists a locally computable UOWHF $\mathcal{H}$ with linear shrinkage. In particular, the above holds for $P = \mathsf{MST}_{d_1,d_2}$ where d_2 is an arbitrary constant, and d_1 is a sufficiently large odd constant.*

The theorem can be used to speed-up the sequential-time complexity of several cryptographic primitives. First, by iterating $\mathcal{H}$ a logarithmic number of times we get a linear-time computable hash function $\mathcal{H}'$ with polynomial shrinkage factor of $m = n^{\varepsilon}$ for an arbitrary constant $\varepsilon > 0$ (the i-th level of the circuit contains $O(n/2^i)$ gates). As observed by [47], such a hash function can be used to obtain asymptotically-optimal signature schemes via the Hash-and-Sign paradigm. To sign an n-bit message x, first hash it to the length of the security parameter κ, and then use some standard signature scheme to sign the hashed value with complexity $\mathrm{poly}(\kappa)$. The overall computational cost is $O(n + \mathrm{poly}(\kappa))$ which is dominated by n for sufficiently long messages (e.g., when n is polynomially longer than the security parameter). Overall, the resulting signature scheme has only *additive cryptographic overhead*, while the complexity of standard signature schemes grows multiplicatively with the security parameter, i.e., $O(n \cdot \mathrm{poly}(\kappa))$.

Collision-Resistance Hashing. So far our discussion was restricted to the case of target collision resistance where the target string is independent of the function $f \stackrel{R}{\leftarrow} \mathcal{F}$. However, one may strive

for a stronger form of security where the target can be chosen based on f. Formally, we say that $\mathcal{F}$ is *collision resistance* if, given $f \xleftarrow{R} \mathcal{F}$, it is hard to find a pair of colliding inputs $x \neq x'$. Under standard intractability assumptions [12], there are collision-resistance hash functions with constant output locality and sublinear shrinkage of $m = n - n^{\varepsilon}$. We mention that the UOWHF constructions that results from Theorem 5.2 can be broken when the adversary is allowed to choose the target string. This leaves open the following question.

Question 5.4. *Is it possible to locally compute collision-resistance hash functions with linear shrinkage? Can such a construction be based on the one-wayness of $\mathcal{F}_{P,n,m}$?*

In fact, for the case of constant *input-locality* (where each input affects a constant number of outputs), the existence of collision-resistance hashing is open even for the case of 1-bit shrinkage. (A positive answer to Question 5.4 would settle this question as well, see [14, Lemma 5.8].)

6 Public-Key Cryptography

So far we considered private-key primitives, a more ambitious task is to try and use random local functions to implement public-key cryptography. In this case, the motivation is not only efficiency, but also *diversity*. Despite its importance, very few candidates for public-key encryptions are known, and these are based on a handful of computational problems of a very structured algebraic or geometric nature from the areas of number theory, lattices, and error-correcting codes (e.g., [33, 59, 53, 3]). Basing public-key primitives on the hardness of random local functions – an assumption with a combinatorial flavor – would allow to broaden and diversify the foundations of public-key cryptography.

One way to achieve this goal is to try and plant a secret "trapdoor" in $f_{G,P} \xleftarrow{R} \mathcal{F}_{P,n,m}$ that, if known, allows to break its one-wayness or pseudorandomness. In [8] such a trapdoor is obtained by planting a non-expanding set in the underlying dependencies hypergraph G. This minor violation of expansion is (hopefully) hard to detect, but still (when known) allows to break the security of the resulting function. Let us elaborate on a simplified variant of this idea.

Construction 6.1 (A Template for Public-Key Encryption)**.** *Consider the following high-level scheme (parameterized with integers $q < r < n < m$ and a predicate P):*

Key-generation *Choose a random $(n, m-q, d)$-hypergraph G_1, choose a small random r-subset of nodes $R \subseteq [n]$, plant a random (r, q, d)-hypergraph G_2 over the nodes in R, and combine the two graphs into a single (n, m, d)-hypergraph G' by collecting all hyperedges in a random order. Publish the combined hypergraph G' as the public-key and keep the locations $Q = (i_1, \ldots, i_q) \in [m]^q$ of the hyperedges of G_1 as the private-key (i.e., G_2 is simply the restriction of G' to the hyperedges in Q).*

Encryption *The bit "zero" is encrypted by $y = f_{G',P}(x)$ where $x \xleftarrow{R} \{0,1\}^n$.*
The bit "one" is encrypted by a random m-bit string $y \xleftarrow{R} \{0,1\}^m$.

Decryption *Let f' be the restriction of $f_{G',P}$ to the outputs in Q. Since these outputs depend only on the inputs in R, the function f' is a mapping from r-bit strings to q-bit strings. Assume that we have an efficient algorithm that checks whether a q-bit string is in the image of f'. In this case, we can decrypt a ciphertext $y \in \{0,1\}^m$ (with some error) by checking if the*

restricted ciphertext y_Q is in the image of f'. If y is in the image of f', decryption succeeds with probability 1, and when y is random, decryption errs with probability at most 2^{r-q} which is negligible if $q = r + \omega(\log n)$. (Alternatively, if $q = r + 1$, then the decryption error can be reduced by repeated encryptions.)

This general template still leaves two main questions:

- (Decryption) How to efficiently decide if a string is in the image of f'?
- (Security) How secure is the resulting scheme?

Several different answers are given in [8] depending on the exact choice of the predicate.

6.1 Noisy-XOR predicate

One suggestion is to let P be the noisy d-XOR predicate with noise rate of $\delta \ll 1/q$. In this case, the probability that none of the outputs in Q are noisy is at least $1 - q\delta \gg 0$ and so decryption can be implemented by Gaussian elimination (treating f' as a linear mapping). To prove that the scheme is secure one has to show that the mapping $f_{G',P}$ remains pseudorandom even when G' is selected according to the key-generation distribution. Observe that G' can be viewed as a random (n, m, d)-hypergraph G which was modified by replacing q of its hyperedges with q "planted hyperedges" which are all incident to a small random r-subset of nodes $R \subseteq [n]$. It is shown that, for a proper choice of the parameters, this modification, from G to G', keeps the mapping $f_{P,G'}$ pseudorandom. Indeed, when $d = 3, q = O(n^{0.2})$ and the output length m is $O(n^{1.4})$, the distributions of G' and G are not too far apart (the statistical distance is bounded away from 1), and, one can show that the hardness of $f_{P,G'}$ is "inherited" from the hardness of $f_{P,G}$. By taking the noise rate $\delta = 1/cq$, for some sufficiently large constant c, we derive the following theorem.

Theorem 6.2 (Corollary 5.6 in [8])**.** *Assume that $\mathcal{F}_{P,n,cn^{1.4}}$ is one-way, where P is the noisy 3-XOR predicate with noise rate $\frac{1}{cn^{0.2}}$, and c is some universal constant.*[17] *Then, there exists a public-key encryption scheme.*

Recall that, by Theorem 3.5, the noisy 3-XOR predicate can be successfully attacked for output length of $m = n^{3/2}$. For the parameters considered in Theorem 6.2, the best known attack has sub-exponential complexity of $\exp(n^{\varepsilon})$ for some constant $\varepsilon > 0$. As a partial evidence for hardness, we further mention that the related task of *refuting* (in the sense of Eq. (8)) a random system of $m = O(n^{1.4})$ 3-XOR equations with noise rate $n^{-0.2}$ is at least as hard as refuting random 3-CNF instance with m clauses [35]. The latter is a longstanding open problem for any $m \leq n^{3/2-\Omega(1)}$.

A variant. It is natural to ask whether Theorem 6.2 can be based on a seemingly more conservative assumption that uses a shorter output length (ideally linear in n) and a larger (constant) locality d. Indeed, most of the proof generalizes to the case of general d, δ and quasilinear output length $m = \Theta(n \log n)$, except that, for this setting of parameters, the distributions of G and G' are far-apart and we cannot employ the information-theoretic argument used in the proof of Theorem 6.2. This limitation can be solved at the expense of presenting an additional intractability

[17] For randomized predicates P, the one-wayness of $\mathcal{F}_{P,n,m}$ asserts that, for a random graph G and a random x, it is hard to recover x from $(G, y = f_{G,P}(x))$. For our setting of parameters, *whp*, x is uniquely determined by (G, y).

assumption called *Decisional Unbalanced Expansion* which essentially asserts that it is hard to distinguish a random (n, m, d)-hypergraph G from a random (n, m, d)-hypergraph G' with a random planted *shrinking* set of q hyperedges. We denote this assumption by $\mathsf{DUE}(n, m, d, q)$.

Theorem 6.3 (Corollary 6.5 in [8]). *Assume that, for some $q = q(n) = o(n)$, constant $d \in \mathbb{N}$ and constant $c > 0$, (1) the $\mathsf{DUE}(n, cn, d, q)$ assumption holds and (2) $\mathcal{F}_{P,n,cn\log n}$ is one-way, where P is the noisy d-XOR predicate with noise rate $1/cq$. Then, there exists a public-key encryption scheme.*

DUE parameters. The use of DUE induces some constraints on the parameters. Indeed, consider the set of nodes R on which the additional q hyperedges are planted. It turns out that if the hypergraph induced on R is much denser than the rest of the hypergraph then one can efficiently distinguish G from G'. For example, if the output-input ratio $c = m/n$ is sufficiently smaller than the degree d, we can distinguish between the two cases by just looking at the degree distribution. Indeed, d, the amount added to the degrees in R, is larger than the standard deviation of the average input degree in G, which is cd. Stronger variants of this attack (e.g., counting the number of short cycles which are incident to Q) apply to the regime where c is small enough so that $c^{\log_d q} \ll n$. Still the DUE problem is conjectured to be intractable when d is a small constant, $q = n^\delta$, for some small constant $\delta \in (0, \frac{1}{2})$ (e.g., $1/10$), and c is a constant which is much larger than $d^{1/\delta}$ (say $100d^{1/\delta}$). Some evidences for the validity of this assumption are given in [8]. Under this setting of parameters, it is conceivable to believe that the resulting public-key encryption achieves sub-exponential security.

Remark. In order to optimize the underlying hardness assumption, the proof (both for Theorem 6.2 and for Theorem 6.3) relies on $f_{G,P}$ being ε-unpredictable and not pseudorandom (for, say, $\varepsilon = 0.1$). Consequently, the resulting public-key encryption scheme is somewhat different than the one presented here (Construction 6.1). Also, this requires to assume that G and G' are hard to distinguish even when some entries of Q are published. It is shown that, for a proper choice of parameters, the two versions of Decisional Unbalanced Expansion are computationally equivalent.

6.2 General predicate

One may further ask whether it is possible to base the construction on general (non-linear) predicate P. While the security proof extends to this variant, decryption be comes problematic as we cannot apply Gaussian elimination. To solve this problem, it is suggested to let $q = O(\log n)$. In this case, the image of f' is of polynomial size and decryption can be implemented efficiently regardless of the structure of P. Security in this case, is based on the pseudorandomness of $\mathcal{F}_{P,n,m}$ and the DUE assumption for (m, n, d) with $q = O(\log n)$-size planted shrinking set. Note that a simple exhaustive search of the shrinking set requires $O(n^q) = n^{\log n}$ time, and so security cannot be better than quasi-polynomial. For super-linear $m = n^{1+\varepsilon}$, no better attack is known.[18]

Theorem 6.4 (Corollary 8.5 in [8]). *For every $q = \Theta(\log n)$ and function $m = m(n), d = d(n)$ and d-local predicate P, if both the $\mathsf{DUE}(m, n, d, q)$ assumption holds and $\mathcal{F}_{P,n,m}$ is pseudorandom, then there exists a public-key encryption scheme.*

[18] If $m = O(n)$ then the aforementioned (cycle-counting) attacks can be used to break security.

A natural open question is to remove the additional DUE assumption, and base public-key encryption on the one-wayness or pseudorandomness of $\mathcal{F}_{P,n,m}$ for a general predicate P. In essence, this would show that public-key cryptography follows from "super-fast" private-key cryptography.

Question 6.5. *Is there a public-key encryption scheme based on the one-wayness of $\mathcal{F}_{P,n,m}$ for a general family of predicates (other than noisy XOR)?*

For some setting of parameters, the DUE assumption follows from the pseudorandomness of $\mathcal{F}_{P,n,m}$ [7, Theorem 1.6], which in turn reduces to the one-wayness of $\mathcal{F}_{P,n,m'}$ (Theorem 4.4). Unfortunately, this result applies only to large values of q (roughly $q = n^{1+\delta/d}$ and $m = n^{1+\delta}$), and so it does not match the parameters needed for public-key encryption. One may hope to settle Question 6.5 affirmatively by either improving the hardness results of [7, Theorem 1.6] or by relaxing the requirements of the above public-key encryption scheme.

Applications. All three schemes can be used to derive an *oblivious transfer* protocol [58], which, in turn, allows to securely compute any multi-party functionality [43]. It is also shown in [8] that the third encryption scheme (with non-linear predicate) implies that the well known problem of PAC-learning $O(\log n)$-juntas [20, 21] is computationally intractable. This essentially follows by noting that our decryption algorithm looks at only $q = O(\log n) = O(\log m)$ of the bits of the m-bit ciphertext, and so it computes a q-junta. As observed in [49], the decryption function of any semantically-secure public-key encryption scheme is hard to learn. Specifically, consider the joint distribution over labeled-examples $(z, b) \in \{0,1\}^m \times \{0,1\}$ where the label is a random message $b \stackrel{R}{\leftarrow} \{0,1\}$ and the example z is a fresh ciphertext of b. Since an adversary can sample labeled examples from this distribution given only the knowledge of the public-key, the ability to learn the decryption function over this distribution allows to break the cryptosystem.[19]

Acknowledgement

This survey originates from two talks given at Oberwolfach Workshop for Computational Complexity (2012) and at The Tenth Theory of Cryptography Conference (TCC 2013). We thank the organizers of these events and to Oded Goldreich, the editor of this special issue, for initiating this survey. We are also deeply grateful to Oded for many valuable suggestions concerning the presentation of these results. The author is also grateful to Boaz Barak, Andrej Bogdanov, Uri Feige, Oded Goldreich, Yuval Ishai, Eyal Kushilevitz, Ryan O'Donnell, Alon Rosen, Dan Vilenchik, and Avi Wigderson for stimulating discussions.

References

[1] D. Achlioptas. *Handbook of Satisfiability*, chapter Random Satisfiability, pages 243–268. IOS Press, 2009.

[19] This argument assumes that the decryption algorithm errs with no more than negligible probability. In contrast, our basic template suffers from noticeable decryption errors, and standard error-reduction techniques (e.g., repetition) increase the locality of decryption. To cope with this problem, it is shown in [8] how to eliminate the decryption errors of Construction 6.1 (for most public-keys) without increasing the locality of the decryption algorithm.

[2] D. Achlioptas and F. Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. In *Proc. 38th STOC*, 2006.

[3] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293, 1997.

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

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

[6] N. Alon and N. Kahale. A spectral technique for coloring random 3-colorable graphs. In *SIAM J. Comput*, pages 346–355, 1994.

[7] B. Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. *SIAM J. Comput*, 42(5):2008–2037, 2013.

[8] B. Applebaum, B. Barak, and A. Wigderson. Public-key cryptography from different assumptions. In *Proc. of 42nd STOC*, pages 171–180, 2010.

[9] B. Applebaum, A. Bogdanov, and A. Rosen. A dichotomy for local small-bias generators. In *Proc. of 9th TCC*, pages 1–18, 2012.

[10] B. Applebaum, Y. Ishai, and E. Kushilevitz. On one-way functions with optimal locality. Unpublished manuscript available at `http://www.eng.tau.ac.il/~bennyap`, 2005.

[11] B. Applebaum, Y. Ishai, and E. Kushilevitz. Computationally private randomizing polynomials and their applications. *Journal of Computational Complexity*, 15(2):115–162, 2006.

[12] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM J. Comput*, 36(4):845–888, 2006. Preliminary version in FOCS 2004.

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

[14] B. Applebaum and Y. Moses. Locally computable UOWHF with linear shrinkage. In *Proc. EUROCRYPT '13*, pages 486–502, 2013.

[15] P. Austrin and E. Mossel. Approximation resistant predicates from pairwise independence. *Computational Complexity*, 18(2), 2009.

[16] B. Barak, S. O. Chan, and P. Kothari. Sum of Squares Lower Bounds from Pairwise Independence. In *Proc. of 47th STOC 2015*, 2015. available at `http://arxiv.org/abs/1501.00734`.

[17] B. Barak, G. Kindler, and D. Steurer. On the optimality of relaxations for average-case and generalized constraint satisfaction problems. In *Proc. 4th of ITCS 2012*, 2013.

[18] E. Ben-Sasson and A. Wigderson. Short proofs are narrow–resolution made simple. *J. ACM*, 48, 2001.

[19] S. Benabbas, K. Georgiou, A. Magen, and M. Tulsiani. SDP gaps from pairwise independence. *Theory of Computing*, 8(12):269–289, 2012.

[20] A. L. Blum. Relevant examples and relevant features: Thoughts from computational learning theory. AAAI Fall Symposium on Relevance, 1994.

[21] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[22] A. Bogdanov, P. Papakonstantinou, and A. Wan. Pseudorandomness for read-once formulas. In *Proc. 52nd FOCS*, 2011.

[23] A. Bogdanov and Y. Qiao. On the security of Goldreich's one-way function. *Computational Complexity*, 21(1):83–127, 2012.

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

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

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

[27] M. Charikar and A. Wirth. Maximizing quadratic programs: Extending grothendieck's inequality. In *Proc. 45th FOCS*, pages 54–60, 2004.

[28] A. Coja-Oghlan. Random constraint satisfaction problems. In *Proc. 5th DCM*, 2009.

[29] J. Cook, O. Etesami, R. Miller, and L. Trevisan. Goldreich's one-way function candidate and myopic backtracking algorithms. In *Proc. of 6th TCC*, pages 521–538, 2009. Full version in Electronic Colloquium on Computational Complexity (ECCC).

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

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

[32] I. Diakonikolas, D. M. Kane, and J. Nelson. Bounded independence fools degree-2 threshold functions. In *Proc. 51st FOCS*, pages 11–20, 2010.

[33] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(5):644–654, Nov. 1976.

[34] U. Feige. Relations between average case complexity and approximation complexity. In *Proc. of 34th STOC*, pages 534–543, 2002.

[35] U. Feige, J. H. Kim, and E. Ofek. Witnesses for non-satisfiability of dense random 3CNF formulas. In *Proc. 38th STOC*, pages 497–508, 2006.

[36] V. Feldman, W. Perkins, and S. Vempala. On the Complexity of Random Satisfiability Problems with Planted Solutions. *ArXiv e-prints*, 2013. Available at `http://arxiv.org/abs/1311.4821`.

[37] A. Flaxman. Random planted 3-SAT. In M.-Y. Kao, editor, *Encyclopedia of Algorithms*. Springer, 2008.

[38] A. Flaxman. A spectral technique for random satisfiable 3CNF formulas. *Random Struct. Algorithms*, 32(4):519–534, 2008.

[39] Goemans and Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42, 1995.

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

[41] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[42] O. Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.

[43] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game (extended abstract). In *Proc. of 19th STOC*, pages 218–229, 1987. See [42, Chapter 7].

[44] J. Håstad. Every 2-CSP allows nontrivial approximation. *Computational Complexity*, 17(4):549–566, 2008.

[45] R. Impagliazzo, N. Nisan, and A. Wigderson. Pseudorandomness for network algorithms. In *Proc. 26th STOC*, pages 356–364, 1994.

[46] Y. Ishai, E. Kushilevitz, X. Li, R. Ostrovsky, M. Prabhakaran, A. Sahai, and D. Zuckerman. Robust pseudorandom generators. In *ICALP (1)*, pages 576–588, 2013.

[47] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography with constant computational overhead. In *Proc. of 40th STOC*, pages 433–442, 2008.

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

[49] M. J. Kearns and L. G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, 1994.

[50] S. Khot. On the power of unique 2-prover 1-round games. In *Proc. 34th STOC*, 2002.

[51] S. Khot. Ruling out PTAS for graph min-bisection, densest subgraph and bipartite clique. In *Proc. of 45th FOCS*, pages 136–145, 2004.

[52] S. Lovett. Unconditional pseudorandom generators for low degree polynomials. *Theory of Computing*, 5(1):69–82, 2009.

[53] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *The Deep Space Network Progress Report, DSN PR 42-44, January and February 1978,*, 1978.

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

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

[56] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proc 21st STOC*, 1989.

[57] R. O'Donnell and D. Witmer. Goldreich's prg: Evidence for near-optimal polynomial stretch. In *Proc. 29th of CCC 2014*, 2014.

[58] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[59] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. of the ACM*, 21(2):120–126, 1978.

[60] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, IT-30(5):776–779, 1984.

[61] M. Tulsiani and P. Worah. LS_+ lower bounds from pairwise independence. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:105, 2012.

[62] E. Viola. The sum of d small-bias generators fools polynomials of degree d. In *IEEE Conference on Computational Complexity*, pages 124–127. IEEE Computer Society, 2008.