On the Concrete Security of Goldreich's Pseudorandom Generator

Geoffroy Couteau[1], Aurélien Dupin[2,3], Pierrick Méaux[4], Mélissa Rossi[2,5,6], and Yann Rotella[6]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
geoffroy.couteau@kit.edu
[2] Thales Communications and Security, Gennevilliers, France
[3] CentraleSupélec, Rennes, France and Irisa, Rennes, France
dupin.aurelien@gmail.com
[4] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium
pierrick.meaux@uclouvain.be
[5] École Normale Supérieure de Paris, Département d'informatique,
CNRS, PSL Research University, Paris, France
melissa.rossi@ens.fr
[6] Inria, Paris, France.
yann.rotella@inria.fr

Abstract. Local pseudorandom generators allow to expand a short random string into a long pseudo-random string, such that each output bit depends on a constant number d of input bits. Due to its extreme efficiency features, this intriguing primitive enjoys a wide variety of applications in cryptography and complexity. In the polynomial regime, where the seed is of size n and the output of size n^{s} for $\mathsf{s} > 1$, the only known solution, commonly known as *Goldreich's PRG*, proceeds by applying a simple d-ary predicate to public random size-d subsets of the bits of the seed.

While the security of Goldreich's PRG has been thoroughly investigated, with a variety of results deriving provable security guarantees against class of attacks in some parameter regimes and necessary criteria to be satisfied by the underlying predicate, little is known about its concrete security and efficiency. Motivated by its numerous theoretical applications and the hope of getting practical instantiations for some of them, we initiate a study of the concrete security of Goldreich's PRG, and evaluate its resistance to cryptanalytic attacks. Along the way, we develop a new guess-and-determine-style attack, and identify new criteria which refine existing criteria and capture the security guarantees of candidate local PRGs in a more fine-grained way.

Keywords: Pseudorandom generators, Algebraic attacks, Guess-and-Determine, Gröbner basis.

1 Introduction

One of the most fundamental problems in cryptography is the question of what makes an efficiently computable function hard to invert. The quest for the simplest design which leads to a primitive resisting all known attacks is at the heart of both symmetric and asymmetric cryptography: while we might be able to build seemingly secure primitives by relying on more and more complex designs to thwart cryptanalysis attempts, such a "security by obscurity" approach is unsatisfying. Instead, as advocated almost two decades ago by Goldreich [Gol00], we should seek to construct the simplest possible function that we do not know how to invert efficiently. Only this way, Goldreich argued, can we better understand what really underlies the security of cryptographic constructions.

Random Local Functions. In an attempt to tackle this fundamental problem, Goldreich suggested a very simple candidate one-way function as a promising target for cryptanalysis: let (n, m) be integers, and let $(\sigma^1, \dots, \sigma^m)$ be a list of m subsets of $[n]$, such that each subset is of small size: for any $i \leq m$, $|\sigma^i| = c(n)$, where $c(n) \ll n$ (in actual instantiations, $c(n)$ can for example be logarithmic in n, or even constant). Fix a simple predicate $P : \{0,1\}^{c(n)} \mapsto \{0,1\}$, and define the function $f : \{0,1\}^n \mapsto \{0,1\}^m$ as follows: on input $x \in \{0,1\}^n$, for any subset S of $[n]$, let $x[\sigma]$ denote the subset of the bits of x indexed by σ. Compute $f(x)$ as $P(x[\sigma^1])||\cdots||P(x[\sigma^m])$ (that is, $f(x)$ is computed by applying the predicate P to all subsets of the bits of x indexed by the sets $\sigma^1, \dots, \sigma^m$). We call *random local functions* the functions obtained by instantiating this template.

In his initial proposal, Goldreich advocated instantiating the above methodology with $m \approx n$ and $c(n) = O(\log n)$, and conjectured that if the subsets $(\sigma^1, \dots, \sigma^m)$ form an expander graph[1], and for an appropriate choice of the predicate P, it should be infeasible to invert the above function f in polynomial time. While setting $c(n)$ to $O(\log n)$ offers stronger security guarantees, the more extreme design choice $c(n) = O(1)$ (also discussed in Goldreich's paper) enhances the above candidate with an appealing feature: it enjoys constant input locality (which puts it into the complexity class NC^0), hence it is highly parallelizable (it can be computed in constant parallel time). It appeared in subsequent works that a stronger variant of Goldreich's conjecture, which considers $m \gg n$ and claims that f is in fact a *pseudorandom generator*, was of particular interest; we will elaborate on this later on.

Local Pseudorandom Generators. The question of whether cryptographic primitives can exist in weak complexity classes such as NC^0 has attracted a lot of attention in the cryptographic community. A primitive of particular interest, which has been the focus of most works on the subject, is the notion of pseudorandom generators (PRGs), which are functions $G : \{0,1\}^n \mapsto \{0,1\}^m$ extending a short random seed into a longer, pseudorandom string. The existence of PRGs in NC^0 was first considered by Cryan and Miltersen in [CM01]. Remarkably, it was shown by Applebaum, Ishai, and Kushilevitz [AIK04, AIK08] that cryptographically secure pseudorandom generators (with linear stretch $m = O(n)$)

[1] The subsets form an expander graph if for some k, every k subsets cover $k + \Omega(n)$ elements of $[n]$. In practice, it suffices to pick once for all the subsets $(\sigma^1, \dots, \sigma^m)$ at random to guarantee that they will be expanding except with $o(1)$ probability.

exist in a complexity class as low as NC^0_4 (the class of constant depth, polysize circuits where each output bit depends on at most 4 input bits), under widely believed standard assumption for the case of PRG with sublinear stretch (such as factorization, or discrete logarithm), and under a specific intractability assumption related to the hardness of decoding "sparsely generated" linear codes, for the case of PRG with linear stretch. While this essentially settled the question of the existence of linear stretch PRGs in NC^0, an intriguing open question remained: could PRGs in NC^0 have *polynomial* stretch, $m = \mathsf{poly}(n)$?

Some early negative results were given by Cryan and Miltersen [CM01] (who ruled out the existence of PRGs in NC^0_3 with stretch $m > 4n$) and Mossel, Shpilka, and Trevisan [MST03] (who ruled out the existence of PRGs in NC^0_4 with stretch $m > 24n$). The authors of [CM01] also conjectured that any candidate PRG with superlinear stretch in NC^0 would be broken by simple, linear distinguishing tests[1]; this conjecture was refuted in [MST03], who gave a concrete candidate PRG in NC^0, by instantiating a random local function with $c = 5$, and the predicate

$$P_5 : (x_1, x_2, x_3, x_4, x_5) \mapsto x_1 + x_2 + x_3 + x_4 x_5 \ .$$

where the + denotes the addition in $\mathbb{F}_2$ *i.e.* the xor.

They proved that this PRG fools linear tests, even when m is a (sufficiently small) polynomial in n. By the previously mentioned negative result on PRGs in NC^0_4, this candidate PRG, which has locality 5, achieves the best possible locality. Recently, there has been a renewed interest in the study of this local PRG, now commonly known as Goldreich's PRG, and its generalizations [BQ09, App12, OW14, CEMT14, App15, ABR16, AL16, IPS08, LV17, BCG+17].

1.1 Implications of Polynomial-Stretch Local Pseudorandom Generators

The original motivation for the study of local pseudorandom generators was the intriguing possibility of designing cryptographic primitives that can be evaluated in *constant time*, using polynomially many cores. While this is already a strong motivation in itself, it was observed in several works that the existence of (poly-stretch) local PRGs had a number of non-trivial implications, and is at the heart of feasibility results for several high-end cryptographic primitives. We provide below a brief overview.

- *Secure computation with constant computational overhead.* In the recent work [IKOS08], the authors explored the possibility of computing cryptographic primitives with essentially optimal efficiency, namely, constant overhead over a naive insecure implementation of the same task. One of their main results establishes the existence of constant-overhead two-party computation protocols for any boolean circuit, assuming the existence of poly-stretch local PRGs (and oblivious transfers). In a recent work [ADI+17a],

[1] A linear test attempts to distinguish a string from random by checking whether the xor of a subset of the bits of the string is biased toward either 0 or 1.

this result was extended to arithmetic circuits, using an arithmetic generalization of local PRGs.

- *Indistinguishability obfuscation (iO).* Introduced in the seminal paper of Barak et al. [BGI+01], iO is a primitive that has received a considerable attention from the crypto community in the past years, as a long sequence of works starting with [SW14] has demonstrated that iO had tremendous theoretical implications, to the point that it is often referred to as being a "crypto-complete" primitive. All known candidate constructions of iO rely, directly or indirectly, on a primitive called k-linear map, for some degree k. Recently, a sequence of papers (culminating with [LT17]) has attempted to find out the minimal k for which a k-linear map would imply the existence of iO (with the ultimate goal of reaching $k = 2$, as bilinear maps are well understood objects). These works have established a close relation between this value k and the existence of pseudorandom generators with poly-stretch, and locality k.[1]
- *MPC-friendly primitives.* Historically, the design of symmetric cryptographic primitives (such as block ciphers, pseudorandom generators, and pseudorandom functions) has been motivated by efficiency considerations (memory consumption, hardware compatibility, ease of implementation,...). The field of multiparty computation (MPC), where parties want to jointly evaluate a function on secret inputs, has led to the emergence of new efficiency considerations: the efficiency of secure evaluation of symmetric primitives is strongly related to parameters such as the circuit depth of the primitive, and the number of its AND gates. This observation has motivated the design of MPC-friendly symmetric primitives in several recent works (*e.g.* [ARS+15, CCF+16, MJSC16, GRR+16]). Local pseudorandom generators make very promising candidate MPC-friendly PRGs (and lead, through the GGM transform [GGM84], to promising candidates for MPC-friendly pseudorandom functions). Secure evaluation of such symmetric primitives enjoys a wide variety of applications.
- *Cryptographic capsules.* In [BCG+17], Boyle et al. studied the recently introduced primitive of homomorphic secret sharing (HSS). An important implication of HSS is that, assuming the existence of a local PRG with poly-stretch, one can obtain multiparty computation protocols in the preprocessing model[2] where the amount of communication between the parties is considerably smaller than the circuit size of the function, by constructing a primitive called cryptographic capsule which, informally, allows to compress correlated (pseudo-)random coins. MPC protocols with low-communication preprocessing have numerous appealing applications; however, the efficiency

[1] The locality requirement can in fact be weakened to a related notion of *block locality*.

[2] In this model, n parties securely compute a function f on private inputs $(x_1, \ldots, x_n)$; in the preprocessing phase, the parties have access to f (but not to the input), and generate some preprocessing material. Then, in the online phase, the parties execute an *information-theoretically secure* protocol to compute $f(x)$, using the preprocessed material. MPC protocols in the preprocessing model are among the most promising candidates for getting practical solutions to the multiparty computation problem.

of the constructions of cryptographic capsule strongly depends on the locality and seed size of the underlying local PRG (both should be as small as possible to get a reasonably efficient instantiation).

In addition to the above (non-exhaustive) overview, we note that the existence of poly-stretch local pseudorandom generators also enjoys interesting complexity-theoretic implications. For example, they have been shown in [AIK08] to imply strong (tight) bounds on the average-case inapproximability of constraints satisfactions problems such as Max3SAT.

1.2 On the Security of Goldreich's PRG

In this section, we provide a brief overview of the state-of-the-art regarding the security of local pseudorandom generators. For a more detailed and well-written overview dating from 2015, we refer the reader to [App15].

Positive Results: Security against Class of Attacks. The seminal paper of Goldreich [Gol00] made some preliminary observations on necessary properties for a local one-way function. Namely, the predicate P must satisfy some non-degeneracy properties, such as being non-linear (otherwise, one could inverse the function using Gaussian elimination). It also noted that to avoid a large class of natural "backtracking" attacks, which make a guess on the values of bit inputs based on local observations and attempt to combine many local solutions into a global solution, the subsets $(S_1, \ldots, S_m)$ should be sufficiently *expanding*: for some k, every k subsets should cover $k + \Omega(n)$ elements of $[n]$. The security of Goldreich's candidate one-way function against a large class of backtracking algorithm was formally analyzed in [AHI05, CEMT14], where it was proven that two restricted types of backtracking algorithms (called "drunk" and "myopic" backtracking algorithms) take exponential time to invert the function (with high probability). They also ran experiments to heuristically evaluate its security against SAT solvers (and observed experimentally an exponential increase in running time as a function of the input length).

The pseudorandomness of random local functions was originally analyzed in [MST03]. They proved (among other results) that the random local function instantiated with the predicate $P_5 : (x_1, x_2, x_3, x_4, x_5) \mapsto x_1 + x_2 + x_3 + x_4x_5$ fools all $\mathbb{F}_2$-linear distinguishers for a stretch up to $m(n) = n^{1.25-\varepsilon}$ (for an arbitrary small constant ε). This result was later extended to a larger stretch $n^{1.5-\varepsilon}$ in [OW14]. In the same paper, the authors proved that this candidate PRG is also secure against a powerful class of attacks, the Lasserre/Parrilo semidefinite programming (SDP) hierarchy, up to the same stretch. Regarding security against $\mathbb{F}_2$-linear attacks, a general dichotomy theorem was proven in [ABR12], which identified a class of *non-degenerate* predicates and showed that for most graphs, a local PRG instantiated with a non-degenerate predicate is secure against linear attacks, and for most graphs, a local PRG instantiated with a degenerate predicate is insecure against linear distinguishers. In general, to fool $\mathbb{F}_2$-linear distinguishers, the predicate should have high *algebraic degree* (in particular, a random local function instantiated with a degree-ℓ predicate cannot be pseudorandom for a stretch ℓ ($m \equiv n^\ell$), as it is broken by a straightforward Gaussian elimination attack).

Being pseudorandom seems to be a much stronger security property than being one-way. Nevertheless, in the case of random local functions, it was shown in [App12] that the existence of local pseudorandom generators follows from the existence of *one-way* random local functions (with sufficiently large output size).

Negative Results. The result of O'Donnell and Witmer [OW14] regarding security against SDP attacks is almost optimal, as attacks from this class are known to break the candidate for a stretch $\Theta(n^{1.5} \log n)$. More generally, optimizing SDP attacks leads to a polytime inversion algorithm for any predicate P which is (even slightly) correlated with some number c of its inputs, as soon as the output size exceeds $m \in \Omega(n^{c/2} + n \log n)$ [OW14, App15]. Therefore, a good predicate should have high *resiliency* (*i.e.* it should be k-wise independent, for a k as large as possible). This result shows, in particular, that a random local function with a constant locality d and with an output size $m > \mathsf{poly}(d) \cdot n$ is insecure when instantiated with a uniformly random predicate P. Combining this observation with the result of Siegenthaler [Sie84], which studied the correlation of d-ary predicates, gives a polytime inversion algorithm for any random local function implemented with a d-ary predicate, and with an output size $m \in \Omega(n^{1/2\lfloor 2d/3 \rfloor} \log n)$.

Bogdanov and Qiao [BQ09] studied the security of random local functions when the output is sufficiently larger than the input (*i.e.*, $m \geq Dn$, for a large constant D). They proved that for sufficiently large D, inverting a random local function could be reduced to finding an *approximate inverse* (*i.e.* finding any x' which is close to the inverse x in Hamming distance), by showing how to invert the function with high probability given an advice x' close to x. For random local function with an output size polynomial in n, $m = n^{\mathsf{s}}$ for some s, this leads to a subexponential-time attack [App15]: fix a parameter ε, assign random values to the $(1 - 2\varepsilon)n$ first inputs, and create a list that enumerates over all possible $2\varepsilon n$ assignments for the remaining variables. Then the list is guaranteed to contain a value x' that agree with the preimage x on a $(1/2+\varepsilon)n$ fraction of the coordinates with good probability. By applying the reduction of [BQ09], using each element of the list as an advice string, one recovers the preimage in time $\mathsf{poly}(n) \cdot 2^{2\varepsilon n}$ provided that $m = \Omega(n/\varepsilon^{2d})$ (d is the arity of the predicate P). In the case of the 5-ary predicate P_5, this leads to an attack in subexponential-time $2^{O(n^{1-(\mathsf{s}-1)/2d})}$ (*e.g.* using $\mathsf{s} = 1.45$ gives an attack in time $2^{O(n^{0.955})}$).

By the previous observations, we know that the predicate of a random local function must have high resiliency and high algebraic degree to lead to a pseudorandom function. A natural question is whether this characterization is also sufficient; this question was answered negatively in [AL16], who proved that a predicate must also have high *bit-fixing degree* to fool linear attacks.[1] In particular, this observation disproved a previous conjecture of Applebaum that XOR-AND predicates (which are natural generalizations of the predicate P_5) could lead to local PRGs with stretch greater than 2 that fools all linear tests (see [AL16, Corollary 1.3]).

In the same work, Applebaum and Lovett considered the class of algebraic attacks on local pseudorandom function, which are incomparable to linear attacks. An algebraic attack against a function $f : \{0,1\}^n \mapsto \{0,1\}^m$ starts with an output y and uses it to initialize a system of polynomial equations over the input variables $x = (x_1, \dots, x_n)$. The system is further manipulated and extended until a solution is found or until the system is refuted. Applebaum and Lovett proved that a predicate must also have high *rational degree* to fool algebraic attacks (a predicate P has rational degree e if it is the smallest integer for which there exist degree e polynomials Q and R, not both zero, such that $PQ = R$). Indeed, if $e < \mathsf{s}$ then P is not s-pseudorandom against algebraic attacks (see [AL16], Theorem 1.4). In the symmetric cryptography community, the rational degree denotes the well-known *algebraic immunity* criterion on Boolean function that underlies the so-called *algebraic attacks* on stream ciphers [CM03, Cou03]. An algebraic immunity of e implies an r-bit fixing degree greater than or equal to $e - r$ ([DGM05], Proposition 1), giving that an high algebraic immunity guarantees both high rational degree and high bit fixing degree. The algebraic degree is equivalent to the 0-bit fixing degree, then it leads to the following characterization: a predicate of a random local function must have high resiliency and high algebraic immunity. In light of this characterization, the authors of [AL16] suggested the XOR-MAJ predicate as a promising candidate for building high-stretch local PRGs, the majority function having optimal algebraic immunity [DMS05].

Security against Subexponential Attacks. While there is a large body of work that studied the security of random local functions, leading to a detailed characterization of the parameters and predicates that lead to insecure instantiations, relatively little is known on the *exact* security of local PRGs instantiated with non-degenerated parameters. In particular, most papers only prove that some classes of polytime attacks provably fail to break candidates local PRGs; however, these results do not preclude the possible existence of non-trivial subexponential attacks (specifically, these polytime attacks do not "degrade gracefully" into subexponential attacks when appropriate parameters are chosen for the PRG; instead, they do always and provably not succeed). To our knowledge, the only results in this regard are the proof from [AHI05, CEMT14] that many backtracking-type attacks require exponential time to invert a random local function, and the subexponential-time attack arising from the work of Bogdanov and Qiao [BQ09]. However, as we saw above, the latter attack only gives a slightly-

[1] A predicate P has r-bit fixing degree e if the minimal degree of the restriction of P obtained by fixing r inputs is e

subexponential algorithm, in time $2^{O(n^{1-(\mathsf{s}-1)/2d})}$ for a d-ary predicate, and an n^{s}-stretch local PRG.

1.3 Our Goals and Results

In this work, we continue the study of the most common candidate local pseudorandom generators. However, we significantly depart from the approach of previous works, in that we wish to analyze the *concrete* security of local PRGs. To our knowledge, all previous works were only concerned about establishing asymptotic security guarantees for candidate local PRGs, without providing any insight on, *e.g.*, which parameters can be conjectured to lead to a primitive with a given bit-security. Our motivations for conducting this study are twofold.

- Several recent results, which we briefly overviewed in Section 1.1, indicate that (poly-stretch) local PRGs enjoy important theoretical applications. However, the possibility of instantiating these applications with concrete PRG candidates remains unclear, as their efficiency quickly deteriorates with the parameters of the underlying PRG. For example, the iO scheme of [LT17], which requires low-degree multilinear maps and therefore might be a viable approach to obtain efficiency improvements in iO constructions (as candidate high-degree multilinear maps are prohibitively expensive); however, it has a cost cubic in the seed size of a poly-stretch local PRG, which renders it practical only if we can safely use local PRGs with reasonably small seeds. Overall, we believe that there is a growing need for a better understanding of the exact efficiency of candidate local PRGs, and providing concrete estimations can prove helpful for researchers willing to understand which efficiency could potentially be obtained for local-PRG-based primitives.
- At a more theoretical level, previous works on (variants of) Goldreich's PRG have identified criteria which characterize the predicates susceptible to lead to secure local PRGs. Identifying such criteria is particularly relevant to the initial goal set up by Goldreich in [Gol00], which is to understand what characteristics of a function is the source of its cryptographic hardness, by designing the simplest possible candidate that resists all attacks we know of. However, existing criteria only distinguish predicates leading to insecure instances from those leading to instances for which no polynomial-time attack is known. We believe that it is also of particular relevance to this fundamental question to find criteria which capture in a more fine-grained way the cryptographic hardness of random local functions.

Our Results. We provide new cryptanalytic insights on the security of Goldreich's pseudorandom generator.

- *A new subexponential attack on Goldreich's PRG.* We start by devising a new attack on Goldreich's PRG. Our attack relies on a *guess-and-determine* technique, in the spirit of the recent attack [DLR16] on the FLIP family of stream ciphers [MJSC16]. The complexity of our attack is $2^{O(n^{2-\mathsf{s}})}$ where s is the stretch and n is the seed size. This complements O'Donnel and Witmer's result [OW14] showing that Goldreich's PRG is likely to be secure for stretch up to 1.5, with a more fine-grained complexity estimation. We implemented our attack and provide experimental results regarding its concrete efficiency, for various seed size and stretch parameters.

- *Generalization.* We generalize the previous attack to a large class of predicates, which are divided into two parts, a linear part and a non-linear part, XORed together. This captures all known candidate generalizations of Goldreich's PRG. Our attack takes subexponential time as soon as the stretch of the PRG is strictly above one. Importantly, our attack does not depend on the locality of the predicate, but only on the number of variables involved in the non-linear part. In a recent work [AL16], Applebaum and Lovett put forth an explicit candidate local PRG (of the form XOR-MAJ), as a concrete target for cryptanalytic effort. Our attack gives a new subexponential algorithm for attacking this candidate.
- *Extending the Applebaum-Lovett polynomial-time algebraic attack.* Applebaum and Lovett recently established that local pseudorandom generators can be broken in polynomial time, as long as the stretch s of the PRG is greater than the *rational degree* e of its predicate. We extend this result as follows: we show that the seed of a large class of local PRGs (which include all existing candidates) can be recovered in polynomial time whenever $\mathsf{s} \geq e - \log N_e / \log n$, where e is the rational degree, n is the seed size, and N_e is the number of independent annihilators of the predicate[1] of degree at most e.
- *Linearization and Gröbner attack.* We complement our study with an analysis of the efficiency of algebraic attacks *à la* Gröbner on Goldreich's PRG. While it is known that Goldreich's PRG (and its variants) provably resists such attacks for appropriate choices of (asymptotic) parameters [AL16], little is known about its exact security against such attacks for concrete choices of parameters. We evaluated the concrete security of Goldreich's PRG against an order-two linearization attack. The existence of such an attack allows to derive bounds on Gröbner basis performance. Using an implemented proof of concept, we introduce heuristic bounds for vulnerable parameters.

As illustrated by our attacks, both the number of annihilators of the predicate and the r bit fixing algebraic immunity play an important role in the security of Golreich's PRG. These criteria were overlooked in all previous works on local PRGs. Last but not least, our concrete analysis indicates that Gröbner basis attacks, although provably "ruled out" asymptotically, matters when studying the vulnerabilities of Goldreich's PRG, and the security of concrete instances.

1.4 Organization of the Paper

Section 2 introduces necessary preliminaries on predicates and local pseudorandom generators. Section 3 describes a guess-and-determine attack on Goldreich's

[1] An annihilator of a predicate P is a non-zero polynomials Q such that $Q \cdot P = 0$

PRG instantiated with the predicate P_5 and analyzes it, where the proofs are given in the full version of our paper [CDM+18]. Section 4 extends this attack to all predicates of the form XOR-MAJ, where the proofs are given in the full version of our paper. Eventually, still in the full version of our paper, an order 2 linearization attack on Goldreich's PRG is described. The same full version of our paper considers the case of using Goldreich's PRG with ordered subset (as was initially advocated in [Gol00]) and provides indications that this weakens its concrete security. Finally, the full version of our paper improves the theorem of Applebaum and Lovett, by taking into account the number of annihilators of the predicate. The full version of our paper contains missing proofs on collisions.

2 Preliminaries

Throughout this paper, n denotes the size of the seed of the PRGs considered. A probabilistic polynomial time algorithm (PPT, also denoted *efficient* algorithm) runs in time polynomial in the parameter n. A positive function f is *negligible* if for any polynomial p there exists a bound $B > 0$ such that, for any integer $k \geq B$, $f(k) \leq 1/|p(k)|$. An event depending on n occurs with *overwhelming probability* when its probability is at least $1 - \mathsf{negl}(n)$ for a negligible function negl. Given an integer k, we write $[k]$ to denote the set $\{1, \ldots, k\}$. Given a finite set S, the notation $X \xleftarrow{\$} S$ means a uniformly random assignment of an element of S to the variable X. Given a string $x \in \{0,1\}^k$ for some k and a subset σ of $[k]$, we let $x[\sigma]$ denote the subsequence of the bits of x whose index belong to σ. Moreover, the i-th bit of $x[\sigma]$ will be denoted by x_{σ_i}.

2.1 Hypergraphs

Hypergraphs generalize the standard notion of graphs (which are defined by a set of nodes and a set of edges, an edge being a pair of nodes) to a more general object defined by a set of nodes and a set of *hyperedges*, each hyperedge being an arbitrary subset of the nodes. We define an (n, m, d)-hypergraph G to be a hypergraph with n vertices and m hyperedges, each hyperedge having cardinality d. The hyperedges are assumed to be ordered from 1 to m, and each hyperedge $\{i_1, i_2, \ldots, i_d\}$ is ordered and satisfies $i_j \neq i_k$ for all $j \leq d$, $k \leq d$, $j \neq k$. We will consider hypergraphs satisfying some expansion property, defined below.

Definition 1 (Expander Graph).
*An (n, m, d)-hypergraph G, denoted $(\sigma^1, \ldots, \sigma^m)$, is (α, β)-*expanding *if for any $S \subset [m]$ such that $|S| \leq \alpha \cdot m$, it holds that $|\cup_{i \in S} \sigma^i| \geq \beta \cdot |S| \cdot d$.*

2.2 Predicates

The constructions of local pseudorandom generators that we will consider in this work rely on predicates satisfying some specific properties. Formally, a predicate P of arity d is a function $P : \{0,1\}^d \mapsto \{0,1\}$. We define below the two properties that were shown to be necessary for instantiating local PRGs:

- *Resiliency.* A predicate P is k-resilient if it has no nontrivial correlation with any linear combination of up to k of its inputs. An example of predicate with maximal resiliency is the parity predicate (*i.e.*, the predicate which xors all its inputs).
- *Algebraic Immunity.* A predicate P has algebraic immunity e, referred to as $\mathsf{AI}(P) = e$, if the minimal degree of a non null function g such that $Pg = 0$ (or $(P+1)g = 0$) on all its entries is e. A local PRG built from a AI-e predicate cannot be pseudorandom with a stretch n^e due to algebraic attacks.

Note that the algebraic immunity (also referred as rational degree in [AL16]) implies a lower bound on the degree and on the bit-fixing degree. Moreover, a high algebraic immunity implies at least the same degree. Hence, for now on, those two criterion are considered as the relevant criteria for evaluating the security of Goldreich's PRG.

We define a particular family of predicates which have been considered as a potential instantiation:

Definition 2 ($\mathsf{XOR}_\ell\mathsf{M}_k$ predicates). *We call $\mathsf{XOR}_\ell\mathsf{M}_k$ predicate a predicate P of arity $\ell + k$ such that M is a predicate of arity k and:*

$$P(x_1, \ldots, x_\ell, z_1, \ldots, z_k) = \sum_{i=1}^{\ell} x_i + M(z_1, \ldots, z_k) \; .$$

We define also a subfamily of $\mathsf{XOR}_\ell\mathsf{M}_k$ predicates, which have been considered in [AL16]:

Definition 3 ($\mathsf{XOR}_\ell\mathsf{MAJ}_k$ predicates). *We call $\mathsf{XOR}_\ell\mathsf{MAJ}_k$ predicate a predicate P of arity $\ell + k$ such that P is a $\mathsf{XOR}_\ell\mathsf{M}_k$ predicate such that M is the majority function in k variables:*

$$M(z_1, \ldots, z_k) = 1 \Leftrightarrow \mathsf{w}_H(z_1, \ldots, z_k) \geq \left\lceil \frac{k}{2} \right\rceil \; ,$$

where w_H denotes the Hamming weight.

2.3 Pseudorandom Generators

Definition. A pseudorandom generator is a deterministic process that expands a short random seed into a longer sequence, so that no efficient adversary can distinguish this sequence from a uniformly random string of the same length. Formally,

Definition 4 (Pseudorandom Generator). *A $m(n)$-stretch pseudorandom generator, for a polynomial m, is an efficient uniform deterministic algorithm PRG which, on input a seed $x \in \{0,1\}^n$, outputs a string $y \in \{0,1\}^{m(n)}$. It satisfies the following security notion: for any probabilistic polynomial-time adversary Adv,*

$$\begin{aligned} &\Pr[y \xleftarrow{\$} \{0,1\}^{m(n)} : \mathsf{Adv}(\mathsf{pp}, y) = 1] \\ \approx &\Pr[x \xleftarrow{\$} \{0,1\}^n, y \leftarrow \mathsf{PRG}(x) : \mathsf{Adv}(\mathsf{pp}, y) = 1] \end{aligned}$$

Here $\approx$ denotes that the absolute value of the difference of the two probabilities is negligible in the security parameters, and pp *stands for the public parameters of the* PRG*. For any $n \in \mathbb{N}$, we denote* PRG_n *the function* PRG *restricted to n-bit inputs. A pseudorandom generator* PRG *is d-*local *(for a constant d) if for any $n \in \mathbb{N}$, every output bit of* PRG_n *depends on at most d input bits.*

Goldreich's Pseudorandom Generator. Goldreich's candidate local PRGs form a family $\mathsf{F}_{G,P}$ of local PRGs: $\mathsf{PRG}_{G,P} : \{0,1\}^n \mapsto \{0,1\}^m$, parametrized by an (n, m, d)-hypergraph $G = (\sigma^1, \ldots, \sigma^m)$ (where $m = m(n)$ is polynomial in n), and a predicate $P : \{0,1\}^d \mapsto \{0,1\}$, defined as follows: on input $x \in \{0,1\}^n$, $\mathsf{PRG}_{G,P}$ returns the m-bit string $(P(x_{\sigma^1_1}, \ldots, x_{\sigma^1_d}), \ldots, P(x_{\sigma^m_1}, \ldots, x_{\sigma^m_d}))$.

Conjecture 1 (Informal). If G is a sufficiently expanding (n, m, d) hypergraph and P is a predicate with sufficiently high resiliency and high algebraic immunity, then the function $\mathsf{PRG}_{G,P}$ is a secure pseudorandom generator.

Note that picking an hypergraph G uniformly at random suffices to ensure that it will be expanding with probability $1 - o(1)$. However, picking a random graph will always give a non-negligible probability of having an insecure PRG. To see that, observe that when the locality d is constant, a random hypergraph G will have two hyperedges containing the same vertices with probability $1/\mathsf{poly}(n)$; for any such graph G, the output of $\mathsf{PRG}_{G,P}$ on a random input can be trivially distinguished from random. Therefore, the security of random local functions is usually formulated non-uniformly, by stating that for a $1 - o(1)$ fraction of all hypergraphs G (and appropriate choice of P), no polytime adversary should be able to distinguish the output of $\mathsf{PRG}_{G,P}$ from random with non-negligible probability.

Fixed hypergraph versus random hypergraphs. Goldreich's candidates local pseudorandom generators require to use a sufficiently expanding hypergraph. Unfortunately, building concrete graphs satisfying the appropriate expansion properties is a non-trivial task. Indeed, all known concrete constructions of expanding bipartite hypergraphs fail to achieve parameters which would allow to construct a PRG with constant locality. Therefore, to our knowledge, in all works using local PRG (see e.g. [IKOS08, App13, Lin17, ADI+17b, BCG+17]), it is always assumed (implicitly or explicitly) that the hypergraph G of the PRG is picked uniformly at random (which makes it sufficiently expanding with probability $1 - o(1)$, even in the constant-locality setting) in a one-time setup phase. Therefore, this is the setting we assume for our cryptanalysis.

Notations. In the first part of this work, we focus on the predicate P_5, assuming that the subsets $\sigma^1, ..., \sigma^m$ are random subsets. The predicate P_5 can be regarded as a Boolean function of five variables:

$$P_5(x_1, x_2, x_3, x_4, x_5) = x_1 + x_2 + x_3 + x_4 x_5 \ .$$

The predicate P_5 has algebraic degree 2 and an algebraic immunity of 2, and is 2-resilient. Let n be the size of the input, *i.e.* the number of initial random bits. We define the stretch s and denote the size m of the output as $m = n^{\mathsf{s}}$. Let $x_1, \ldots, x_n \in \mathbb{F}_2$ be the input random bits and $y_1, \ldots, y_m \in \mathbb{F}_2$ be the output bits. The m public equations E_i for $1 \leq i \leq m$ are drawn as follows:

– a subsequence of $[n]$ of size 5 is chosen uniformly at random. Let us call it

$$\sigma^i = [\sigma^i_1, \sigma^i_2, \sigma^i_3, \sigma^i_4, \sigma^i_5] \ .$$

– E_i is the quadratic equation of the form

$$x_{\sigma^i_1} + x_{\sigma^i_2} + x_{\sigma^i_3} + x_{\sigma^i_4} x_{\sigma^i_5} = y_i \ .$$

The public system Σ that we consider is then defined with the m equations, that is $(E_i)_{1 \leq i \leq m}$.

Ordered and unordered. There are two different cases to consider:

1. (Ordered case) σ^i is ordered, *i.e.* $\sigma^i_1 < \sigma^i_2 < \sigma^i_3 < \sigma^i_4 < \sigma^i_5$.
2. (Unordered case) The order σ^i's elements is arbitrary.

However, in the core of the paper, we will consider the **unordered case**, as we'll provide evidence that the vulnerabilities are even more important for the ordered case in the full version of our paper [CDM+18].

Matrix inversion complexity. Our attacks require a sparse matrix inversion algorithm. We consider the Wiedemann's algorithm [Wie86], the complexity of which is $O(n^2)$ in our context, since there are less than $d \cdot n$ non-zero elements of our matrices. Other algorithms could be used, but the complexity of our attacks would have to be modified accordingly.

3 Guess and Determine Cryptanalysis of Goldreich's PRG with P_5

In this section, we describe a new subexponential seed recovery attack on Goldreich's PRG when instantiated within the predicate P_5. Our attack is a *Guess and Determine* like attack, which is a widely used technique in symmetric cryptanalysis [HR00, EJ00]. As an example, a similar attack [DLR16] has been done on the preliminary version of the stream cipher FLIP [MJSC16] (which can be interpreted as an instance of Goldreich's PRG with linear locality and fixed security parameters). The idea of guessing elements before making algebraic analysis has been also introduced in [Bet11] under the name of *hybrid attacks*. In the following, we sketch a similar idea applied to the highly structured Goldreich's PRG .

3.1 Overview of the Attack

Using the above notations, we further make the following observations on Goldreich's PRG instantiated with P_5.

Observations.

Quasi-linearity If either $x_{\sigma_4^i}$ or $x_{\sigma_5^i}$ is known, then the corresponding equation becomes a linear equation. This is the main vulnerability that we use to mount our attack.

Collisions If two equations have the same monomial of degree 2, then the sum of these equations becomes linear (details are given in Section 3.2). Using this phenomenon, we can also get linear equations. We first analyze the number c of pairs of equations that shares a monomial of degree 2. Let the notion of collision refer to this phenomenon.

Definition 5 (Collision). *A collision is a couple $(i, j) \in [m]^2$ such that $i \neq j$ and $\{\sigma_4^i, \sigma_5^i\} = \{\sigma_4^j, \sigma_5^j\}$.*

Combining both observations, a subexponential attack can be derived. The main idea is to find linear equations using collisions and quasi-linearity.

The attack.

step 1 Find all collisions and derive the corresponding linear equations. Let c be the number of linear equations obtained with this step.

step 2 Take a small subset of ℓ variables in $\{x_1, \dots, x_n\}$, called $x_{i_1}, \dots, x_{i_\ell}$, such that by guessing them, $n - c$ new equations are generated (ℓ is formally defined in Definition 6).

step 3 For all 2^ℓ possible values of $(x_{i_1}, \dots, x_{i_\ell})$, build the system of at least n linear equations, solve it[1], find a candidate seed and check if that candidate matches the public evaluation of the PRG. If so, then it is the secret seed and the guess is correct.

Definition 6 (Number of guesses ℓ). *Let an instance of Goldreich's PRG be generated with n variables and m equations. Let c be the number of collisions. Let us define ℓ as a sufficient number of guesses required to build $n - c$ linear equations.*

The above attack works as long as the systems of linear equations obtained in step 2 and 3 above contain an invertible subsystem of size sufficiently large to recover the seed. Our experiments confirm that this is always the case. We formalize this observation with a combinatorial hypothesis: define $\mathcal{D}_n$ to be the distribution over $\mathbb{F}_2^{n\times n}$ obtained by sampling the hypergraph of Goldreich's PRG

[1] If more than n linear equations are recovered from Step 1 and 2, the system is unlikely to be solvable for an incorrect guess. In that case, it is not necessary to check if the public output matches with the candidate seed.

at random (with $d = 5$), finding c linear equations from the collisions, taking the smallest subset of variables which suffices to recover $n' \geq n - c$ additional linear equations, guessing at random the value of these variables, and outputting the $n \times n$ matrix A_n of the linear system (if $n' > n$, we truncate to n equations for simplicity).

Hypothesis 1 *There exists a constant γ such that for every sufficiently large $n \in \mathbb{N}$, the matrix A_n contains with overwhelming probability an invertible subsystem of $\gamma \cdot n$ equations, where the probability is taken over the coins of $A_n \overset{\$}{\leftarrow} \mathcal{D}_n$.*

In the full version of this work [CDM+18], we provide a detailed analysis of Hypothesis 1. Specifically:

- By applying the result of [BQ09], which describes a polytime seed recovery attack given an approximate preimage of the PRG, we formally show that Hypothesis 1 implies that our attack succeeds with overwhelming probability.
- We conduct detailed experimentations. In our experiments, the matrix A_n always contains an invertible subsystem of $\gamma \cdot n$ equations, with $\gamma > 0.9$.
- We show that Hypothesis 1 is related to well-established conjectures in mathematics, related to the distribution of the rank of random sparse matrices. Unfortunately, formally proving Hypothesis 1, even under some heuristics (e.g. replacing $\mathcal{D}_n$ by the uniform distribution over sparse matrices), appears to be a highly non-trivial mathematical problem, which requires techniques far out of the scope of the current paper.
- Eventually, we show that our attack can be modified to (provably) break the *pseudorandomness* of Goldreich's PRG, without having to rely on any unproved hypothesis. Hence, Hypothesis 1 seems to be only necessary for showing that our attack breaks the *one-wayness* of Goldreich's PRG.

In the next part, we give more details of our attack and we prove that the complexity of this attack will always be smaller than

$$O(n^2 2^{n^{2-\mathrm{s}}}) .$$

We later introduce experimental results in Section 3.3.

3.2 Complexity analysis and details

Assessing the number of collisions. As previously noticed, collisions can be used to build linear equations. For example, let us assume we have the following two equations in Σ:

$$x_{\sigma^i_1} + x_{\sigma^i_2} + x_{\sigma^i_3} + x_{\sigma^i_4} x_{\sigma^i_5} = y_i \tag{1}$$

$$x_{\sigma^j_1} + x_{\sigma^j_2} + x_{\sigma^j_3} + x_{\sigma^i_4} x_{\sigma^i_5} = y_j \tag{2}$$

then adding equation (1) and equation (2) gives us the following linear equation:

$$x_{\sigma^i_1} + x_{\sigma^i_2} + x_{\sigma^i_3} + x_{\sigma^j_1} + x_{\sigma^j_2} + x_{\sigma^j_3} = y_i + y_j$$

However, we stress that if we had a third colliding equation:

$$x_{\sigma_1^k} + x_{\sigma_2^k} + x_{\sigma_3^k} + x_{\sigma_4^i} x_{\sigma_5^i} = y_k \qquad (3)$$

then we could only produce a single other linear equation (w.l.o.g. (1) + (3)), since the other combination ((2) + (3)) would be linearly equivalent to the two previous linear equations.

Hence, this problem can be seen as a balls-into-bins problem: m balls are randomly thrown into $\binom{n}{2}$ bins and we want to know how many balls in average hit a bin that already contains at least one ball. Indeed, this number will approximate the value c of the algorithm.

Proposition 1 (Average number of collisions). *Let n be the number of variables, and m be the number of equations, let C be the random variable counting the number of collisions on the degree two monomials in the whole system. Then, the average number of collisions is:*

$$\mathbb{E}(C) = m - \binom{n}{2} + \binom{n}{2} \left(\frac{\binom{n}{2} - 1}{\binom{n}{2}} \right)^m \in O(n^{2(\mathsf{s}-1)}) \ .$$

The proof of this proposition is given in the full version [CDM+18]. Tab. 1 gives the evaluation of this formula for some set of parameters. Our experimental results (see Section 3.3) corroborate these expectations and show that the number of collisions is always very close to this expected average.

We now assess the complexity of the first step.

Lemma 1. *In the worst case, Step 1 has complexity $O(m \cdot \log(m))$.*

The proof is given in the full version of our paper [CDM+18].

Finding the smallest subset of guesses. The dominant term of the complexity of our attack is given by the number of guesses ℓ we have to make in the second step. Thus, minimizing ℓ is important. Consequently, the variables of the seed that we guess correspond to those appearing the most in the monomials of degree two. Then, the worst case happens when the instance of the PRG is such that there is no best set of guesses. In this specific unlikely setting, each guess generates the exact same amount of linear equations. Here, we bound the number of guesses with the minimum number of guesses for a worst case system.

Table 1. Average number of collisions

n	256	512	1024	2048	4096
$\mathsf{s} = 1.45$	142	269	506	946	1771
$\mathsf{s} = 1.4$	83	145	254	442	773
$\mathsf{s} = 1.3$	28	42	64	97	147

Proposition 2 (Number of guesses). *For any instance with n variables, m equations and c collisions, an upper bound on the sufficient number of guesses required to build $n - c$ linear equations is:*

$$\ell \leq \left\lfloor \frac{n(n-c)}{2(m-c)+n} + 1 \right\rfloor . \tag{4}$$

The proof is given in the full version of our paper [CDM+18]. Eventually, Equation 4 can be approximated with

$$\left\lfloor \frac{n(n-c)}{2(m-c)+n} + 1 \right\rfloor \simeq O\left(\frac{n^{2-\mathsf{s}}}{2}\right) . \tag{5}$$

We show further in Section 3.3 that experimental results are much better. We stress that this theoretical worst case expectation is far from experience. Some explanations of this gap are given in the full version of our paper.

The complexity of Step 2 is given by the following lemma.

Lemma 2. *Step 2 has complexity $O(\ell \cdot m)$ which is $O(n^2)$ with Equation 5 estimation.*

The proof is given in the full version of our paper.

Solving the linear system. Now, ℓ variables $\{x_{i_1}, \dots, x_{i_\ell}\}$ are chosen to be guessed and an exhaustion over all the 2^ℓ values of these variables is necessary. For every possible guess, one can try to solve the linear equations collected in the previous steps. In the case that more than n equations are collected, the system is overdetermined and thus may not be solvable. If so, then the guess is incorrect, else we obtain a candidate seed. This candidate can be either confirmed or rejected using the public quadratic system and the public output of the PRG. If the candidate is rejected, then the guess is also incorrect. However, if the candidate matches the public evaluation of the PRG, then the candidate seed is the secret seed with overwhelming probability[1] and the search can be stopped.

The complexity of this attack is given by the following lemma.

Lemma 3. *The complexity of Step 3 is*

$$O\left(n^\omega 2^{\frac{n^{2-s}}{2}}\right) ,$$

which is also the asymptotic complexity of the full attack.

The proof is given in the full version of our paper [CDM+18].

[1] It is very unlikely that two seeds give the same output by evaluating the same quadratic system. Even though, if it is the case, this procedure still finds an equivalent seed which makes the system insecure.

3.3 Experiment

Distribution of the number of collisions. The theoretical results of Table 1 are verified in practice, as shown in Fig. 1 for the particular case of $n = 1024$ and $s = 1.4$. As expected with the analytical formula, the number of collisions is very close to 254 in average. Moreover, our experimental results are very dense around the average, suggesting that the distribution has a low variance.

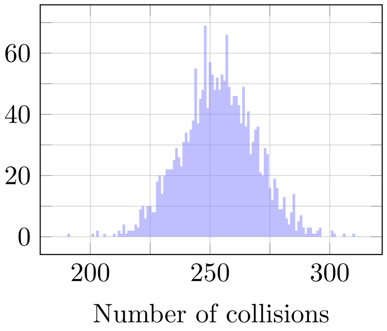

Fig. 1. Number of collisions for $n = 1024$ and $s = 1.4$ with 2000 tests

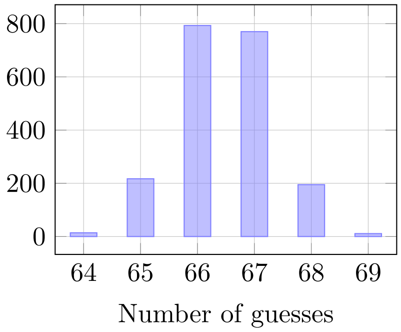

Fig. 2. Number of guesses for $n = 2048$ and $s = 1.3$ with 2000 tests

Implementation of the attack. Since the study of this paper is the concrete security of Goldreich's PRG , it is important to practically check if the attack presented in Section 3.1 can be efficient when implemented. For this purpose, we provide a proof of concept in Python.

One can note that the practical attack should be on average more efficient than assessed theoretically. Indeed, the asymptotic complexity of Proposition 3 is estimated in the worst case and pessimistic approximations were made on $n-c$ and on the value of ℓ. Hence, we experimented this attack for different stretches and different values of n and we effectively noticed that the complexity in average is much smaller than the expected complexity. Table 2 represents the theoretical number of guesses necessary to recover the seed and Table 3 represents the average number of guesses actually needed in the experiment. Moreover, we also noticed that the number of guesses needed to invert the system has a very low variance, as shown in Fig. 2.

With this experiment, we were able to estimate the practical security of Goldreich's PRG against the guess and determine approach with 80 bits of security. Indeed, for one instance of the PRG, the complexity of the seed recovery can be easily derived from the number ℓ of guesses as $2^{\ell} n^{\omega}$. So to assess the 80 bits security, one can evaluate the average number of guesses necessary for one choice of (n, s) and check if the complexity is lower than 2^{80}. For that, for 30 values of $n \in [2^7, 2^{14}]$, we delimited the smallest stretch for which the average

Table 2. Theoretical number of guesses in the worst case

n	256	512	1024	2048	4096
$\mathsf{s} = 1.45$	4	7	11	18	27
$\mathsf{s} = 1.4$	9	15	23	37	58
$\mathsf{s} = 1.3$	20	34	56	94	156

Table 3. Experimental number of guesses in average

n	256	512	1024	2048	4096
$\mathsf{s} = 1.45$	4	6	9	14	21
$\mathsf{s} = 1.4$	6	11	17	27	44
$\mathsf{s} = 1.3$	13	23	39	65	110

number of guesses allows a 80 bits attack. Each average has been done on 1000 measurements because the variance was very small. Fig. 3 represents the limit on vulnerable (n, s) parameters. Above the line, the parameters are on average insecure against the guess and determine attack.

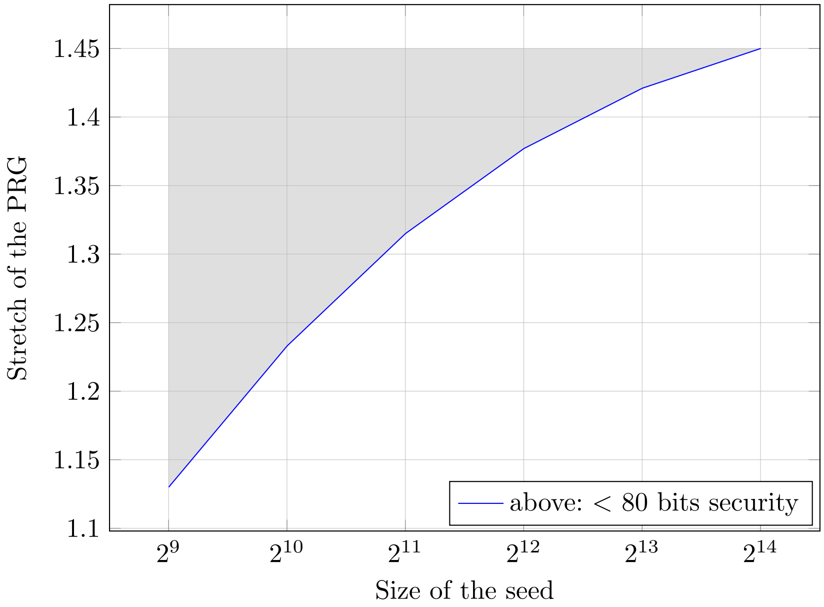

Fig. 3. Limit stretch for vulnerable instances. The gray zone above the curve denotes the insecure choices of parameters.

Candidate Non-Vulnerable Parameters. We were able to estimate the practical range of parameters that appear to resist to this attack. To assess them, we estimated the number of guesses necessary and deduced the bit security. With many measurements (1024 for each set of parameters), we could find the limit stretch for parameters that are, not vulnerable to our attack. The couples (n, s)

Table 4. Challenge parameters for seed recovery attacks. The first line contains the parameter n and below are represented the associated stretches s.

Elementary operations	512	1024	2048	4096
$< 2^{80}$	1.120	1.215	1.296	1.361
$< 2^{128}$	1.048	1.135	1.222	1.295

that possess the maximal s with an expected security of 80 or 128 bits[1] are conjectured to be the limit for non vulnerable parameters. These couples[2] are represented by the two lines in Fig. 4.

We also introduce certain parameters in Table 4 as challenges for improving the cryptanalysis of Goldreich's PRG. These parameters correspond to choices of the seed size and the stretch which cannot be broken in less than 2^{80} (resp. 2^{128}) operations with the attacks of this paper. Further study is required to assess confidence in the security level given by these parameters.

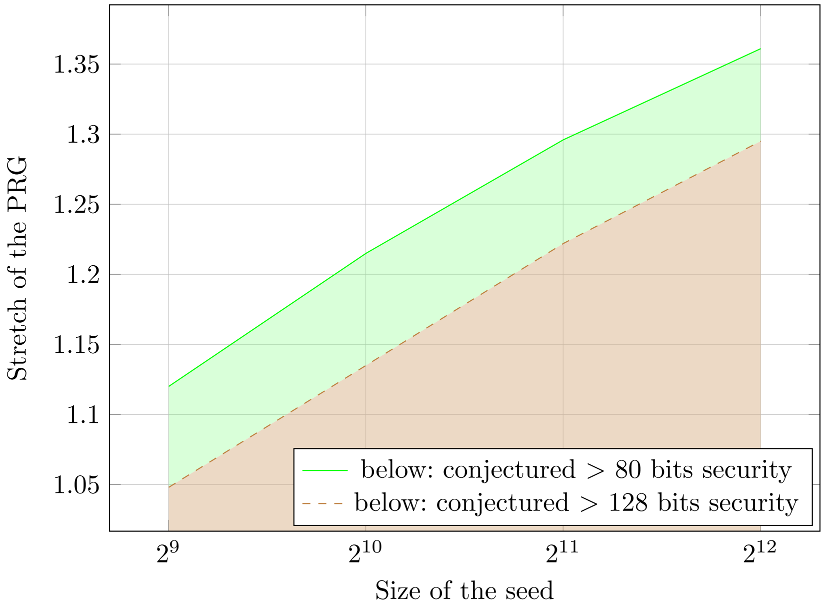

Fig. 4. Limit stretch for conjectured non vulnerable instances.

[1] We actually took a margin of 10% to take into account the possible improvements of our implementation

[2] This curve should not be extrapolated because outside of its range, Gröbner attacks seem more powerful, see Fig. 5

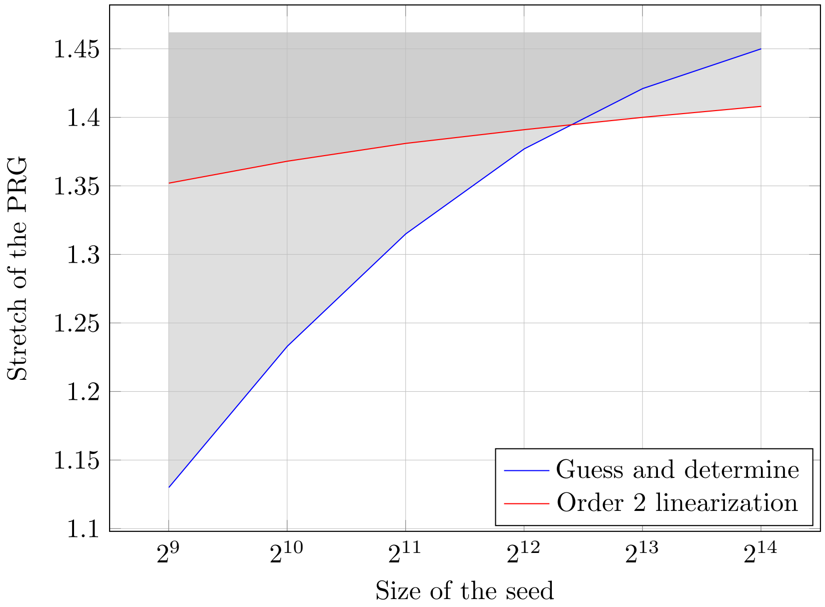

Fig. 5. Limit stretch for vulnerable parameters with 80 bits of security against both guess and determine (Section 3) and order 2 linearization attacks (See the full version of our paper). The gray zone above the curves denotes the insecure choices of parameters.

3.4 Other Algebraic Cryptanalysis

To complement this attack, we also made an analysis of the efficiency of algebraic attacks with Gröbner basis on Goldreich's PRG. While it is known that Goldreich's PRG (and its variants) provably resists such attacks for appropriate choices of (asymptotic) parameters ([AL16], Theorem 5.5), little is known about its exact security against such attacks for concrete choices of parameters.

Since Goldreich's PRG is far from a Boolean random quadratic system, the performance of a Gröbner basis strategy is hard to assess with the existing theory. In order to give an intuition on how Gröbner basis algorithms would behave on Goldreich's PRG with predicate P_5, we provide in the full version of our paper [CDM+18] an easy-to-understand order two linearization attack. This polynomial attack leads to a practical seed recovery for certain parameters (n, s) and we derive a heuristic bound for vulnerable (n, s) for 80 bits of security. The existence of such an attack allows to estimate bounds on Gröbner basis performance. Using an implemented proof of concept, we introduce heuristic bounds for vulnerable parameters. From this linearization attack performance and complexity, we derive a heuristic bound on vulnerable (n, s) parameters against a Gröbner basis technique. We refer the reader to the full version of our paper for the complete analysis.

3.5 Conclusion

We described in this section a guess and determine attack against Goldreich's PRG. In the full version of our paper, we complement this result with an analysis of the security of Goldreich's PRG against an order 2 linearization attack (*à la* Gröbner). We represent on Figure 5 the range of parameters for which Goldreich's PRG is conjectured to have 80 bits of security against those two attacks. As illustrated on the graph, the guess and determine approach targets more parameters for low n while the linearization attack performs better for $n > 4000$. Although Goldreich's PRG is conjectured to be theoretically secure for a stretch approaching 1.5 by an arbitrary constant, our analysis shows that a very large seed must be used to achieve at least 80 bits of security with such stretch. In particular, if a stretch of 1.4 is needed, no seed smaller than 5120 bits should be used. Similarly, for a stretch as small as 1.1, the seed must be at least 512 bits long.

4 Generic Attacks against Goldreich's PRG

Beyond the predicate P_5 we investigate the security of other predicates for higher stretches, and show that the considered criteria are not sufficient to determine the security. In the full version of our paper, we prove that the number of independent annihilators of the predicate has to be taken into account. Hence, the algebraic immunity is not enough, as we provide a new bound on the stretch that refines the theorem of Applebaum and Lovett. On the other side, we provide in this section an improvement of the *guess and determine* technique, combined with an algebraic attack. This generalization can be seen as an hybrid attack as defined in [Bet11].

4.1 A Subexponential-Time Algorithm

The theorem of Applebaum and Lovett for polynomial-time algorithms regarding algebraic attacks can be improved, as shown in the full version of our paper. In this section, we focus on subexponential-time algorithms. The idea here is to generalize our initial attack of Section 3 against the PRG instantiated with the predicate P_5, to all other considered predicates. Therefore we generalize the attack to all $\mathsf{XOR}_\ell\mathsf{M}_k$ predicates and then more particularly to the $\mathsf{XOR}_\ell\mathsf{MAJ}_k$ predicates.

The principle. Let n be the size of the seed of the PRG with stretch s, and let P be a predicate with locality d. The general idea is to guess r variables of the seed, and solve the corresponding system of equations for each possible value of those r bits. For each equation obtained, an equation of smaller or equal degree can be derived using the principle of the algebraic immunity. Then, the complexity of the attack mainly depends on the values of r and the algebraic immunity of the functions we obtain. It corresponds to the general principle of algebraic attacks with guess and determine ([MJSC16]), for which we can affine the complexity in the particular case of $\mathsf{XOR}_\ell\mathsf{M}_k$ predicates. We begin by considering the complexity of an attack targeting the degree of the M predicate after guessing some bits, based on the following remark:

Remark 1. As soon as $k-1$ variables among the k variables of M are fixed, a linear equation can be found, as the output of M depends on only one variable and as XOR_ℓ is linear.

The attack. Our sub-exponential time algorithm works as follows:

step 1 Fix r variables of the seed $(x_{i_1},\dots,x_{i_r})$, with $r \in O\left(n^{\frac{k-s}{k-1}}\right)$.
step 2 For all 2^r possible values of $x_{i_1},\dots,x_{i_r}$, recover the corresponding linear system of equations.
step 3 Solve the system in $(n-r)^\omega$ operations; if there is a contradiction go back to step 2, otherwise add the solution to the list.
step 4 Return the list of solutions.

This attack works as long as the system of linear equations obtained in step 3 above contains an invertible subsystem of size sufficiently large to recover the seed. We then apply Hypothesis 1 with A_n being the linear system obtained by guessing at random the 2^r possible values of $x_{i_1},\dots,x_{i_r}$.

Complexity analysis. The complexity is dominated by Step 3, as we repeat this step 2^r times (we have to solve a system of linear equations of size $n-r$ for each possible values of the r bits), the complexity of this algorithm is sub-exponential: $O(n^\omega 2^r)$. Eventually, the final complexity is determined by the following proposition:

Proposition 3. *For an overwhelming proportion of Goldreich's* PRG *instantiated with a* $\mathsf{XOR}_\ell\mathsf{M}_k$ *predicate, under Hypothesis 1 on step 2 system, the complexity order of the previous algorithm can be approximated by :*

$$2^{n^{\frac{k-s}{k-1}}} \cdot n^\omega \ .$$

The proof is given in the full version of our paper.

Remark 2. It is important to notice that the parameter of this attack does not rely directly on the locality, but only on the number k of variables that appear in the nonlinear part M, hence, it improves the complexity of [BQ09]. Indeed, the generic complexity of Bogdanov and Qiao is roughly $O(2^{n^{1-(s-1)/2d}})$ where d denotes the locality, as our algorithm has a complexity that is in $O\left(n^\omega \cdot 2^{n^{1-(s-1)/(k-1)}}\right)$, with $k-1<d$, by definition of k.

Moreover, the predicate requires a high resiliency to avoid linear attacks, and one of the most natural constructions to build a resilient function is to add an independent linear part to a function. It corresponds to the $\mathsf{XOR}_\ell\mathsf{M}_k$ predicates, which have a resiliency of at least $\ell - 1$ given by the xor part. It is also possible to build resilient functions differently, which seems to be a better choice regarding this attack. For the case of P_5, we have $k = 2$, that gives us an attack in $O(n^\omega 2^{n^{2-s}})$.

Possible improvement. This algorithm only relies on the number of variables of the non-linear part, but not on its algebraic immunity. Instead of fixing variables in order to obtain linear equations in the non-linear part of a $\mathsf{XOR}_\ell\mathsf{M}_k$ predicate, an attacker can fix variables in order to recover equations of degree greater than 1. Indeed, using the algebraic immunity of the M predicate, the attacker can recover such equations by fixing less than k bits in the M part. By doing so, it appears that the relevant criterion regarding this attack is no longer the algebraic immunity, neither the r-bit fixing degree defined in [AL16], but a generalization of the two. The efficiency of the attack will depend on the algebraic immunity of the predicates obtained after doing some guesses, and on the probability of getting predicates (in fewer variables) with this algebraic immunity (or smaller). A lower bound on the algebraic immunity that can be obtained with r guesses is given by the r-bit fixing algebraic immunity (introduced first in term of recurrent algebraic immunity in [MJSC16] to bound the complexity of algebraic attacks combined with guess and determine) defined in the following sense:

Definition 7. *(r-bit fixing algebraic immunity) Let f be a Boolean function with d variables. For any $0 \leq r \leq d$, and $b = (b_1, \ldots, b_r) \in \{0,1\}^r$, $i = (i_1, \ldots, i_r) \in [d]^r$ such that $i_1 < i_2 < \cdots < i_r$, we note $f_{(b,i)}$ the restriction of f where the r variables indexed by $i_1, \ldots, i_r$ are fixed to the value $b_1, \ldots, b_r$. Then f has r-bit fixing algebraic immunity a if*

$$\min\left(\mathsf{AI}(f_{(b,i)}) : i = (i_1, \ldots, i_r) \in [d]^r,\, i_1 < i_2 < \cdots < i_r,\, b \in \{0,1\}^r\right) = a$$

where AI denotes the algebraic immunity.

For the case of $\mathsf{XOR}_\ell\mathsf{M}_k$ predicates we prove in the full version of our paper [CDM$^+$18] an upper bound on the r-bit fixing algebraic immunity. Thereafter, determining the number of predicates with this algebraic immunity that could be reached guessing r variables will lead to other sub-exponential time algorithms. The description and analysis of this algorithm applied on $\mathsf{XOR}_\ell\mathsf{M}_k$ predicates is given in the full version of our paper. However, this algorithm only generalizes the result given by the first algorithm as it considers systems of equations of degree greater than one. But it does not assume any property on the M predicate, and leads to consider the maximum algebraic immunity that can be provided by this part when some variables are fixed. Considering the principle of the r-bit fixing algebraic immunity, we can try to find guesses which lower this algebraic immunity, leading to an attack with even better complexity.

In the following, we show on the XOR-MAJ predicates how only taking into account specific values of guessed bits (but changing the positions that we guess) enables to target a low algebraic immunity with enough equations.

Application to XOR-MAJ Predicates. In the previous algorithms, we fix r bits that never change, but we test all possible values for those bits. However, it might be of interest to change the bits that we guess, by taking into account a specific value for those bits, such that we decrease more drastically the degree of the equations that we get. Using the notations of Definition 7, it boils down to finding values of $b \in \{0,1\}^r$ such that $\mathsf{AI}(f_{(i,b)})$ is low for enough i.

Let us consider the $\mathsf{XOR}_\ell\mathsf{MAJ}_k$ predicate (Definition 3), then our initial algorithm breaks the construction with complexity $O(n^\omega 2^{n^{(k-\mathrm{s})/(k-1)}})$, and its generalization with complexity $O\left(2^{n^{\frac{1+j-\mathrm{s}+\lceil (k-j)/2\rceil}{j}}} n^{\omega(\lceil \frac{k-j}{2}\rceil+1)}\right)$ for all integer j such that $1 \leq j \leq k$. Moreover, this algorithm is an improvement only for bigger stretches. In the following, we change the way we make our guesses, in order to capture how the r-bit fixing algebraic immunity is a relevant criterion.

In these algorithms, one can notice that fixing j bits among the k variables that appear in the majority function can derive different degrees of equations, depending on the value of the bits that are guessed: fixing $\lceil \frac{k}{2} \rceil$ bits all to 0 (or all to 1) will derive directly linear equations. Indeed, for the majority function, if strictly more than half of the bits are supposed to be all zero, then the corresponding output has to be 0 by definition of the majority, and respectively 1 if all these bits are ones. On the other side, fixing a quarter of bits to be ones and a quarter of bits to be zero will derive an other majority function taken other half of the bits, which is clearly non-linear for k big enough.

Hence, instead of fixing r bits and guess all possible values of those bits, we choose r bits, guessing that all those bits are all one or all zero, and repeat this until the guess is right (the position of the r guessed variables changes, not the value). This particular guess-and-determine is exactly what Duval, Lallemand and Rotella investigated in [DLR16] on the FLIP family of stream ciphers (and which complexity can be bounded through the r-bit fixing algebraic immunity, [MJSC16] Section 3.4).

Description of the algorithm.

step 1 Fix randomly r variables of the seed $(x_{i_1}, \ldots, x_{i_r})$.

step 2 Assume that all of them are equal to zero, solve the corresponding linear system, add the solution to the list.

step 3 Assume that all the r variables are equal to one, solve the corresponding linear system, add the solution to the list.

step 4 If in the solution list there is one with no contradiction with the PRG output, output the solution as the seed. Otherwise, empty the list and go back to Step 1.

As for the first algorithm, we assume that Hypothesis 1 is verified with A_n representing the linear systems of Step 2 and 3.

Complexity analysis. The complexity is dominated by the number of repetition of Step 2 and Step 3, we determine it through the following proposition:

Proposition 4. *For an overwhelming proportion of Goldreich's* PRG *instantiated with a* $\mathsf{XOR}_\ell\mathsf{MAJ}_k$ *predicate, under Hypothesis 1 for Step 2 and 3 systems, the seed can be recovered in time complexity of order:*

$$n^\omega 2^{n^{1-\frac{s-1}{\lceil\frac{k}{2}\rceil+1}}}.$$

The proof is given in the full version of our paper.

This algorithm captures something else than the previous ones, as it shows that one has to consider all possible choices of guesses in order to evaluate exactly the security of such constructions. In other words, it shows that the r-bit fixing algebraic immunity is exactly the relevant criterion to resist our attack, as it defines the smallest algebraic immunity that can be considered for an attack. However, one must also take the probability that a corresponding guess happens on the equations into account. Hence there exists a trade-off between the choice of the good guesses, and the probability that the corresponding equation of small degree can be derived.

4.2 Open Questions

The attacks and their variants described here asked lot of open questions. For the polynomial time algorithm using the number of linearly independent annihilators, we do not take into account some dependencies into different equations as explained in the full version of our paper [CDM+18]. Hence, the condition on the stretch that we gave could be improved by considering dependencies on the subsets.

For the subexponential-time attack that uses the r bit fixing algebraic immunity, we do not know if the bound given in the full version of our paper is tight, that is if there exist predicates, such that fixing any bits will still derive Boolean functions with fewer variables that reach the maximal algebraic immunity. In other words, is it possible to have a perfect predicate regarding the r bit fixing algebraic immunity? Recalling that it is the relevant criterion in this context.

Moreover, this bound does not depend on the value of the bits that are guessed, whereas this might have an influence, as shown on the XOR-MAJ predicate. For example, the Boolean function $x_0 + x_1x_2x_3x_4$ is of algebraic immunity 2, but fixing x_1 to be 1 will derive a Boolean function that is still of algebraic immunity 2, but fixing $x_1 = 0$ will bring directly an equation of degree 1. Hence, all choices of guess are not equivalent, implying that different choices of guesses could improve the complexity of our subexponential-time algorithm, depending strongly on the predicate.

Last but not least, how the first idea of using different annihilators can improve the subexponential-time algorithms using guess and determine?

Acknowledgments

We thank Jean-Pierre Tillich and Benny Applebaum for useful discussions and observations. We also are indebted to Guénaël Renault for fruitful discussions about Gröbner basis approaches, and to the reviewers of ASIACRYPT for their useful comments. This research has been partially funded by ANRT under the programs CIFRE N 2015/1158 and 2016/1583. We acknowledge the support of the French Programme d'Investissement d'Avenir under national project RISQ P141580. The first author was supported by ERC grant 724307 (project PREP-CRYPTO). The fifth author was partially supported by the French Agence Nationale de la Recherche through the BRUTUS project under Contract ANR-14-CE28-0015.

References

ABR12. B. Applebaum, A. Bogdanov, and A. Rosen. A dichotomy for local small-bias generators. In *TCC 2012*, *LNCS* 7194, pages 600–617. Springer, Heidelberg, March 2012.

ABR16. B. Applebaum, A. Bogdanov, and A. Rosen. A dichotomy for local small-bias generators. *Journal of Cryptology*, 29(3):577–596, July 2016.

ADI$^+$17a. B. Applebaum, I. Damgård, Y. Ishai, M. Nielsen, and L. Zichron. Secure arithmetic computation with constant computational overhead. Cryptology ePrint Archive, Report 2017/617, 2017. http://eprint.iacr.org/2017/617.

ADI$^+$17b. B. Applebaum, I. Damgård, Y. Ishai, M. Nielsen, and L. Zichron. Secure arithmetic computation with constant computational overhead. In *Crypto'17*, pages 223–254, 2017.

AHI05. M. Alekhnovich, E. A. Hirsch, and D. Itsykson. Exponential lower bounds for the running time of dpll algorithms on satisfiable formulas. *Journal of Automated Reasoning*, 35(1-3):51–72, 2005.

AIK04. B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. In *45th FOCS*, pages 166–175. IEEE Computer Society Press, October 2004.

AIK08. B. Applebaum, Y. Ishai, and E. Kushilevitz. On pseudorandom generators with linear stretch in nc 0. *Computational Complexity*, 17(1):38–69, 2008.

AL16. B. Applebaum and S. Lovett. Algebraic attacks against random local functions and their countermeasures. In *48th ACM STOC*, pages 1087–1100. ACM Press, June 2016.

App12. B. Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In *44th ACM STOC*, pages 805–816. ACM Press, May 2012.

App13. B. Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. *SIAM J. Comput.*, 42(5):2008–2037, 2013.

App15. B. Applebaum. The cryptographic hardness of random local functions – survey. Cryptology ePrint Archive, Report 2015/165, 2015. http://eprint.iacr.org/2015/165.

ARS$^+$15. M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner. Ciphers for MPC and FHE. In *EUROCRYPT 2015, Part I*, *LNCS* 9056, pages 430–454. Springer, Heidelberg, April 2015.

BCG$^+$17. E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, and M. Orrù. Homomorphic secret sharing: Optimizations and applications. In *ACM CCS 17*, pages 2105–2122. ACM Press, 2017.

Bet11. L. Bettale. *Cryptanalyse algebrique : outils et applications.* PhD Thesis, 2011.

BGI$^+$01. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *CRYPTO 2001*, *LNCS* 2139, pages 1–18. Springer, Heidelberg, August 2001.

BQ09. A. Bogdanov and Y. Qiao. On the security of goldreich's one-way function. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 392–405. Springer, 2009.

CCF$^+$16. A. Canteaut, S. Carpov, C. Fontaine, T. Lepoint, M. Naya-Plasencia, P. Paillier, and R. Sirdey. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. In *FSE 2016*, *LNCS* 9783, pages 313–333. Springer, Heidelberg, March 2016.

CDM$^+$18. G. Couteau, A. Dupin, P. Méaux, M. Rossi, and Y. Rotella. On the concrete security of goldreich?s pseudorandom generator. In *To appear on eprint*, 2018.

CEMT14. J. Cook, O. Etesami, R. Miller, and L. Trevisan. On the one-way function candidate proposed by goldreich. *ACM Transactions on Computation Theory (TOCT)*, 6(3):14, 2014.

CM01. M. Cryan and P. B. Miltersen. On pseudorandom generators in nc 0. In *International Symposium on Mathematical Foundations of Computer Science*, pages 272–284. Springer, 2001.

CM03. N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In *EUROCRYPT 2003*, *LNCS* 2656, pages 345–359. Springer, Heidelberg, May 2003.

Cou03. N. T. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In *CRYPTO*, pages 176–194, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

DGM05. D. K. Dalai, K. C. Gupta, and S. Maitra. Cryptographically significant Boolean functions: Construction and analysis in terms of algebraic immunity. In *FSE 2005*, *LNCS* 3557, pages 98–111. Springer, Heidelberg, February 2005.

DLR16. S. Duval, V. Lallemand, and Y. Rotella. Cryptanalysis of the FLIP family of stream ciphers. In *CRYPTO 2016, Part I*, *LNCS* 9814, pages 457–475. Springer, Heidelberg, August 2016.

DMS05. D. K. Dalai, S. Maitra, and S. Sarkar. Basic theory in construction of Boolean functions with maximum possible annihilator immunity. Cryptology ePrint Archive, Report 2005/229, 2005. http://eprint.iacr.org/2005/229.

EJ00. P. Ekdahl and T. Johansson. SNOW - a new stream cipher. In *Proceedings of First NESSIE Workshop*, Heverlee, Belgique, 2000.

GGM84. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions (extended abstract). In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984.

Gol00. O. Goldreich. Candidate one-way functions based on expander graphs. Cryptology ePrint Archive, Report 2000/063, 2000. http://eprint.iacr.org/2000/063.

GRR$^+$16. L. Grassi, C. Rechberger, D. Rotaru, P. Scholl, and N. P. Smart. MPC-friendly symmetric key primitives. In *ACM CCS 16*, pages 430–443. ACM Press, October 2016.

HR00. P. Hawkes and G. G. Rose. Exploiting multiples of the connection polynomial in word-oriented stream ciphers. In *ASIACRYPT 2000*, *LNCS* 1976, pages 303–316. Springer, Heidelberg, December 2000.

IKOS08. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography with constant computational overhead. In *40th ACM STOC*, pages 433–442. ACM Press, May 2008.

IPS08. Y. Ishai, M. Prabhakaran, and A. Sahai. Secure arithmetic computation with no honest majority. Cryptology ePrint Archive, Report 2008/465, 2008.

Lin17. H. Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In *CRYPTO 2017, Part I*, *LNCS* 10401, pages 599–629. Springer, Heidelberg, August 2017.

LT17. H. Lin and S. Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local PRGs. In *CRYPTO 2017, Part I*, *LNCS* 10401, pages 630–660. Springer, Heidelberg, August 2017.

LV17. A. Lombardi and V. Vaikuntanathan. Limits on the locality of pseudorandom generators and applications to indistinguishability obfuscation. In *TCC 2017, Part I*, LNCS, pages 119–137. Springer, Heidelberg, March 2017.

MJSC16. P. Méaux, A. Journault, F.-X. Standaert, and C. Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In *EUROCRYPT 2016, Part I*, *LNCS* 9665, pages 311–343. Springer, Heidelberg, May 2016.

MST03. E. Mossel, A. Shpilka, and L. Trevisan. On e-biased generators in NC0. In *44th FOCS*, pages 136–145. IEEE Computer Society Press, October 2003.

OW14. R. ODonnell and D. Witmer. Goldreich's prg: evidence for near-optimal polynomial stretch. In *Computational Complexity (CCC), 2014 IEEE 29th Conference on*, pages 1–12. IEEE, 2014.

Sie84. T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications (corresp.). *IEEE Transactions on Information theory*, 30(5):776–780, 1984.

SW14. A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

Wie86. D. Wiedemann. Solving sparse linear equations over finite fields. *IEEE transactions on information theory*, 32(1):54–62, 1986.