Worst-Case Subexponential Attacks on PRGs of Constant Degree or Constant Locality

Akın Ünal

Department of Computer Science
ETH Zurich
Zurich, Switzerland
akin.uenal@inf.ethz.ch

February 1, 2023

Abstract. In this work, we will give new attacks on the pseudorandomness of algebraic pseudorandom number generators (PRGs) of polynomial stretch. Our algorithms apply to a broad class of PRGs and are in the case of general local PRGs faster than currently known attacks. At the same time, in contrast to most algebraic attacks, subexponential time and space bounds will be proven for our attacks without making any assumptions of the PRGs or assuming any further conjectures. Therefore, we yield in this text the first subexponential distinguishing attacks on PRGs from constant-degree polynomials and close current gaps in the subexponential cryptoanalysis of lightweight PRGs.
Concretely, against PRGs $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ that are computed by polynomials of degree d over a field $\mathbb{Z}_q$ and have a stretch of $m = n^{1+e}$ we give an attack with space and time complexities $n^{O(n^{1-\frac{e}{d-1}})}$ and noticeable advantage $1 - O(n^{1-\frac{e}{d-1}}/q)$, if q is large. If F is of constant *locality* d and q is constant, we construct a second attack that has a space and time complexity of $n^{O(\log(n)^{\frac{1}{(q-1)d-1}} \cdot n^{1-\frac{e}{(q-1)d-1}})}$ and noticeable advantage $1 - O((\log(n)/n^e)^{\frac{1}{(q-1)d-1}})$.

1 Introduction

A pseudorandom number generator (PRG) is a deterministic algorithm $F : \{0,1\}^n \to \{0,1\}^m$ that stretches a given string of bits i.e. $m > n$. We expect a PRG to expand a uniformly drawn string to a longer string of bits that sufficiently simulates randomness. More formally, for a PRG F its output – when evaluated on a short uniformly random string – should be for a certain class of computational models indistinguishable from a longer uniformly random string, even if the algorithm F is publicly known.

PRGs are an important tool in the toolbox of cryptography besides one-way functions [1, 27], pseudorandom permutations and pseudorandom functions. Further, in complexity theory, the existence of PRGs implies the derandomization of certain complexity classes [38]. For example, it is known that the existence of so-called *high-end* PRGs implies that **P** equals **BPP** [29]. Additionally, PRGs have the real world task of simulating cryptographic pseudorandomness in deterministic software applications.

Of particular interest are PRGs that can be efficiently evaluated. Very prominent examples are *local* PRGs [26]. Each output bit of a local PRG depends on only a constant number of input bits. Besides their simplicity, local PRGs are an important building block in advanced cryptographic constructions, e.g. two-party protocols for computing circuits with constant overhead [30] or indistinguishability obfuscation [31, 32]. Assuming additionally the pseudorandomness of arithmetic PRGs $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ where each output value is computed by a polynomial of constant degree over $\mathbb{Z}_q$ leads to arithmetization of such primitives, like e.g. arithmetic two-party protocols [4].

Since PRGs play such a crucial role in cryptography, cryptoanalysis of PRGs is of general importance. In particular, local PRGs $F : \{0,1\}^n \to \{0,1\}^m$ of poly-stretch, i.e. $m \geq n^{1+e}$ for some constant $e > 0$, have been the subject of various attacks, and it could be shown that such PRGs can be distinguished by subexponential-size [1] circuits, or even poly-size circuits if $e > 0.5$ [3, 5, 11, 17, 39, 41].

PRGs of constant degree, i.e. PRGs that can be computed by polynomials of constant degree over some finite field, can be seen as a generalization of local PRGs. However, constant-degree PRGs have received much less attention in cryptoanalytic literature than local PRGs. While there is a huge collection of algebraic attacks on refuting and inverting constant-degree PRGs like F4/F5 and the XL-algorithms [15, 16, 19, 23, 24, 36, 44], we do not know of any attacks whose time-complexity for poly-stretch constant-degree PRGs is guaranteed to be subexponential even in the worst case. We intend to close this gap by introducing a new algebraic attack that is provably subexponential against poly-stretch PRGs of constant degree.

1.1 Contribution

In this text, we will introduce new algebraic attacks on PRGs and prove upper bounds for their complexities and lower bounds for their advantages in the worst case. Let $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ with $m \geq n^{1+e}$. Then, we give the following attacks on the pseudorandomness of F:

- If F is of degree d over $\mathbb{Z}_q$, we have an attack with subexponential space and time complexities $n^{O(n^{1-\frac{e}{d-1}})}$. The advantage of this attack is $1 - O(n^{1-\frac{e}{d-1}}/q)$, which is noticeable if q is large enough.
- If q is constant and F is of *locality* d, we give a second attack with subexponential space and time complexities $n^{O(\log(n)^{\frac{1}{(q-1)d-1}} \cdot n^{1-\frac{e}{(q-1)d-1}})}$. For this attack, we will prove a noticeable advantage of $1 - O\left((\log(n)/n^e)^{\frac{1}{(q-1)d-1}}\right)$.

[1] The notion of *subexponentiality* is ambiguous in literature. Here, we denote by subexponential a function that is contained in $\bigcup_{c<1} 2^{O(n^c)}$.

- Additionally, if q should be small (e.g. $q \in O(n^{1-\frac{e}{d-1}})$), we give a third attack for PRGs F of constant degree d with complexities $n^{O(n^{1-\frac{e}{d-1}})}$ for which we can guarantee a subexponentially small advantage of $q^{-O(n^{1-\frac{e}{d-1}})}$. We give the details of this attack in Appendix A.1.

To the best of our knowledge, we give the first distinguishing algorithms on constant-degree PRGs that are provably subexponential in the worst case for sufficiently large moduli. Additionally, our second and third attack algorithms are faster than the attacks of Bogdanov & Qiao [11], which work against general local PRGs, and are almost as fast as the attacks of Couteau *et al.* [17], which only work against special local PRGs.

Furthermore, we can draw important insights for Groebner basis-based algorithms. In Section 6, we will compare the algorithms here with typical Groebner basis-based attacks and prove an upper bound for the *degree of regularity*, a popular heuristic for estimating the complexity of most Groebner basis-based algorithms.

1.2 Technical Overview

We want to motivate and explain here the ideas behind our new attacks. Let q be a prime, $\mathbb{Z}_q$ be the finite field of size q and $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ be a PRG of degree d. I.e., the i-th output value of F is computed by a polynomial $f_i \in \mathbb{Z}_q[X] := \mathbb{Z}_q[X_1, \ldots, X_n]$ of total degree $\leq d$. Now, assume we would know a non-zero polynomial $h \in \mathbb{Z}_q[Y] := \mathbb{Z}_q[Y_1, \ldots, Y_m]$ that vanishes on the image of F i.e.

$$h(F(x)) = 0 \tag{1}$$

for all $x \in k^n$. Let D be the total degree of h. Since h is not the zero polynomial, we have according to the famous Schwartz-Zippel lemma [40]

$$\Pr_{y \leftarrow \mathbb{Z}_q^m}[h(y) = 0] \leq \frac{D}{q}. \tag{2}$$

I.e., while h will always be zero on the image of F, the probability that h vanishes on a random point can be controlled by D/q. If D is sublinear and q is sufficiently large, $q \geq n$ for example, h gives us a strong indicator for distinguishing image points of F from random points of $\mathbb{Z}_q^m$. In fact, by using h we can distinguish the distribution $(F(x))_{x \leftarrow \mathbb{Z}_q^n}$ from $(y)_{y \leftarrow \mathbb{Z}_q^m}$ with advantage at least $1 - \frac{D}{q}$.

However, the following two questions remain:

1. For which degrees D can we guarantee the existence of a non-zero polynomial h of degree D that vanishes on the image of F?
2. Even if we know that a such a polynomial h must exist, how can we algorithmically compute it?

Finding Algebraic Relations. The set of polynomials h that vanish on each $F(x)$ has a specific algebraic structure. To explore this structure, we consider the following morphism of $\mathbb{Z}_q$-algebras:

$$\begin{aligned} \phi : \mathbb{Z}_q[Y_1, \ldots, Y_m] &\longrightarrow \mathbb{Z}_q[X_1, \ldots, X_n] && (3) \\ g(Y_1, \ldots, Y_m) &\longmapsto g(f_1(X), \ldots, f_m(X)). && (4) \end{aligned}$$

ϕ maps polynomials in $\mathbb{Z}_q[Y]$ to polynomials in $\mathbb{Z}_q[X]$ by substituting each variable Y_i by the polynomial $f_i(X)$. Denote by $\ker \phi$ the kernel of ϕ, i.e.

$$\ker \phi = \{g \in \mathbb{Z}_q[Y] \mid \phi(g) = 0\}\,. \tag{5}$$

If g lies in $\ker \phi$, we have $\phi(g) = g(f_1(X), \ldots, f_m(X)) = 0$. In particular, we have for each $x \in \mathbb{Z}_q^n$ then

$$g(f_1(x), \ldots, f_m(x)) = \phi(g)(x_1, \ldots, x_m) = 0. \tag{6}$$

This means, the kernel of ϕ contains polynomials h that are of interest for us.

Therefore, we can restate our questions as follows:

1. For what D can we guarantee the existence of a non-zero element of $\ker \phi$?
2. How can we compute all elements of $\ker \phi$ up to degree D?

To answer the first question, we define the following $\mathbb{Z}_q$-vector spaces for $\ell \in \mathbb{N}$:

$$\begin{aligned} \mathbb{Z}_q[X]^{\leq \ell} &:= \{g \in \mathbb{Z}_q[X] \mid \deg g \leq \ell\}\,, && (7) \\ \mathbb{Z}_q[Y]^{\leq \ell} &:= \{g \in \mathbb{Z}_q[Y] \mid \deg g \leq \ell\}\,. && (8) \end{aligned}$$

The vector spaces $\mathbb{Z}_q[X]^{\leq \ell}$ and $\mathbb{Z}_q[Y]^{\leq \ell}$ contain all elements of $\mathbb{Z}_q[X]$ resp. $\mathbb{Z}_q[Y]$ of total degree $\leq \ell$. They are spanned by all monomials in the X- resp. Y-variables of degree $\leq \ell$. Therefore, we have

$$\dim_{\mathbb{Z}_q} \mathbb{Z}_q[X]^{\leq \ell} = \binom{n+\ell}{\ell} \quad \text{and} \quad \dim_{\mathbb{Z}_q} \mathbb{Z}_q[Y]^{\leq \ell} = \binom{m+\ell}{\ell}. \tag{9}$$

Now, we want to restrict ϕ on $\mathbb{Z}_q[Y]^{\leq \ell}$. Remember that F is a PRG of degree d, i.e., each f_i is a polynomial of degree d. It is easy to see that ϕ stretches the degree of each polynomial by at most a factor of d. I.e., we have for each $g \in \mathbb{Z}_q[Y]$

$$\deg \phi(g) = \deg g(f_1(X), \ldots, f_m(X)) \leq d \cdot \deg g. \tag{10}$$

So, by restricting ϕ on $\mathbb{Z}_q[Y]^{\leq \ell}$, we get a linear map

$$\phi^\ell : \mathbb{Z}_q[Y]^{\leq \ell} \longrightarrow \mathbb{Z}_q[X]^{\leq d \cdot \ell} \tag{11}$$

for each ℓ. For linear maps, it is quite easy to guarantee the existence of non-trivial kernel elements. In fact, by dimension formulas, we have

$$\begin{aligned} \dim_{\mathbb{Z}_q} \ker \phi^\ell &\geq \dim_{\mathbb{Z}_q}(\mathbb{Z}_q[Y]^{\leq \ell}) - \dim_{\mathbb{Z}_q}(\mathbb{Z}_q[X]^{\leq d \cdot \ell}) && (12) \\ &= \binom{m+\ell}{\ell} - \binom{n+d\cdot\ell}{d\cdot\ell}. && (13) \end{aligned}$$

Therefore, it suffices to find the smallest D s.t.

$$\binom{m+D}{D} > \binom{n+d\cdot D}{d\cdot D}. \tag{14}$$

As we already stated, we are interested here in PRGs of poly-stretch, so let $e > 0$ be constant s.t. $m \geq n^{1+e}$. We claim that inequality Eq. (14) holds for $D \in \Omega(n^{1-\frac{e}{d-1}})$. To see this, note that we have

$$\begin{aligned}
&\binom{m+D}{D} > \binom{n+d\cdot D}{d\cdot D} && (15)\\
\iff &\frac{(m+D)\cdots(m+1)}{D\cdots 1} > \frac{(n+dD)\cdots(n+1)}{(dD)\cdots 1} && (16)\\
\iff &(m+D)\cdots(m+1)\cdot(dD)\cdots(D+1) > (n+dD)\cdots(n+1). && (17)
\end{aligned}$$

To show Eq. (17), we lower bound the LHS terms $(dD)\cdots(D+1) > D^{(d-1)D}$ and $(m+D)\cdots(m+1) > m^D$. Further, for the simplicity of this exposition, we approximate $(n+dD)\cdots(n+1)$ by n^{dD}. We then get roughly

$$\begin{aligned}
&(m+D)\cdots(m+1)\cdot(dD)\cdots(D+1) && (18)\\
&> m^D \cdot D^{(d-1)D} && (19)\\
&\geq n^{(1+e)D}\cdot n^{(1-\frac{e}{d-1})\cdot(d-1)D} && (20)\\
&= n^{(1+e)D+(d-1-e)D} && (21)\\
&= n^{dD} \approx (n+dD)\cdots(n+1). && (22)
\end{aligned}$$

This shows that the degree $D \in \Omega(n^{1-\frac{e}{d-1}})$ is a plausible bound for non-trivial elements in $\ker\phi$. In Section 3, we will show that we can choose any $D \geq c\cdot n^{1-\frac{e}{d-1}}$ for a constant $c \in (2,4]$ that depends on d.

The above considerations also give us a straight-forward algorithm for computing a non-zero element $h \in \ker\phi$: For each $\ell = 1,\ldots,D$, we compute a matrix representation of the linear map

$$\phi^\ell : \mathbb{Z}_q[Y]^{\leq \ell} \longrightarrow \mathbb{Z}_q[X]^{\leq d\cdot\ell}. \tag{23}$$

By using Gaussian elimination, we can then check if this matrix has a non-trivial kernel vector. Such a non-trivial kernel vector corresponds to a non-trivial kernel element $h \in \ker\phi$ of degree ℓ. By our observations above, we know that for $\ell = D = c\cdot n^{1-\frac{e}{d-1}}$, this algorithm must eventually find a non-zero polynomial.

The space and time complexities of this algorithm is in each step dominated by computing the Gaussian elimination of a matrix of shape $M_\ell \times N_\ell$ where $M_\ell = \binom{m+\ell}{\ell}$ and $N_\ell = \binom{n+d\ell}{d\ell}$. Therefore, we need to store $M_D\cdot N_D \in n^{O(n^{1-\frac{e}{d-1}})}$ field elements and perform $D\cdot M_D\cdot N_D^2 \in n^{O(n^{1-\frac{e}{d-1}})}$ arithmetic operations in $\mathbb{Z}_q$.

Evaluating h on a point $y \in \mathbb{Z}_q^m$ costs $D\cdot M_D \in n^{O(n^{1-\frac{e}{d-1}})}$ field operations. The advantage of using h in distinguishing a random point from an image point of F is at least $1 - D/q$. Hence, for $q \in \omega(n^{1-\frac{e}{d-1}})$ and $m \geq n^{1+e}$, we have an attack algorithm with noticeable advantage, which is subexponential in the worst case.

We give a detailed description of the algorithms sketched here and formal proofs for their correctness in Section 3 and Section 4.

Handling Small Moduli. Note, that we cannot guarantee any advantage of the above algorithm if $q \leq D = cn^{1-\frac{e}{d-1}}$. In fact, it may be that the above algorithm will retrieve a polynomial $h \in \mathbb{Z}_q[Y]$ that vanishes on almost all points of $\mathbb{Z}_q^m$. In the general case of constant-degree PRGs F, one can improve the above algorithm s.t. the found polynomial h is *reduced* modulo the field equations $Y_1^q - Y_1, \ldots, Y_m^q - Y_m$. I.e., each monomial term of h contains each variable Y_i at most $q-1$ times. For such reduced polynomials h of degree $D = O(n^{1-\frac{e}{d-1}})$, one can show that their probability to vanish on a random point $y \leftarrow \mathbb{Z}_q^m$ is upper-bounded by $1 - q^{-D}$. This gives a distinguishing attack for the PRG F with subexponentially small advantage $q^{-D} = q^{-O(n^{1-\frac{e}{d-1}})}$. We detail this attack in Appendix A.1.

Local PRGs of Constant Moduli. While the advantage of the above attack may be much higher in practice (since the probability that h vanishes on a random point may be higher than q^{-D}), from a theoretical point of the view the postulated subexponential advantage is not satisfying.

Fortunately, in the case where the modulus q is constant and F is of constant locality, we can use a little trick to noticeably boost the advantage of our attack. For simplicity, we will assume here that q is 2, however the following approach works for each constant modulus:

Let $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ be of locality d. This means, the i-th bit of the output of F is computed by a function $f_i : \mathbb{Z}_2^n \to \mathbb{Z}_2$ that only depends on d of its inputs. Choose a prime number $p \in [n, 2n]$ and note that – due to the locality of F – for each f_i we can find a polynomial $f_i' \in \mathbb{Z}_p[X]$ of degree d that coincides with f_i on $\{0,1\}^n$, i.e., we have $f_i'(x) = f_i(x)$ for each $x \in \{0,1\}^n$. So, instead of attacking the pseudorandomness of F, we can focus on the pseudorandomness of the map $F' : \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ of degree d that consists of the polynomials $f_1', \ldots, f_m'$. However, distinguishing a random point $y \leftarrow \mathbb{Z}_p^m$ from $F'(x) = F(x)$, for $x \leftarrow \{0,1\}^n$, is obviously simple, since the latter will always lie in $\{0,1\}^m$. To come up for that, we set $m' := \frac{m}{3 \log p}$ and draw a uniformly random matrix $A \leftarrow \mathbb{Z}_p^{m' \times m}$. According to the Leftover Hash Lemma, the distributions

$$(A, Ay)_{y \leftarrow \{0,1\}^m} \quad \text{and} \quad (A, y')_{y' \leftarrow \mathbb{Z}_p^{m'}} \tag{24}$$

are statistically very close. Therefore, if $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ is pseudorandom, then the map $G : \mathbb{Z}_p^n \to \mathbb{Z}_p^{m'}$ of degree d that maps x to $A \cdot F'(x)$ must be, too. However, we can apply our first attack against G. Since $m \geq n^{1+e}$, we have $m' \geq n^{1+e}/(3 \log p)$. We will show that this results in an attack of time and space complexity $n^{O(\log(n)^{\frac{1}{d-1}} \cdot n^{1-\frac{e}{d-1}})}$ and noticeable advantage

$$1 - O(\log(n)^{\frac{1}{d-1}} \cdot n^{1-\frac{e}{d-1}})/p \geq 1 - O((\log(n)/n^e)^{\frac{1}{d-1}}). \tag{25}$$

Going back to F, we get an algorithm of subexponential complexity that has a noticeable advantage in distinguishing images of F from random bit strings $y \leftarrow \{0,1\}^m$. We detail this attack in Appendix A.1.

1.3 Related Work

We try to give here a short survey of the current cryptoanalytic literature on PRGs.

Linear Tests and Low-Degree Correlation. A *linear test* for a PRG $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ is a degree-1 polynomial $L \in \mathbb{Z}_q[Y]$ that has a noticeable advantage

$$\left| \Pr_{x \leftarrow \mathbb{Z}_q^n}[L(F(x)) = 0] - \Pr_{y \leftarrow \mathbb{Z}_q^m}[L(y) = 0] \right| \tag{26}$$

in distinguishing random points from image points of F. While linear tests form a very simple class of attacks against PRGs, it can be shown that they are a good sanity check in the case of local PRGs: a local random PRG that is secure against linear tests also fools other classes of distinguishers like e.g. $\mathbf{AC}^0$, l-wise tests and degree-2 threshold functions [2, Proposition 4.10]. Mossel *et al.* [37] shows that there exist PRGs of constant locality s.t. each linear test only has negligible advantage against those PRGs, even if the PRG is of polynomial stretch $m = n^{1+e}$. Their construction is based on the famous tri-sum-and predicate

$$X_1X_2 + X_3 + X_4 + X_5 \tag{27}$$

that gets applied on random subsets of the input to compute the output bits of the PRG.

If we allow the degree of L to be greater than 1, we get a polynomial test of higher degree. Viola [43] showed that for each constant d a PRG can be constructed that cannot be distinguished by degree-d tests with noticeable advantage (his constructions allows non-constant values for d, however such d reduce the stretch of the PRG substantially).

Groebner Basis-Based Attacks. A huge class of attacks against PRGs of constant degree constitute of algebraic attacks [15, 16, 19, 23, 24, 36, 44]. These attacks aim to invert the potential image of a PRG by computing a Groebner basis or something similar in the case of XL-algorithms.

These algorithms work well in practice, and it has been suspected that they give subexponential attack algorithms against PRGs of polynomial stretch [9]. However, computing a Groebner basis can be a task of double exponential complexity in the worst case, and therefore those algorithms do not give us provable subexponential attacks. In Section 6, we will give a deeper comparison of our algorithms with Groebner basis-based algorithms.

Random Local Functions. A *random local function* is a PRG $F : \{0,1\}^n \to \{0,1\}^m$ where each output bit is computed by a fixed predicate $P : \{0,1\}^d \to \{0,1\}$ that is applied on a random subset of bits of the input string. The notion of random local functions has been put forth by Goldreich [26] and was the subject of a great body of cryptoanalytic literature. For exhaustive surveys and studies on the security of random local functions, we refer the reader to the works of Applebaum [2] and Couteau *et al.* [17]. We will only review here some attacks on random local functions, which we think are the most relevant for the context of this work:

1. It is known that F can be inverted in polynomial time and with high probability if $m \in \Omega(\log(n) \cdot n^{\frac{\lfloor 2d/3 \rfloor}{2}})$ [2]. First note, that F can be efficiently inverted by linearization of the corresponding polynomial equation system if it is of stretch $m \in \omega(n^{\deg P})$, where $\deg P$ denotes the degree of P as a polynomial over $\mathbb{F}_2$.
   This means, the degree of P must be greater than $d/3$ if we want to avoid the above attack for $m \geq n^{\frac{d}{3}}$. However, if $\deg P \geq d/3$, then P is correlated with the sum of $c \leq d - \frac{d}{3}$ of its variables [41]. I.e., P can be written as
   $$P(Z_1, \ldots, Z_d) = Z_1 + \ldots + Z_c + N(Z_1, \ldots, Z_d) \tag{28}$$
   where N is a biased predicate i.e. $\Pr_{z \leftarrow \{0,1\}^d}[N(z) = 0] \neq \frac{1}{2}$. When solving the system $F(x) = y$, one can see the N predicates as dependent noise added to linear equations. This constrained noisy linear equation system can be solved efficiently if $m \in \Omega(n^{c/2})$ [14, 25].
2. There is a subexponential inversion attack [2, 11] on $F(x)$ that utilizes approximations of the correct inverse and has a runtime complexity of $2^{O(n^{1-\frac{e}{2d}})}$ (if $m \geq n^{1+e}$). The idea is to assign random bits to the first $(1 - 2n^{-\frac{e}{2d}})$ bits of an approximate solution. By iterating over all possible $x' \in \{0,1\}^n$ with the given prefix, one will find an approximation that coincides with x on at least $(\frac{1}{2} + n^{-\frac{e}{2d}})n$ of its bits with probability at least $\frac{1}{2}$. This approximation can now be used to find efficiently and with high probability the correct solution x.
   Note, that the time complexity $2^{O(n^{1-\frac{e}{2d}})}$ of this algorithm is worse than the time complexity $n^{O(\log(n)^{\frac{1}{d-1}} \cdot n^{1-\frac{e}{d-1}})}$ of the algorithm we sketched against d-local PRGs of stretch n^{1+e}.
3. Couteau *et al.* [17] constructed a guess-and-determine-style attack on PRGs $F : \{0,1\}^n \to \{0,1\}^{n^{1+e}}$. Their attack guesses – in an intelligent way – a portion of the bits of x and tries to extract a linear equation system from the system $F(x) = y$ for the unguessed input bits. If the predicate P for F is of the form
   $$P(X_1, \ldots, X_r) = X_1X_2 + X_3 + \ldots + X_r, \tag{29}$$
   they can prove that their attack will succeed in distinguishing random points from images of F and has a time complexity of $2^{O(n^{1-e})}$. Note, that r does not need to be constant.

They even generalize their attack to work with general predicates

$$P(X_1, \dots, X_r) = M(X_1, \dots, X_d) + X_{d+1} + \dots + X_r, \tag{30}$$

for any predicate $M : \{0,1\}^d \to \{0,1\}$ with d constant, and get an attack algorithm of time complexity $2^{O(n^{1-\frac{e}{d-1}})}$. However, to prove a high success probability of the generalization of their attack they need to assume a special conjecture that depends on M.

4. While there are a lot of efficient attacks against local PRGs of sufficient stretch, it is known that algebraic attacks against d-local PRGs of stretch n^{1+e} will have a time complexity of at least $2^{O(n^{1-32\frac{e}{d-2}})}$ in the worst case [2, Theorem 5.5]. This means, up to some constants in the exponent, the time complexities we achieve with our attacks are optimal for algebraic attacks.

Attacks Based on Sum-of-Squares. Sum-of-Squares attacks are a special class of SDP-based attacks. These attacks were discovered recently and used to refute several candidate light-weight PRGs of polynomial stretch for indistinguishability obfuscation schemes [7, 8]. While these attacks are efficient, they need to make special assumptions about the PRGs they attack, which limits the generality of those attacks. We will list below some PRGs for which a sum-of-squares attack can successfully distinguish PRG images from random points:

1. Let $F : \{0,1\}^{nb} \to \{0,1\}^m$ be *two block-local*, i.e., the input is partitioned into n blocks of size b and each output depends on two blocks. If $m \in \Omega(2^{2b} \cdot \log^2(n) \cdot n)$ is big enough, then there is an efficient attack on F [7].
2. Let $c > 0$ be a constant and let Y be a distribution over $\mathbb{R}$ s.t. we have $\Pr_{y \leftarrow Y}[y \notin [a, a+c]] \geq \frac{1}{10}$ for each $a \in \mathbb{R}$. Let $F : \{0,1\}^n \to \mathbb{R}^m$ be a PRG of degree d over the reals s.t. the polynomials in F have at most s monomials. If $m \in \Omega(\log^2(n) \cdot s \cdot n^{\lceil d/2 \rceil})$ is big enough and if we assume a special assumption for the polynomials $f_1, \dots, f_m$, there is an efficient attack that can successfully distinguish images of F from points $y \leftarrow Y^m$ [7].
3. Let $t \in \mathrm{poly}(n)$ and let Q be a distribution of quadratic polynomials in $\mathbb{R}[X]$ with some special properties. If $m \in \log(n)^{\Omega(1)} \cdot n$ is big enough, there is an efficient algorithm that can extract with high probability the input x from $(F, F(x))$ where we sample $x \leftarrow [-t, t]^n$ and $F \leftarrow Q^m$ [8].

1.4 Organization of this Text

In Section 2, we will introduce some algebraic and cryptographic preliminaries. In Section 3, we will give an algorithm that finds non-trivial polynomials that vanish on the images of PRGs $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ of constant degree d and prove that one can find such polynomials of sublinear degree if F is of polynomial stretch $m = n^{1+e}$. In Section 4, we will give a distinguishing attack on F of time and space complexity $n^{O(n^{1-\frac{e}{d-1}})}$ and prove that it has an advantage of at least $1 - O(n^{1-\frac{e}{d-1}}/q)$.

In Section 5, we will use the attack of Section 4, to derive an attack on PRGs $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ of constant locality d over a constant modulus q. This attack will have complexities in $n^{O(\log(n)^{\frac{1}{(q-1)d-1}} \cdot n^{1-\frac{e}{(q-1)d-1}})}$.

Finally, in Section 6, we will give an exhaustive comparison between our algorithms and Groebner basis-based algorithms. In this section, we will prove a lower bound for the degree of regularity.

In Appendix A.1, we will investigate the case of small constant moduli q. We will show in this section, that one can find a polynomial of sublinear degree that vanishes on the image of a degree-d PRG $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ of stretch $m = n^{1+e}$, but does not vanish everywhere on $\mathbb{Z}_q^m$. This leads to a second attack on degree-d PRGs of complexity $n^{O(n^{1-\frac{e}{d-1}})}$ and subexponential advantage $q^{-O(n^{1-\frac{e}{d-1}})}$.

In Appendix A.3, we will give some algebraic background.

Acknowledgements. I want to thank my doctoral supervisor, Dennis Hofheinz, for proofreading this text. Further, I would like to thank the reviewers for their questions and suggestions.

2 Preliminaries

2.1 Notation

Denote by $\mathbb{N} = \{1, 2, 3, \ldots\}$ the set of natural numbers and by $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ the set of natural numbers plus zero.

For the rest of this text, by k we will always denote a field and by $k[X_1, \ldots, X_n]$ resp. $k[Y_1, \ldots, Y_m]$ the corresponding polynomial ring, for $n, m \in \mathbb{N}$. Since the numbers of X and Y variables will always be n resp. m, by abuse of notation, we will write $k[X]$ resp. $k[Y]$ instead of $k[X_1, \ldots, X_n]$ resp. $k[Y_1, \ldots, Y_m]$.

Let $f \in k[X]$. When we speak of f's *degree* we always mean its *total degree* that is the minimum number $d \in \mathbb{N}_0$ s.t. f can be written as a k-linear combination of monomials that are the product of $\leq d$ variables.

If S is a finite set, we denote by $x \leftarrow S$ the fact that the random variable x is drawn uniformly and independently at random from S.

For a number $q \in \mathbb{N}$, we define the finite ring $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$.

We will denote by n the security parameter in this text. The parameter $m = m(n)$ will in most cases be dependent on n. For this to be consistent, we assume in those cases that m is time-constructible.

We call a function $\epsilon : \mathbb{N} \to [0, 1]$ negligible, if we have $\lim_{n\to\infty} \epsilon(n) \cdot n^d =$ for each $d \in \mathbb{N}$. By $\mathrm{poly}(n) := \left\{ f : \mathbb{N} \to \mathbb{N} \mid \exists c, d \in \mathbb{N} : f(n) \leq n^d + c \right\}$ we denote the set of polynomially bounded functions of the natural numbers.

Given two discrete distributions $\mathcal{X}$ and $\mathcal{Y}$, we define their statistical distance as $\Delta(\mathcal{X}, \mathcal{Y}) := \frac{1}{2} \sum_x |\mathcal{X}(x) - \mathcal{Y}(x)|$.

Given two k-vector spaces V and W, we denote by $V \oplus W$ their direct sum, i.e. it must hold $V \cap W = 0$.

2.2 Mathematical Preliminaries

We will introduce now some basic facts and notions for the polynomial ring $k[X]$:

Remark 1. Let $n \in \mathbb{N}$. Let k be any field and consider the polynomial ring $k[X] = k[X_1, \dots, X_n]$. The ring $k[X]$ is graded and can be written as

$$k[X] = \bigoplus_{\ell=0}^{\infty} k[X]^{\ell} \tag{31}$$

where $k[X]^{\ell}$ is the finite-dimensional k-vector space generated by all monomials of total degree $= \ell$, i.e.

$$k[X]^{\ell} = \operatorname{span}_k \left\{ X_1^{a_1} \cdots X_n^{a_n} \mid a_1, \dots, a_n \in \mathbb{N}_0, a_1 + \ldots + a_n = \ell \right\}. \tag{32}$$

By $k[X]^{\leq \ell}$ we denote the space generated by all monomials of degree $\leq \ell$, i.e.

$$k[X]^{\leq \ell} := \bigoplus_{i=0}^{\ell} k[X]^{i}. \tag{33}$$

The dimensions of $k[X]^{\ell}$ and $k[X]^{\leq \ell}$ are given by

$$\dim_k k[X]^{\ell} = \binom{n+\ell-1}{\ell} \quad \text{and} \quad \dim_k k[X]^{\leq \ell} = \binom{n+\ell}{\ell}. \tag{34}$$

Sometimes, we will use the notion $X^{\alpha_1}, X^{\alpha_2}, \dots$ to denote monomials

$$X_1^{a_{1,1}} \cdots X_n^{a_{1,n}}, X_1^{a_{2,1}} \cdots X_n^{a_{2,n}}, \dots. \tag{35}$$

In those cases, the $\alpha_1, \alpha_2, \dots \in \mathbb{N}_0^n$ are multi-indices given by

$$\alpha_i = (a_{i,1}, \dots, a_{i,n}). \tag{36}$$

Definition 1 (Dual Morphisms). *Let k be any field and $k[X] = k[X_1, \dots, X_n]$. Let $f_1, \dots, f_m \in k[X]$ and $k[Y] = k[Y_1, \dots, Y_m]$. The function*

$$F : k^n \longrightarrow k^m \tag{37}$$

$$x \longrightarrow (f_1(x), \dots, f_m(x)) \tag{38}$$

*gives us a geometrical map that is continuous in the Zariski topology. It has a **dual morphism** of k-algebras*

$$\phi : k[Y] \longrightarrow k[X] \tag{39}$$

$$Y_i \longrightarrow f_i(X) \tag{40}$$

that maps each polynomial $h \in k[Y]$ to a polynomial $h(f_1(X), \dots, f_m(X))$ in $k[X]$ by substituting each appearance of Y_i in h by f_i for each $i \in [m]$.

Definition 2 (Algebraic Independence). *We call $f_1, \dots, f_m$* ***algebraically independent*** *if the morphism ϕ from Definition 1 is injective.*

If ϕ is not injective, we call an element $h \in \ker \phi$ of its kernel an ***algebraic relation*** *of the elements $f_1, \dots, f_m$.*

When working with polynomials over $k = \mathbb{Z}_q$ for q sufficiently large, the Schwartz-Zippel Lemma is a helpful tool to lower bound the probability that a fixed polynomial vanishes on a random point of $\mathbb{Z}_q^m$.

Lemma 1 (Schwartz-Zippel [40]). *Let $q \in \mathbb{N}$ be a prime and let $m, d \in \mathbb{N}$. Let $h \in k[Y]$ be a polynomial of degree d. Then, we can bound the probability of h vanishing on a random point of $\mathbb{Z}_q^m$ by*

$$\Pr_{y \leftarrow \mathbb{Z}_q^m}[h(y) = 0] \leq d/q. \tag{41}$$

2.3 Cryptographic Preliminaries

In this subsection, we will introduce the notion of pseudorandom number generators, and define a simple security game for them.

Definition 3 (Pseudorandom Number Generators). *Let $m : \mathbb{N} \to \mathbb{N}$ be a time-constructible function and let k be any field. A* ***pseudorandom number generator*** *(PRG) is a family of functions $F = (F_n)_{n \in \mathbb{N}}$ s.t. each F_n is a deterministic function*

$$F_n : k^n \longrightarrow k^m. \tag{42}$$

We call m the ***stretch*** *of the PRG. If there is a constant $e > 0$ s.t. $m \geq n^{1+e}$, we say that $(F_n)_{n \in \mathbb{N}}$ is a* ***poly-stretch*** *PRG.*

Remark 2. If $F = (F_n)_{n \in \mathbb{N}}$ is a PRG, we will, by abuse of notation, just write

$$F : k^n \to k^m. \tag{43}$$

For a given n, we will further write F when we actually mean F_n.

The adversaries in this text are always given a description of F_n (which we will simply denote by F) that allows the adversary to efficiently evaluate F_n on points of k^n. We assume that this description of F_n always contains binary representations of the numbers n, m and a description of the field k that allows the adversary to perform arithmetic operations over k. Additionally, if F is of locality or degree $d \in \mathbb{N}$ (in the sense of Definition 4), we expect the description of F to contain a binary representation of d.

Definition 4 (Locality and Degree of PRGs). *Let $F = (F_n)_n$ be a PRG of stretch m over k. Let $d \in \mathbb{N}$. For $n \in \mathbb{N}$ and $i \in [m]$, we denote by $f_{n,i} : k^n \to k$ the function of the i-th output of F_n. I.e., $f_{n,1}, \dots f_{n,m}$ are uniquely determined by*

$$F(x) = (f_{n,1}(x), \dots, f_{n,m}(x)) \tag{44}$$

for all $x \in k^n$.

1. *We say that F is d-**local** if each of its output values depends on only d input values. I.e. for each $n \in \mathbb{N}$ and $i \in [m]$ there is a function $g : k^d \to k$ and indices $l_1, \dots, l_d \in [n]$ s.t. we have for each $x \in k^n$*

$$f_{n,i}(x_1, \dots, x_n) = g(x_{l_1}, \dots, x_{l_d}).$$

2. *We say that F is of **degree** d if each $f_{n,i}$ can be computed by a polynomial of degree d. I.e., for each $n \in \mathbb{N}$ and $i \in [m]$ the function $f_{n,i} : k^n \to k$ coincides with a polynomial in $k[X]$ of degree $\leq d$. In this case, by abuse of notation, we will directly interpret $f_{n,i}$ as an element of $k[X]^{\leq d}$.*

For a given n, we will simply write $f_1, \dots, f_m$ instead of $f_{n,1}, \dots, f_{n,m}$ to denote the partial functions of F. We will usually say in those cases that F is made up of or consists of $f_1, \dots, f_m$.

Definition 5 (Security Game for PRGs). *Let k be finite now and let $F : k^n \to k^m$ be a PRG. We describe here a non-interactive security game between a probabilistic challenger $\mathcal{C}$ and a (potentially probabilistic) adversary $\mathcal{A}$. The game is parametrized by n and proceeds in the following steps:*

1. *$\mathcal{C}$ draws a bit $b \leftarrow \{0,1\}$. If $b = 0$, it samples a preimage $x \leftarrow k^n$ uniformly at random, computes $F(x)$ and sends $(F, F(x))$ to $\mathcal{A}$. If $b = 1$, it samples $y \leftarrow k^m$ and sends (F, y) to $\mathcal{A}$.*
2. *$\mathcal{A}$ receives (F, y^*) for some $y^* \in k^m$ and must decide which bit b has been drawn by $\mathcal{C}$. It makes some computations on its own without interacting with $\mathcal{C}$ and finally sends a bit b' to $\mathcal{C}$.*

$\mathcal{A}$ wins an instance of this game iff $b = b'$ holds at the end. We define $\mathcal{A}$'s advantage against F by

$$\begin{aligned} \mathsf{adv}_F(\mathcal{A}) &:= 2\Pr[\mathcal{A} \text{ wins}] - 1 && (45) \\ &= \Pr_{x \leftarrow k^n}[\mathcal{A}(F, F(x)) = 0] + \Pr_{y \leftarrow k^m}[\mathcal{A}(F, y) = 1] - 1 && (46) \end{aligned}$$

where we take the probability over the randomness of $\mathcal{A}$ and $\mathcal{C}$.

We define $\mathcal{A}$'s space complexity to be the number of bits and elements of k it stores simultaneously in step 2, and we define its time complexity by the number of bit-operations and arithmetical operations over k it performs in step 2.

Definition 6. *We say that an algorithm is **subexponential** if there is a constant $e \in [0,1)$ s.t. its time and space complexities lie in $2^{O(n^e)}$.*

Lemma 2 (Leftover Hash Lemma (Matrix Version) [21]). *Let $p \in \mathbb{N}$ be a prime and let $p, m, m', q \in \mathbb{N}$ be natural numbers, $q \geq 2$.*

If we draw $A_1, A_2 \leftarrow \mathbb{Z}_p^{m' \times m}, y_1 \leftarrow \{0, \dots, q-1\}^m, y_2 \leftarrow \mathbb{Z}_p^{m'}$, we have

$$\Delta((A_1, A_1 y_1), (A_2, y_2)) \leq \frac{1}{2}\sqrt{2^{m' \cdot \log p - m}}. \tag{47}$$

3 Finding Algebraic Relations

In this section, we introduce an algorithm $\mathcal{B}1$ that – given a set of polynomials – finds an algebraic relation among these polynomials. Further, we will prove upper bounds for the degree of this relation and for the complexity of the algorithm.

Now, let $n, m, d \in \mathbb{N}$ and let k be any field in this section. Let $F : k^n \to k^m$ be a polynomial mapping of degree $\leq d$ that is given by polynomials $f_1, \ldots, f_m \in k[X]$ of degree $\leq d$.

Denote by $\phi : k[Y] \to k[X]$ the dual morphism to F. Note, that ϕ expands the degrees of its inputs by a factor of at most d, i.e., we have for each $\ell \in \mathbb{N}_0$

$$\phi(k[Y]^{\leq \ell}) \subseteq k[X]^{\leq d \cdot \ell}. \tag{48}$$

Let $\ker \phi = \{h \in k[Y] \mid \phi(h) = 0\}$ be the kernel of ϕ. Our aim is to compute a non-trivial element of $\ker \phi$.

We will propose a straight-forward approach for this task: For $\ell = 1, 2, \ldots$, the algorithm $\mathcal{B}1$ will compute a monomial basis for $k[Y]^{\leq \ell}$ and check – by linear algebra – if the vector space $k[Y]^{\ell} \cap \ker \phi$ is non-trivial. If $k[Y]^{\ell} \cap \ker \phi$ contains a non-trivial element eventually, $\mathcal{B}1$ will output it and terminate. Formally, $\mathcal{B}1$ is given by:

Definition 7. *The algorithm $\mathcal{B}1$ gets as input numbers $n, m, d \in \mathbb{N}$, a description of k and a description of a polynomial map $F : k^n \to k^m$. It has to output a non-zero element of* $\ker \phi$*.*

$\mathcal{B}1$ sets an iteration variable $\ell := 1$ and proceeds in the following steps:

1. *$\mathcal{B}1$ computes $N := \binom{n+d\ell}{d\ell}$ and $M := \binom{m+\ell}{\ell}$*
2. *$\mathcal{B}1$ computes a finite list*

$$\begin{aligned} & (Y_1^{a_1} \cdots Y_m^{a_m} \mid a_1, \ldots, a_m \in \mathbb{N}_0, a_1 + \ldots + a_m \leq \ell) & (49) \\ = \; & (Y^{\alpha_1}, \ldots, Y^{\alpha_M}) & (50) \end{aligned}$$

   of all monomials in $k[Y]$ of degree $\leq \ell$.
3. *$\mathcal{B}1$ applies ϕ to each Y^{α_i} and computes a second list $(\phi(Y^{\alpha_1}), \ldots, \phi(Y^{\alpha_M}))$ of polynomials in $k[X]$ of degree $\leq d\ell$.*
4. *Let $X^{\beta_1}, \ldots, X^{\beta_N}$ be the set of all monomials in $k[X]$ of degree $\leq d\ell$. Then, $X^{\beta_1}, \ldots, X^{\beta_N}$ is a basis of $k[X]^{\leq d\ell}$ and for each $\phi(Y^{\alpha_i})$ there is a unique column vector $w_i = (w_{i,1}, \ldots, w_{i,N}) \in k^N$ s.t.*

$$\phi(Y^{\alpha_i}) = \sum_{j=1}^{N} w_{i,j} \cdot X^{\beta_j}. \tag{51}$$

   $\mathcal{B}1$ computes for each Y_{α_i} the corresponding vector w_i and writes down the matrix

$$W_\ell := \big(w_1 | \ldots | w_M\big) \in k^{N \times M}. \tag{52}$$

5. *$\mathcal{B}1$ uses Gaussian elimination to compute a basis for the vector space*

$$K_\ell := \left\{ r \in k^M \mid W_\ell \cdot r = 0 \right\}. \tag{53}$$

6. *If K_ℓ is the trivial null-space, $\mathcal{B}1$ increases ℓ by one and goes back to step 2.*
7. *Otherwise, $\mathcal{B}1$ chooses an arbitrary non-zero vector $r \in K_\ell$, computes the polynomial*

$$h := r_1 \cdot Y^{\alpha_1} + \ldots + r_M \cdot Y^{\alpha_M} \in k[Y] \tag{54}$$

of total degree $\leq \ell$ and outputs it.

We will show the following properties for $\mathcal{B}1$:

Lemma 3. *Let $n, m, d \in \mathbb{N}$ s.t. $m > n$. Let $F : k^n \to k^m$ be a polynomial map of degree $\leq d$. Let $y \in k^m$. We have the following:*

1. *On input n, m, d and F, $\mathcal{B}1$ will always terminate after a finite number of steps and output a polynomial h.*
2. *The polynomial h outputted by $\mathcal{B}1$ will always lie in $\ker \phi$ and be non-zero.*

Proof. 1. Note that $m > n$. The first claim of the lemma is equivalent to stating that m elements of $k[X]$ must be algebraically dependent and $\phi : k[Y] \to k[X]$ cannot be injective. This is a well-known fact in algebra and is easy to prove, however writing down a formally correct proof will make the notions of transcendency bases and function fields necessary. Therefore, we moved the proof of this statement to Appendix A.3.

2. Assume that $\mathcal{B}1$ stops after D iterations and outputs h. Then, h is a polynomial in $k[Y]$ of degree D and can be written as

$$h := r_1 \cdot Y^{\alpha_1} + \ldots + r_M \cdot Y^{\alpha_M} \in k[Y] \tag{55}$$

where $M = \binom{m+D}{D}$ and r is a non-zero kernel element of R_D. I.e., we have

$$\sum_{i=1}^{M} r_i \cdot w_i = 0. \tag{56}$$

Since the entries of w_i are exactly the coefficients of $\phi(Y^{\alpha_i})$, we have

$$\phi(h) = \phi\left(\sum_{i=1}^{M} r_i \cdot Y^{\alpha_i}\right) = \sum_{i=1}^{M} r_i \cdot \phi\left(Y^{\alpha_i}\right) = 0. \tag{57}$$

Ergo, $h \in \ker \phi$.

Lemma 4. *Assume that $\mathcal{B}1$ terminates after D iterations. Then, its space complexity can be bounded by $O(NM)$ and its time complexity can be bounded by $O(DN^2M)$ for $N = \binom{n+d \cdot D}{d \cdot D}$ and $M = \binom{m+D}{D}$.*

Proof. In each iteration step, $\mathcal{B}1$ computes a matrix of shape at most $N \times M$ over k. Therefore, the number of bits and elements of k it needs to store simultaneously can be bounded by $O(NM)$.

We can bound the time complexity of each iteration step from above by the time complexity of the D-th iteration step. In this step, $\mathcal{B}1$ performs Gaussian elimination on an $N \times M$-matrix which needs $O(N^2M)$ arithmetical operations over k. Therefore, the number of bit-operations and arithmetical operations $\mathcal{B}1$ needs to do in each step can be bounded by $O(N^2M)$, and $\mathcal{B}1$'s total time complexity can be bounded by $O(DN^2M)$.

Note, that $\mathcal{B}1$ starts at $\ell = 1$ and increases ℓ by one subsequently. Since $\mathcal{B}1$ terminates only if it finds a non-trivial element in $k[Y]^\ell \cap \ker\phi$, this means that the number D of iterations $\mathcal{B}1$ has to perform is exactly the lowest total degree of non-zero elements of $\ker\phi$.

Lemma 5. *$\mathcal{B}1$ terminates after D iterations iff $D = \min\{\deg h \mid h \in \ker\phi, h \neq 0\}$.*

3.1 Bounding *D*

We have seen in the last subsection that the time and space complexity of $\mathcal{B}1$ is substantially influenced by D. Since D is the minimal degree of a non-trivial element of $\ker\phi$, our aim in this subsection is to bound the degree of algebraic relations for all sets of polynomials $f_1, \ldots, f_m$ of degree $\leq d$.

Let ϕ_ℓ be the restriction of ϕ on $k[Y]^{\leq \ell}$. Then, each ϕ_ℓ is a linear map of type $k[Y]^{\leq \ell} \to k[X]^{\leq d\cdot\ell}$. We can guarantee that ϕ_ℓ has a non-trivial kernel, if the dimension of $k[Y]^{\leq \ell}$ exceeds the dimension of $k[X]^{\leq d\cdot\ell}$. Now, the dimensions of $k[Y]^{\leq \ell}$ and $k[X]^{\leq d\cdot\ell}$ are given by

$$\dim_k(k[Y]^{\leq \ell}) = \binom{m+\ell}{\ell} \quad \text{and} \quad \dim_k(k[X]^{\leq d\cdot\ell}) = \binom{n+d\cdot\ell}{d\cdot\ell}. \tag{58}$$

Therefore, we get for algorithm $\mathcal{B}1$:

Lemma 6. *Let D be the number of iterations of $\mathcal{B}1$. Then, we have*

$$D \leq \min\left\{\ell \in \mathbb{N} \mid \binom{m+\ell}{\ell} > \binom{n+d\cdot\ell}{d\cdot\ell}\right\}. \tag{59}$$

Inequality Eq. (59) gives us a tool to compute a worst-case bound for $\mathcal{B}1$'s complexity for each possible case of polynomials $f_1, \ldots, f_m$. In the next lemma, we will show that we can bound D by $O\left(\left(n^d/m\right)^{\frac{1}{d-1}}\right)$.

Lemma 7 (Main Inequality). *Let $d \in \mathbb{N}, d \geq 2$. Let $m : \mathbb{N} \to \mathbb{N}$ be a function with $m(n) \geq 2^{2d-1} \cdot d^{d-1} \cdot n$.*

Then, we have for all integers $n \geq 2d$

$$\binom{m(n)+L(n)}{L(n)} > \binom{n+dL(n)}{dL(n)} \tag{60}$$

where $L(n) = \left\lceil \left(\frac{(2n)^d}{m}\right)^{\frac{1}{d-1}} \right\rceil$.

Proof. Let $n \in \mathbb{N}$. In the proof, by abuse of notation, we write $m = m(n)$ and $L = L(n)$.

Note, that we have $2n \geq n + dL$, since

$$\begin{aligned} n \geq dL \iff{}& n \geq d \cdot \left\lceil \left(\frac{(2n)^d}{m} \right)^{\frac{1}{d-1}} \right\rceil && (61)\\ \Leftarrow \quad{}& n \geq d \cdot \left(\left(\frac{(2n)^d}{m} \right)^{\frac{1}{d-1}} + 1 \right) && (62)\\ \iff{}& n - d \geq d \cdot \left(\frac{(2n)^d}{m} \right)^{\frac{1}{d-1}} && (63)\\ \iff{}& (n-d)^{d-1} \geq d^{d-1} \cdot \frac{(2n)^d}{m} && (64)\\ \iff{}& m \geq d^{d-1} \cdot 2^d \cdot \left(\frac{n}{n-d} \right)^{d-1} \cdot n && (65)\\ \overset{n \geq 2d}{\Leftarrow} \quad{}& m \geq d^{d-1} \cdot 2^d \cdot 2^{d-1} \cdot n && (66) \end{aligned}$$

where the last inequality is required in the premise of the lemma.

Now, for the claimed inequality of the lemma, we have the following chain of equivalent inequalities

$$\begin{aligned} & \binom{m+L}{L} > \binom{n+dL}{dL} && (67)\\ \iff{}& \frac{(m+L)\cdots(m+1)}{L!} > \frac{(n+dL)\cdots(n+1)}{(dL)!} && (68)\\ \iff{}& (m+L)\cdots(m+1) \cdot (dL)\cdots(L+1) > (n+dL)\cdots(n+1) && (69) \end{aligned}$$

Note, that we have for all n the inequalities

$$\begin{aligned} (m+L)\cdots(m+1) &> m^L, && (70)\\ (dL)\cdots(L+1) &> L^{(d-1)L}. && (71) \end{aligned}$$

For the right-hand side, we have

$$(n+dL)\cdots(n+1) \leq (n+dL)^{dL} \leq (2n)^{dL} = 2^{dL} \cdot n^{dL}. \qquad (72)$$

By using the inequalities Eqs. (70) to (72), we see that Eq. (69) is implied by the inequality

$$m^L \cdot L^{(d-1)L} \geq 2^{dL} \cdot n^{dL}. \qquad (73)$$

By reducing Eq. (73) to the L-th root, we get the equivalent inequality

$$m \cdot L^{(d-1)} \geq 2^d \cdot n^d. \qquad (74)$$

Now, it is easy to show that this inequality holds:

$$m \cdot L^{(d-1)} \geq m \cdot \left(\frac{(2n)^d}{m}\right) = 2^d \cdot n^d. \tag{75}$$

This completes the proof.

In particular, for PRGs of poly-stretch $m \geq n^{1+e}$, for some constant $e > 0$, we get by Lemma 7

$$D = \left\lceil \left(\frac{(2n)^d}{m}\right)^{\frac{1}{d-1}} \right\rceil \leq \left\lceil \left(\frac{(2n)^d}{n^{1+e}}\right)^{\frac{1}{d-1}} \right\rceil = \left\lceil 2^{\frac{d}{d-1}} \cdot n^{\frac{d-1-e}{d-1}} \right\rceil \in O(n^{1-\frac{e}{d-1}}). \tag{76}$$

While $n^{1-\frac{e}{d-1}}$ is non-constant for $e < d-1$, it implies that we can bound the complexity of $\mathcal{B}1$ subexponentially by $n^{O(n^{1-\frac{e}{d-1}})}$.

The Lemmas 4 to 7 imply the following theorem:

Theorem 1. *Let $d \in \mathbb{N}$ be constant and $m \in \omega(n)$. Let $f_1, \ldots, f_m \in k[X]$ be polynomials of degree $\leq d$.*

Then, the algorithm $\mathcal{B}1$ in Definition 7 outputs a non-trivial element of $\ker \phi$ *of degree $O\left(\left(n^d/m\right)^{\frac{1}{d-1}}\right)$. Its space and time complexities lie in $n^{O((n^d/m)^{\frac{1}{d-1}})}$.*

If $m \geq n^{1+e}$ for a constant $e > 0$, the degree of the output of $\mathcal{B}1$ lies in $O\left(n^{1-\frac{e}{d-1}}\right)$ and $\mathcal{B}1$ is of subexponential complexity $n^{O(n^{1-\frac{e}{d-1}})}$.

4 Attacks on Constant-Degree PRGs over Large Moduli

In this section, we will focus on the case $k = \mathbb{Z}_q$ for a prime q that is sufficiently high (e.g. $q \in \Omega(n)$). We claim that in this case the algorithm $\mathcal{B}1$ from Definition 7 gives us a subexponential attack on each PRG of constant degree over $\mathbb{Z}_q$ and poly-stretch. In this section, we will prove:

Theorem 2. *Let $d \in \mathbb{N}$ be constant and $m \in \omega(n)$. Let $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ be a PRG of degree d over $\mathbb{Z}_q$.*

Then, there is an attack algorithm $\mathcal{A}1$ whose time and space complexities are bounded from above by $n^{O((n^d/m)^{\frac{1}{d-1}})}$. Further, there exists a constant $c > 0$ s.t. $\mathcal{A}1$'s advantage in the security game Definition 5 is lower bounded by

$$\mathsf{adv}_F(\mathcal{A}1) \geq 1 - c \cdot \left(n^d/m\right)^{\frac{1}{d-1}} \cdot \frac{1}{q}. \tag{77}$$

If $m \geq n^{1+e}$ for some constant $e > 0$, then $\mathcal{A}1$'s complexities are in $n^{O(n^{1-\frac{e}{d-1}})}$ and its advantage is at least $1 - c \cdot n^{1-\frac{e}{d-1}}/q$.

The attack $\mathcal{A}1$ on F is defined as follows:

Definition 8. *$\mathcal{A}1$ receives as input a description of F that includes the numbers $n, m, q, d \in \mathbb{N}$ and an element $y^* \in \mathbb{Z}_q^m$. The goal of $\mathcal{A}1$ is to output 0, if y^* lies in the image of F, and 1, otherwise.*

$\mathcal{A}1$ proceeds in two simple steps:

1. *$\mathcal{A}1$ executes the algorithm $\mathcal{B}1$ from Definition 7 on the input n, m, d, q, F and receives a non-zero polynomial $h \in \mathbb{Z}_q[Y]$ as output.*
2. *$\mathcal{A}1$ outputs 0 if $h(y^*) = 0$. Otherwise, $\mathcal{A}1$ outputs 1.*

The bound on the time and space complexities of $\mathcal{A}1$ follows now from Theorem 1. The advantage of $\mathcal{A}1$ can be bounded as follows:

If $b = 0$ in the security game of Definition 5, then the challenger $\mathcal{C}$ samples $x \leftarrow \mathbb{Z}_q^n$ and gives the pseudorandom image $y^* = F(x)$ to $\mathcal{A}1$. The polynomial h outputted by $\mathcal{B}1(F)$ lies in the kernel of ϕ, i.e., we have the equality $h(F(X)) = 0$ of polynomials in $\mathbb{Z}_q[X]$. In particular, we have $h(F(x)) = 0$ for each $x \in \mathbb{Z}_q^n$. Therefore, $\mathcal{A}1$ always outputs 0 if $b = 0$.

If $b = 1$ in the security game in Definition 5, then the challenger $\mathcal{C}$ samples a uniformly random $y \leftarrow \mathbb{Z}_q^m$ and gives $y^* = y$ to $\mathcal{A}1$. Since h is non-zero and of degree $O((n^d/m)^{\frac{1}{d-1}})$, the probability that h vanishes on y can be bounded by

$$\Pr_{y \leftarrow \mathbb{Z}_q^n}[h(y) = 0] \leq O((n^d/m)^{\frac{1}{d-1}})/q \tag{78}$$

according to Lemma 1. Therefore, $\mathcal{A}1$ will output 1 in this case with probability at least $1 - O((n^d/m)^{\frac{1}{d-1}})/q$.

For the overall advantage of $\mathcal{A}1$, we get

$$\begin{aligned} \mathsf{adv}_F(\mathcal{A}1) = &\Pr_{x \leftarrow \mathbb{Z}_q^n}[\mathcal{A}1(F, F_n(x)) = 0] + \Pr_{y \leftarrow \mathbb{Z}_q^m}[\mathcal{A}1(F, y) = 0] - 1 && (79)\\ \geq &1 + 1 - O((n^d/m)^{\frac{1}{d-1}})/q - 1 = 1 - O((n^d/m)^{\frac{1}{d-1}})/q. && (80) \end{aligned}$$

Remark 3. Algorithm $\mathcal{A}1$ proceeds in two steps: in its first step, it uses $\mathcal{B}1$ to compute an algebraic relation h of F, and in its second step, it uses h to decide if the given image $y^* \in \mathbb{Z}_q^m$ is truly random.

However, since the PRG F is fixed and publicly known, the attack $\mathcal{A}1$ can be interpreted as an attack with preprocessing: In a first phase, the so-called *preprocessing* or *offline* phase, $\mathcal{A}1$ uses $\mathcal{B}1$ to compute an algebraic relation h of F of degree D (without seeing the value $y^* \in \mathbb{Z}_q^m$).

In a second phase, the so-called *online* phase, $\mathcal{A}1$ receives $y^* \in \mathbb{Z}_q^m$ and only needs to evaluate h on y^*.

If $m \geq n^{1+e}$, then the degree of h is bounded by $D \leq cn^{1-\frac{e}{d-1}}$ for some constant c. The evaluation of h requires $(D+1) \cdot \binom{m+D}{D}$ arithmetic operations over $\mathbb{Z}_q$ which will be much less than the time $\mathcal{B}1$ needs (since $\mathcal{B}1$ needs to reduce a matrix of shape $\binom{m+D}{D} \times \binom{n+d \cdot D}{d \cdot D}$).

Therefore, from a practical point of view, it makes more sense to interpret $\mathcal{A}1$ as an attack with preprocessing, where we invest a big one-time cost to find a relation h of F in the preprocessing phase, and then a smaller, but still subexponential, cost of $(D+1) \cdot \binom{m+D}{D}$ to decide challenges of F.

5 Attacks on Constant-Locality PRGs over Constant Moduli

In this section, we will focus on the case where the modulus q is *constant* and assume additionally that $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ is a PRG of constant *locality* d (the case where F may be of arbitrary locality over small modulus is handled in Appendix A.1).

Theorem 3. *Let $d, q \in \mathbb{N}$ be constants where q is a prime. Let $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ be a PRG of locality d with stretch $m \in \omega(n)$.*

There is an attack $\mathcal{A}2$ on F and a constant $c > 0$ s.t. $\mathcal{A}2$'s space and time complexities are bounded by $n^{O((n^{d(q-1)}\log(n)/m)^{\frac{1}{d(q-1)-1}})}$ and whose advantage in the security game of Definition 5 is at least

$$\mathsf{adv}_F(\mathcal{A}2) \geq 1 - c \cdot (n\log(n)/m)^{\frac{1}{d(q-1)-1}}. \tag{81}$$

If $q = 2$ and $m \geq n^{1+e}$ for some constant $e > 0$, then the complexities of $\mathcal{A}2$ lie in $n^{O(n^{1-\frac{e}{d-1}} \cdot \log(n)^{\frac{1}{d-1}})}$ and its advantage is at least $1 - c \cdot (\log(n)/n^e)^{\frac{1}{d-1}}$.

The idea of $\mathcal{A}2$ is to convert F to a PRG $G : \mathbb{Z}_p^n \to \mathbb{Z}_p^{m'}$ of degree $d(q-1)$ over $\mathbb{Z}_p$ with stretch $m' = \lfloor m/(3\log(p)) \rfloor$ for a prime $p \geq n$.

Let $f_1, \ldots, f_m \in \mathbb{Z}_q[X]$ be the polynomials that make up F. Since each f_i is d-local, there are polynomials $f'_1, \ldots, f'_m \in \mathbb{Z}_p[X]$ of degree $\leq d(q-1)$ that coincide with $f_1, \ldots, f_m$ on $\{0, \ldots, q-1\}^n$. In fact, for $i \in [m]$, let $j_1, \ldots, j_d \in [n]$ and $u_i : \{0, \ldots, q-1\}^d \to \{0, \ldots, q-1\}$ be s.t. for all $x \in \{0,1\}^n$

$$f_i(x) = u_i(x_{j_1}, \ldots, x_{j_d}). \tag{82}$$

For $z \in \{0, \ldots, q-1\}$, let $s_z(Z) \in \mathbb{Z}_p[Z]$ be the univariate polynomial of degree $q-1$ with $s_z(z) = 1$ and $s_z(z') = 0$ for $z' \in \{0, \ldots, q-1\} \setminus \{z\}$ (for each z, there is exactly one polynomial s_z of degree $q-1$ with these properties). Then, the polynomial $f'_i \in \mathbb{Z}_p[X]$ is given by

$$f'_i(X) := \sum_{z \in \{0,\ldots,q-1\}^d} u_i(z) \cdot s_{z_1}(X_{j_1}) \cdots s_{z_d}(X_{j_d}) \tag{83}$$

and its degree is $\deg s_{z_1} + \ldots + \deg s_{z_d} = d(q-1)$.

However, the image of the $f'_1, \ldots, f'_m$ does not look random over $\mathbb{Z}_q$, since it is contained in $\{0, \ldots, q-1\}^m$ (if the input is chosen from $\{0, \ldots, q-1\}^n$). To compensate for that, we use the Leftover Hash Lemma. Let $F' = (f'_1, \ldots, f'_m) : \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ be the collection of all f'_i. $\mathcal{A}2$ samples now a random matrix $A = (a_{i,j})_{i,j} \leftarrow \mathbb{Z}_p^{m' \times m}$ and defines a PRG $G : \mathbb{Z}_p^n \to \mathbb{Z}_p^{m'}$ by

$$G(X) := A \cdot F'(X). \tag{84}$$

I.e., if G consists of the polynomials $g_1, \dots, g_{m'}$, each g_i is given by

$$g_i = \sum_{j=1}^{m} a_{i,j} \cdot f'_j. \tag{85}$$

Now, G is a PRG over $\mathbb{Z}_p$ of degree $d(q-1)$. According to Lemma 2, the image of G will *look random* (relative to $\mathbb{Z}_p^{m'}$) if the image of F looks random (relative to $\{0, \dots, q-1\}^m$). Finally, $\mathcal{A}2$ can use $\mathcal{A}1$ from Theorem 2 to break the pseudorandomness of G (and break therefore the pseudorandomness of F).

We will now formally define how $\mathcal{A}2$ proceeds:

Definition 9. *Let $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ be a PRG of locality d over constant modulus q consisting of polynomials $f_1, \dots, f_m \in \mathbb{Z}_q[X]$. The algorithm $\mathcal{A}2$ receives as input a description of F, which includes the numbers $n, m, d, q \in \mathbb{N}$, and an element $y^* \in \mathbb{Z}_q^m$. The goal of $\mathcal{A}2$ is to output 0, if y^* lies in the image of F, and 1, otherwise.*

$\mathcal{A}2$ proceeds in the following steps:

1. *$\mathcal{A}2$ searches for a prime number $p \in \{n, n+1, \dots, 2n\}$. Because of Bertrand's postulate we know that such a prime must exist.*
2. *$\mathcal{A}2$ sets $m' := \lfloor m/(3 \log p) \rfloor$*
3. *$\mathcal{A}2$ computes polynomials $f'_1, \dots, f'_m \in \mathbb{Z}_p[X]$ of degree $d(q-1)$ that coincide with $f_1, \dots, f_m$ on $\{0, \dots, q-1\}^n$.*
4. *$\mathcal{A}2$ draws a random matrix $A \leftarrow \mathbb{Z}_p^{m' \times m}$ and sets*

$$G(X) := A \cdot F'(X). \tag{86}$$

   Now, $G : \mathbb{Z}_p^n \to \mathbb{Z}_p^{m'}$ is a polynomial map of degree $d(q-1)$.

5. *$\mathcal{A}2$ interprets $y^* \in \mathbb{Z}_q^m$ as a vector in $\{0, \dots, q-1\}^m \subseteq \mathbb{Z}_p^m$ and computes*

$$y'^* := A \cdot y^* \in \mathbb{Z}_p^m. \tag{87}$$

6. *$\mathcal{A}2$ runs algorithm $\mathcal{A}1$ on (G, y'^*) and returns the output of $\mathcal{A}1$.*

It is easy to see that the time and space complexities of $\mathcal{A}2$ are dominated by the complexities of $\mathcal{A}1$, which are upper-bounded by

$$n^{O((n^{d(q-1)}/m')^{\frac{1}{d(q-1)-1}})} = n^{O((n^{d(q-1)} \log(n)/m)^{\frac{1}{d(q-1)-1}})}. \tag{88}$$

To bound the advantage of $\mathcal{A}2$, we first distinguish two cases:

1. If $y^* = F(x)$ for some $x \in \mathbb{Z}_q^n$, then y'^* will be of the form

$$y'^* = Ay^* = AF'(x) = G(x). \tag{89}$$

   In those cases, $\mathcal{A}1$ will always output zero.

2. If y^* is a random element of $\mathbb{Z}_q^m$, then Lemma 2 states that the statistical distance of the distributions

$$(A, {y'}^*) \text{ and } (A, r) \tag{90}$$

for $r \leftarrow \mathbb{Z}_p^{m'}$ is less than $\frac{1}{2}\sqrt{2^{m' \log(p) - m}} \leq \frac{1}{2} p^{-m'}$. Therefore, the probability that $\mathcal{A}1$ outputs 1 in this case can be lower bounded by

$$\begin{aligned} & 1 - \frac{O((n^{d(q-1)} \log(n)/m)^{\frac{1}{d(q-1)-1}})}{q} - \frac{1}{2} p^{-m'} && (91) \\ \geq & 1 - \frac{O((n^{d(q-1)} \log(n)/m)^{\frac{1}{d(q-1)-1}})}{n} - \frac{1}{2} n^{-m'} && (92) \\ \geq & 1 - O((n \log(n)/m)^{\frac{1}{d(q-1)-1}}). && (93) \end{aligned}$$

Now, we can bound the advantage of $\mathcal{A}2$ in the security-game of Definition 5 as follows:

$$\begin{aligned} \mathsf{adv}_F(\mathcal{A}2) \geq & \Pr_{y \leftarrow \{0,1\}^m}[\mathcal{A}2(F, y) = 1] + \Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}2(F, F(x)) = 0] - 1 && (94) \\ \geq & 1 - O((n \log(n)/m)^{\frac{1}{d(q-1)-1}}) + 1 - 1 && (95) \\ \geq & 1 - O((n \log(n)/m)^{\frac{1}{d(q-1)-1}}). && (96) \end{aligned}$$

6 Comparison of our Algorithms with Groebner Basis Algorithms

In this section, we will give an in-depth comparison between our algorithms and Groebner basis-based algorithms. We will discuss parallels and differences, talk about the degree of regularity (a popular heuristic for Groebner basis-based algorithms) and compare the time and space complexities of Groebner basis-based and our algorithms.

Groebner Basis-Based Algorithms. The literature contains a huge variety of algebraic algorithms that solve polynomial equation systems by computing Groebner bases. The most famous ones are F4 [24] and F5 [23] and the XL algorithms [15, 16, 19, 36, 44]. All these algorithms try – when given a list of polynomials as input – to compute a Groebner basis of the ideal generated by the given polynomials (in case of the XL algorithms, it does not need to be a Groebner basis but something equivalent [6, 42]).

We will first describe a prototype Groebner basis algorithm [34] that is based on Macaulay matrices [35] and underlies all modern algorithms for computing Groebner bases:

Definition 10. *Let k be any field.*

The algorithm $\mathcal{G}1$ gets as input numbers $n, m, d \in \mathbb{N}$, polynomials $f_1, \ldots, f_m \in k[X] = k[X_1, \ldots, X_n]$ and a description of an ordering $\leq$ of the monomials of $k[X]$ s.t. we have for all monomials $X^\alpha, X^\beta, X^\gamma$:

$$1 \leq X^\alpha, \tag{97}$$

$$X^\alpha \leq X^\beta \implies X^\alpha \cdot X^\gamma \leq X^\beta \cdot X^\gamma. \tag{98}$$

$\mathcal{G}1$ proceeds as follows for $\ell = 1, 2, \ldots$:

1. *$\mathcal{G}1$ computes the set*

$$S_\ell := \bigcup_{i=1}^{m} \left\{ X_1^{a_1} \cdots X_n^{a_n} \cdot f_i \mid a_1, \ldots, a_n \in \mathbb{N}_0, \deg(f_i) + a_1 + \ldots + a_n \leq \ell \right\}. \tag{99}$$

2. *Set $N := \binom{n+\ell}{\ell}$ and $M := \#S_\ell$. $\mathcal{G}1$ writes down an $M \times N$ matrix W_ℓ where each column represents a monomial of degree $\leq \ell$ and each row represents an element of S_ℓ. The columns are ordered according to $\leq$ s.t. the biggest monomial is on the left end of the matrix while 1 is on the right end.*
   The i, j-th entry of W_ℓ is given by the coefficient of the i-th element of S_ℓ regarding the j-th monomial.
3. *$\mathcal{G}1$ applies Gaussian elimination to W_ℓ and receives a new matrix W'_ℓ.*
4. *$\mathcal{G}1$ checks if the rows of W'_ℓ form a Groebner basis. If so, $\mathcal{G}1$ outputs the polynomials corresponding to the rows of W'_ℓ. Otherwise, $\mathcal{G}1$ increments ℓ and goes back to step 1.*

Usually, Groebner basis algorithms are used to compute solutions of polynomial equation systems. Indeed, if the system $f_1(X) = 0, \ldots, f_m(X) = 0$ only has one solution $x \in \overline{k}^n$ over the algebraic closure $\overline{k} \supset k$, then any Groebner basis of $(f_1, \ldots, f_m)$ must contain the equations $X_1 - x_1, \ldots, X_n - x_n$ (up to scalar multiples).

If we want to use $\mathcal{G}1$ to decide if a point $y \in k^m$ lies in the image of a map $F : k^n \to k^m$ made up of polynomials $f_1, \ldots, f_m$, we invoke $\mathcal{G}1$ on the polynomials $f_1(X) - y_1, \ldots, f_m(X) - y_m$. If the Groebner basis returned by $\mathcal{G}1$ contains a unit $k^\times$, then the polynomial equation system $F(X) = y$ is unsatisfiable. Otherwise, there exist a solution x s.t. $F(x) = y$, however x may lie in the algebraic closure $\overline{k}^n$ (this problem can be fixed by adding the field equations of k to the list of polynomials given to $\mathcal{G}1$).

We can now see that $\mathcal{G}1$ and the algorithm $\mathcal{B}1$ have similar strategies: both compute elements of increasing degree of specific ideals. However, $\mathcal{G}1$ computes elements of the ideal generated by the $f_1, \ldots, f_m$ and stops only if it finds a Groebner basis. On the other side, $\mathcal{B}1$ computes the kernel of $\phi : k[Y] \to k[X]$ and stops if it finds any non-zero element of $\ker \phi$.

Degree of Regularity. It is complicated to bound the space and time complexities of $\mathcal{G}1$ and its relatives. Ideally, we would like to know the maximum iteration number D_{solv} s.t. the span of $S_{D_{\mathrm{solv}}}$ contains a Groebner basis. This number is known as the **solving degree** of the system $f_1, \ldots, f_m$. It depends heavily on the elements $f_1, \ldots, f_m$ and is hard to be computed exactly.

Therefore, there exists a variety of different degrees that are used to estimate the solving degree: the degree of regularity [10], the first fall degree [20, 22], the last fall degree [28] and the Castelnuvo-Mumford regularity [12]. We will only focus on the degree of regularity here and refer the reader to [12, 13] for excellent surveys on the different degrees.

The degree of regularity is the most popular heuristic to estimate the complexities of Groebner basis algorithms. It is neither a lower nor an upper bound for the solving degree, however in special cases where all $f_1, \ldots, f_m$ are semiregular or generic [9, 18] the degree of regularity is indeed an upper bound.

To formally define the degree of regularity, we will introduce here the concept of Hilbert series: Consider the polynomial ring $k[X] = k[X_1, \ldots, X_n]$. It is graded where we assign to each X_i the degree 1. The dimension of the i-th grade of $k[X]$ is given by $k[X]^i = \binom{n+i-1}{i}$. We define the **Hilbert series** $h_{k[X]}(T) \in \mathbb{Z}[\![T]\!]$ of $k[X]$ as the formal power series in the variable T where the i-th coefficient is the dimension of $k[X]^i$. I.e.

$$h_{k[X]}(T) := \sum_{i=0}^{\infty} \dim_k(k[X]^i) \cdot T^i = \sum_{i=0}^{\infty} \binom{n+i-1}{i} \cdot T^i. \tag{100}$$

Note, that we have

$$h_{k[X]}(T) = \left(\sum_{i=0}^{\infty} T^i \right)^n = \frac{1}{(1-T)^n}. \tag{101}$$

Now, consider the polynomial ring $k[Y] = k[Y_1, \ldots, Y_m]$ where each variable Y_i gets assigned the degree d. $k[Y]$ is graded, and we have for $\dim_k(k[Y]^i)$

$$\dim_k(k[Y]^i) = \begin{cases} \binom{m + \frac{i}{d}}{m}, & \text{if } i \bmod d = 0. \\ 0, & \text{otherwise.} \end{cases} \tag{102}$$

The Hilbert series of $k[Y]$ is given by

$$h_{k[Y]}(T) := \sum_{i=0}^{\infty} \dim_k(k[Y]^i) \cdot T^i = \sum_{i=0}^{\infty} \binom{n+i-1}{i} \cdot T^{di}. \tag{103}$$

Similarly to $h_{k[X]}$, we have

$$h_{k[Y]}(T) = \left(\sum_{i=0}^{\infty} T^{di} \right)^m = \frac{1}{(1-T^d)^m}. \tag{104}$$

The **degree of regularity** for the parameters n, m, d is now defined as the smallest number $D_{\mathrm{reg}} \in \mathbb{N}_0$ s.t. the D_{reg}-th coefficient of the formal power series

$$h_1(T) := \frac{h_{k[X]}(T)}{h_{k[Y]}(T)} = \frac{(1-T^d)^m}{(1-T)^n} = (1 + T + \ldots + T^{d-1})^n \cdot (1-T^d)^{m-n} \tag{105}$$

is less or equal to zero.

Now, consider the following Hilbert series

$$h_2(T) := h_{k[X]}(T) - h_{k[Y]}(T) \tag{106}$$

$$= \sum_{i=0}^{\infty} \binom{n-1+i}{n-1} T^i - \sum_{i=0}^{\infty} \binom{m-1+i}{m-1} \cdot T^{di}. \tag{107}$$

If we are given polynomials $f_1, \ldots, f_{m-1}$ in $n-1$ variables $X_1, \ldots, X_{n-1}$ of degree d, then the coefficients $\binom{n-1+di}{n-1}$ are the dimensions of $k[X_1, \ldots, X_{n-1}]^{\leq di}$, the coefficients $\binom{m-1+i}{m-1}$ are the dimensions of $k[Y_1, \ldots, Y_{n-1}]^{\leq i}$ and the differences $\binom{m-1+i}{m-1} - \binom{n-1+di}{n-1}$ lower bound the dimensions of $(\ker \phi^{di})$ for $\phi : k[Y_1, \ldots, Y_{m-1}]^{\leq i} \to k[X_1, \ldots, X_{n-1}]^{\leq di}$. This means $\ker \phi$ will contain an element of degree i if the i-th coefficient of $h_2(T)$ is negative. Therefore, we define the **degree of kernel** for the parameters $n-1, m-1, d$ as the smallest number $D_{\text{ker}} \in \mathbb{N}_0$ s.t. the D_{ker}-th coefficient of $h_2(T)$ is negative.

Note, that the degree of the smallest non-trivial element of $\ker \phi$ bounds exponentially the complexities of the algorithms $\mathcal{B}1$ from Definition 7 and subsequently the complexities of the attacks $\mathcal{A}1$ and $\mathcal{A}2$. Therefore, the degree of kernel tells us something about the complexity of the algebraic attacks in this paper, while the degree of regularity gives us insight about the complexities of Groebner basis-based approaches. Both numbers come from similar Hilbert series, and we can relate them as follows:

Theorem 4. *Let D_{reg} be the degree of regularity for the parameters n, m, d and D_{ker} be the degree of kernel for $n-1, m-1, d$. Then, we have*

$$D_{\text{reg}} \leq D_{\text{ker}}. \tag{108}$$

Proof. Note, that we have the relationship

$$h_1(T) = (1 - T^d)^m \cdot h_2(T) + 1 \tag{109}$$

for the series h_1 and h_2. It therefore suffices to show that multiplication with $(1 - T^d)^m$ do not increase the position of the first negative coefficient.

For this sake, we define in this proof for a Hilbert series $h(T) = \sum_{i=0}^{\infty} c_i \cdot T^i$

$$\operatorname{ind}(h) := \min(\{i \in \mathbb{N}_0 \mid c_i < 0\} \cup \{\infty\}). \tag{110}$$

We claim that we have for each Hilbert series h

$$\operatorname{ind}((1 - T^d) \cdot h) \leq \operatorname{ind}(h). \tag{111}$$

In fact, we can compute

$$\begin{aligned}(1-T^d)\cdot h(T) &= (1-T^d)\cdot(\sum_{i=0}^{\infty} c_i \cdot T^i) && (112)\\ &= \sum_{i=0}^{\infty} c_i \cdot T^i - T^d \cdot \sum_{i=0}^{\infty} c_i \cdot T^i && (113)\\ &= \sum_{i=0}^{\infty}(c_i - c_{i-d}) \cdot T^i && (114)\end{aligned}$$

where we set $c_i = 0$ for $i < 0$. Now, we have that $c_{\mathrm{ind}(h)} - c_{\mathrm{ind}(h)-d}$ must be negative, since $c_{\mathrm{ind}(h)}$ is negative and each coefficient before $c_{\mathrm{ind}(h)}$ must be greater or equal to zero.

For the series h_1, h_2 it follows now

$$\mathrm{ind}(h_1(T) - 1) = \mathrm{ind}((1-T^d)^m \cdot h_2(T)) \leq \mathrm{ind}(h_2(T)) = D_{\mathrm{ker}}. \qquad (115)$$

Since the degree of regularity is the first coefficient of h_1 that is non-positive, it follows

$$D_{\mathrm{reg}} \leq \mathrm{ind}(h_1(T) - 1) \leq D_{\mathrm{ker}}. \qquad (116)$$

Combined with Lemma 7, we get immediately the following corollary for the degree of regularity:

Corollary 1. *Let $n, m, d \in \mathbb{N}$ s.t. $n - 1 \geq 2d$ and $m - 1 \geq 2^{2d-1} \cdot d^{d-1} \cdot (n-1)$. Then, we can bound the degree of regularity for the parameters n, m, d by*

$$D_{\mathrm{reg}} \leq \left\lceil \left(\frac{(2n-2)^d}{m-1}\right)^{\frac{1}{d-1}} \right\rceil. \qquad (117)$$

In particular, if $m - 1 \geq (n-1)^{1+e}$, we have for sufficiently large n

$$D_{\mathrm{reg}} \leq \left\lceil 2^{\frac{d}{d-1}} \cdot (n-1)^{1-\frac{e}{d-1}} \right\rceil. \qquad (118)$$

One can suspect that this upper bound is – on a very coarse asymptotic level – tight. Applebaum showed [5] that each algebraic refutation attack on a PRG of constant locality that stems from a predicate of high rational degree must have a complexity of at least $\Omega(n^{1-32\frac{e}{d-2}})$. This insinuates that the degree of regularity is lower-bounded by $\Omega(n^{1-32\frac{e}{d-2}})$ and the above upper bound is optimal up to some constants in the exponent.

A Groebner Basis-Based Distinguishing Algorithm. Theorem 4 insinuates that Groebner basis-based algorithms should not have a higher complexity than the algorithms presented in this text. We want to investigate this further:

Let $F : k^n \to k^m$ be a polynomial map of degree d and let $\phi : k[Y] \to k[X]$ be its dual morphism. Let $h \in \ker \phi$ be non-zero of degree D and let $y \in k^m$.

If we run the algorithm $\mathcal{G}1$ with the input $f_1(X)-y_1, \ldots, f_m(X)-y_m$, it will in its $d \cdot D$-th step produce a set $S_{d \cdot D}$ whose linear span contains the polynomial

$$h(f_1(X), \ldots, f_m(X)) - h(y_1, \ldots, y_m) = h(y_1, \ldots, y_m). \tag{119}$$

If y does not lie in the image of F, then – with high probability – $h(y)$ will be non-zero and $\mathcal{G}1$ has found a contradiction (and will output a Groebner basis that contains a unit of k). On the other side, if y does lie in the image of F, then the ideal generated by $f_1(X) - y_1, \ldots, f_m(X) - y_m$ is proper and the set S_ℓ will never contain units of k in its span. This observation gives rise to the following Macaulay matrix-based distinguishing attack on PRGs:

Definition 11. *Let k be any field.*

The algorithm $\mathcal{G}2$ gets as input numbers $n, m, d \in \mathbb{N}$, a polynomial map $F : k^n \to k^m$ and a point $y \in k^m$. Its aim is to output 0, if $y \in F(k^n)$, and 1 otherwise.

As a first step, $\mathcal{G}$ computes the smallest number D s.t.

$$\binom{m+D}{m} > \binom{n+d \cdot D}{n}. \tag{120}$$

Then, for $\ell = 1, \ldots, d \cdot D$, it proceeds as follows:

1. *$\mathcal{G}2$ computes the set*

$$S_\ell := \bigcup_{i=1}^{m} \left\{ X_1^{a_1} \cdots X_n^{a_n} \cdot f_i \mid a_1, \ldots, a_n \in \mathbb{N}_0, \deg(f_i) + a_1 + \ldots + a_n \leq \ell \right\}. \tag{121}$$

2. *$\mathcal{G}2$ checks if $\operatorname{span}_k(S_\ell)$ contains 1. If $1 \in \operatorname{span}_k(S_\ell)$, $\mathcal{G}2$ outputs 1.*
3. *Otherwise, $\mathcal{G}2$ increments ℓ by one and goes back to the first step of this list.*

If at the end, $\mathcal{G}2$ didn't output 1 in all of its $d \cdot D$ iterations, $\mathcal{G}2$ outputs 0.

The following properties can be shown for $\mathcal{G}2$:

Lemma 8. *1. If y lies in the image of F, then $\mathcal{G}2(F, y)$ will always output 0.*
2. If $\mathcal{A}1(F, y)$ outputs 1, then $\mathcal{G}2(F, y)$ will output 1, too.

I.e., $\mathcal{G}2$ is at least as correct as $\mathcal{A}1$.

Which Algorithm is Faster? Finally, we want to compare the performance of $\mathcal{G}2$ and $\mathcal{B}1/\mathcal{A}1$ when distinguishing PRGs $F : k^n \to k^m$ of polynomial degree d and stretch $m \geq n^{1+e}$.

The complexity of $\mathcal{G}2$ is dominated by checking $1 \in \operatorname{span}_k(S_\ell)$ in each iteration. To do this, $\mathcal{G}2$ must apply Gaussian reduction to a matrix with $\leq m\binom{n+\ell}{n}+1$

rows and $\binom{n+\ell}{n}$ columns for $\ell = 1, \ldots, dD$. In comparison, $\mathcal{B}1$ must apply Gaussian reduction to a matrix of shape $\binom{m+\ell}{m} \times \binom{n+d\ell}{n}$ for $\ell = 1, \ldots, D$. By ignoring polynomial factors and assuming that $\binom{m+D}{m}$ will be approximately as big as $\binom{n+dD}{n}$, we see that both algorithms must apply Gaussian reduction to a matrix of shape approximately $N \times N$ in their last step where $N = \binom{n+dD}{dD}$. However, in the case of $\mathcal{G}2$, this matrix is sparse and has only $\binom{n+d}{d}$ non-zero entries per row. Since sparse matrices admit faster Gaussian reductions [15], the computational cost of $\mathcal{G}2$ in each step can be upper-bounded by $O(\binom{n+d}{d}N^2)$, while the computational cost of $\mathcal{B}1$ lies in $O(N^3)$ per step.

However, note that $\mathcal{B}1$ does not need to know y in advance. In fact, if the PRG $F : k^n \to k^m$ is fixed and publicly known, one can use $\mathcal{B}1$ in a costly preprocessing phase to compute a polynomial $h \in k[Y]$ of degree D. The polynomial h has at most $\binom{m+D}{D} \approx N$ coefficients and can be evaluated in time $D \cdot N$.

We can now answer the question, which algorithm is faster, as follows:

1. In situations where F is not previously known, $\mathcal{G}2$ gives the faster distinguishing attack and has a complexity of approximately

$$\operatorname{poly}(n) \cdot \binom{n+dD}{dD}^2 \tag{122}$$

for $D = \left\lceil 2^{\frac{d}{d-1}} \cdot n^{1-\frac{e}{d-1}} \right\rceil$.

2. In situations where F is previously known, it makes more sense to use $\mathcal{B}1$ in a preprocessing phase and invest a one-time cost of approximately

$$\operatorname{poly}(n) \cdot \binom{n+dD}{dD}^3 \tag{123}$$

field operations to obtain a non-zero polynomial $h \in \ker \phi$ of degree $\leq D$. Evaluating h in the online phase takes approximately

$$D \cdot \binom{n+dD}{dD} \tag{124}$$

field operations.

References

1. Applebaum, B. *Pseudorandom generators with long stretch and low locality from random local one-way functions* in *44th ACM STOC* (eds Karloff, H. J. & Pitassi, T.) (ACM Press, May 2012), 805–816.
2. Applebaum, B. *Cryptographic Hardness of Random Local Functions-Survey* in *TCC 2013* (ed Sahai, A.) **7785** (Springer, Heidelberg, Mar. 2013), 599.
3. Applebaum, B. *The Cryptographic Hardness of Random Local Functions – Survey* Cryptology ePrint Archive, Report 2015/165. `https://eprint.iacr.org/2015/165`. 2015.

4. Applebaum, B., Damgård, I., Ishai, Y., Nielsen, M. & Zichron, L. *Secure Arithmetic Computation with Constant Computational Overhead* in *CRYPTO 2017, Part I* (eds Katz, J. & Shacham, H.) **10401** (Springer, Heidelberg, Aug. 2017), 223–254.
5. Applebaum, B. & Lovett, S. *Algebraic attacks against random local functions and their countermeasures* in *48th ACM STOC* (eds Wichs, D. & Mansour, Y.) (ACM Press, June 2016), 1087–1100.
6. Ars, G., Faugère, J.-C., Imai, H., Kawazoe, M. & Sugita, M. *Comparison Between XL and Gröbner Basis Algorithms* in *Advances in Cryptology - ASIACRYPT 2004* (ed Lee, P. J.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2004), 338–353. ISBN: 978-3-540-30539-2.
7. Barak, B., Brakerski, Z., Komargodski, I. & Kothari, P. K. *Limits on Low-Degree Pseudorandom Generators (Or: Sum-of-Squares Meets Program Obfuscation)* in *EUROCRYPT 2018, Part II* (eds Nielsen, J. B. & Rijmen, V.) **10821** (Springer, Heidelberg, 2018), 649–679.
8. Barak, B., Hopkins, S. B., Jain, A., Kothari, P. & Sahai, A. *Sum-of-Squares Meets Program Obfuscation, Revisited* in *EUROCRYPT 2019, Part I* (eds Ishai, Y. & Rijmen, V.) **11476** (Springer, Heidelberg, May 2019), 226–250.
9. Bardet, M., Faugère, J.-C. & Salvy, B. *Complexity of Gröbner basis computation for Semi-regular Overdetermined sequences over F_ 2 with solutions in F_ 2* Research Report RR-5049 (INRIA, 2003). `https://hal.inria.fr/inria-00071534`.
10. Bardet, M., Faugère, J.-C. & Salvy, B. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations (2004).
11. Bogdanov, A. & Qiao, Y. *On the Security of Goldreich's One-Way Function* in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* (eds Dinur, I., Jansen, K., Naor, J. & Rolim, J.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009), 392–405. ISBN: 978-3-642-03685-9.
12. Caminata, A. & Gorla, E. *Solving Multivariate Polynomial Systems and an Invariant from Commutative Algebra* in *Arithmetic of Finite Fields* (eds Bajard, J. C. & Topuzoğlu, A.) (Springer International Publishing, Cham, 2021), 3–36. ISBN: 978-3-030-68869-1.
13. Caminata, A. & Gorla, E. Solving Degree, Last Fall Degree, and Related Invariants. *J. Symb. Comput.* **114,** 322–335. ISSN: 0747-7171. `https://doi.org/10.1016/j.jsc.2022.05.001` (2023).
14. Charikar, M. & Wirth, A. *Maximizing Quadratic Programs: Extending Grothendieck's Inequality* in *45th FOCS* (IEEE Computer Society Press, Oct. 2004), 54–60.
15. Cheng, C.-M., Chou, T., Niederhagen, R. & Yang, B.-Y. *Solving Quadratic Equations with XL on Parallel Architectures* in *CHES 2012* (eds Prouff, E. & Schaumont, P.) **7428** (Springer, Heidelberg, Sept. 2012), 356–373.
16. Courtois, N., Klimov, A., Patarin, J. & Shamir, A. *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations* in *EU-*

ROCRYPT 2000 (ed Preneel, B.) **1807** (Springer, Heidelberg, May 2000), 392–407.

17. Couteau, G., Dupin, A., Méaux, P., Rossi, M. & Rotella, Y. *On the Concrete Security of Goldreich's Pseudorandom Generator* in *ASIACRYPT 2018, Part II* (eds Peyrin, T. & Galbraith, S.) **11273** (Springer, Heidelberg, Dec. 2018), 96–124.
18. Diem, C. *The XL-Algorithm and a Conjecture from Commutative Algebra* in *ASIACRYPT 2004* (ed Lee, P. J.) **3329** (Springer, Heidelberg, Dec. 2004), 323–337.
19. Ding, J., Buchmann, J., Mohamed, M., Moahmed, W. & Weinmann, R. Mutantxl. *SCC,* 16–22 (Jan. 2008).
20. Ding, J. & Schmidt, D. in *Number Theory and Cryptography: Papers in Honor of Johannes Buchmann on the Occasion of His 60th Birthday* (eds Fischlin, M. & Katzenbeisser, S.) 34–49 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013). ISBN: 978-3-642-42001-6. `https://doi.org/10.1007/978-3-642-42001-6_4`.
21. Dodis, Y., Reyzin, L. & Smith, A. *Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data* in *EUROCRYPT 2004* (eds Cachin, C. & Camenisch, J.) **3027** (Springer, Heidelberg, May 2004), 523–540.
22. Dubois, V. & Gama, N. *The Degree of Regularity of HFE Systems* in *ASIACRYPT 2010* (ed Abe, M.) **6477** (Springer, Heidelberg, Dec. 2010), 557–576.
23. Faugère, J. C. *A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5)* in *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation* (Association for Computing Machinery, Lille, France, 2002), 75–83. ISBN: 1581134843. `https://doi.org/10.1145/780506.780516`.
24. Faugére, J.-C. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra* **139,** 61–88. ISSN: 0022-4049. `https://www.sciencedirect.com/science/article/pii/S0022404999000055` (1999).
25. Goemans, M. X. & Williamson, D. P. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM* **42,** 1115–1145. ISSN: 0004-5411. `https://doi.org/10.1145/227683.227684` (1995).
26. Goldreich, O. in *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman* (ed Goldreich, O.) 76–87 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011). ISBN: 978-3-642-22670-0. `https://doi.org/10.1007/978-3-642-22670-0_10`.

27. Håstad, J., Impagliazzo, R., Levin, L. A. & Luby, M. A Pseudorandom Generator from any One-way Function. *SIAM Journal on Computing* **28,** 1364–1396 (1999).
28. Huang, M.-D. A., Kosters, M. & Yeo, S. L. *Last Fall Degree, HFE, and Weil Descent Attacks on ECDLP* in *CRYPTO 2015, Part I* (eds Gennaro, R. & Robshaw, M. J. B.) **9215** (Springer, Heidelberg, Aug. 2015), 581–600.
29. Impagliazzo, R. & Wigderson, A. *P = BPP if E Requires Exponential Circuits: Derandomizing the XOR Lemma* in *29th ACM STOC* (ACM Press, May 1997), 220–229.
30. Ishai, Y., Kushilevitz, E., Ostrovsky, R. & Sahai, A. *Cryptography with constant computational overhead* in *40th ACM STOC* (eds Ladner, R. E. & Dwork, C.) (ACM Press, May 2008), 433–442.
31. Jain, A., Lin, H. & Sahai, A. *Indistinguishability Obfuscation from Well-Founded Assumptions* in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, Virtual, Italy, 2021), 60–73. ISBN: 9781450380539. https://doi.org/10.1145/3406325.3451093.
32. Jain, A., Lin, H. & Sahai, A. *Indistinguishability Obfuscation from LPN over Fp, DLIN, and PRGs in NC0* in *Advances in Cryptology – EUROCRYPT 2022* (eds Dunkelman, O. & Dziembowski, S.) (Springer International Publishing, Cham, 2022), 670–699. ISBN: 978-3-031-06944-4.
33. Lang, S. *Algebra* ISBN: 9781461300410 146130041X (Springer, New York, NY, 2002).
34. Lazard, D. *Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations* in *Computer Algebra* (ed van Hulzen, J. A.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 1983), 146–156. ISBN: 978-3-540-38756-5.
35. Macaulay, F. The algebraic theory of modular systems. *Cambridge Mathematical Library* **xxxi** (1916).
36. Mohamed, M. S. E., Mohamed, W. S. A. E., Ding, J. & Buchmann, J. A. *MXL2: Solving Polynomial Equations over GF(2) Using an Improved Mutant Strategy* in *Post-quantum cryptography, second international workshop, PQCRYPTO 2008* (eds Buchmann, J. & Ding, J.) (Springer, Heidelberg, Oct. 2008), 203–215.
37. Mossel, E., Shpilka, A. & Trevisan, L. *On e-Biased Generators in NC0* in *44th FOCS* (IEEE Computer Society Press, Oct. 2003), 136–145.
38. Nisan, N. & Wigderson, A. *Hardness vs. Randomness (Extended Abstract)* in *29th FOCS* (IEEE Computer Society Press, Oct. 1988), 2–11.
39. ODonnell, R. & Witmer, D. *Goldreich's PRG: Evidence for Near-Optimal Polynomial Stretch* in (June 2014), 1–12. ISBN: 978-1-4799-3626-7.
40. Schwartz, J. T. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM* **27,** 701–717. ISSN: 0004-5411. https://doi.org/10.1145/322217.322225 (1980).

41. Siegenthaler, T. Correlation-immunity of nonlinear combining functions for cryptographic applications (Corresp.) *IEEE Transactions on Information Theory* **30,** 776–780 (1984).
42. Sugita, M., Kawazoe, M. & Imai, H. Relation between the XL Algorithm and Gröbner Basis Algorithms. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E89-A,** 11–18. ISSN: 0916-8508. `https://doi.org/10.1093/ietfec/e89-a.1.11` (2006).
43. Viola, E. *The Sum of d Small-Bias Generators Fools Polynomials of Degree d* in *2008 23rd Annual IEEE Conference on Computational Complexity* (2008), 124–127.
44. Yang, B.-Y. & Chen, J.-M. *All in the XL Family: Theory and Practice* in *Information Security and Cryptology – ICISC 2004* (eds Park, C.-s. & Chee, S.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005), 67–86. ISBN: 978-3-540-32083-8.

Supplementary Material

A Appendix

A.1 Finding *Reduced* Algebraic Relations

In this subsection, we want to focus on the case where $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ is a PRG of constant *degree* d and where q is small (2 or $O(\sqrt{n})$ for example). Note, that Theorem 2 does not give us a meaningful attack for small values of q.

We will construct a Subexponential attack algorithm $\mathcal{A}3$ on F in those cases that has a subexponentially small advantage in distinguishing outputs of F from randomness. For this end, we will need to compute algebraic relations $h \in \mathbb{Z}_q[Y_1, \ldots, Y_m]$ of sublinear degree, which are *reduced* modulo the field equations, i.e. each monomial Y_i appears with degree at most Y_i^q. First, we will talk about the structure of reduced polynomials:

Remark 4. Denote by $I \subset \mathbb{Z}_q[X]$ the ideal generated by the field equations of $\mathbb{Z}_q$, i.e.

$$I := (X_1^q - X_1, \ldots, X_n^q - X_n). \tag{125}$$

The ring $\mathbb{Z}_q[X]/I$ is not graded any more, since I is not a homogenous ideal. However, it is still filtrated where the filtration steps are given by the vector spaces

$$\mathbb{Z}_q[X]^{\leq \ell}/(I \cap \mathbb{Z}_q[X]^{\leq \ell}). \tag{126}$$

A basis for $\mathbb{Z}_q[X]^{\leq \ell}/(I \cap \mathbb{Z}_q[X]^{\leq \ell})$ is given by the set of all monomials of degree $\leq \ell$ where each variable occurs at most $q-1$ times. Therefore, we have for $\ell \leq n$

$$\binom{n}{\ell} \leq \dim_{\mathbb{Z}_q}(\mathbb{Z}_q[X]^{\leq \ell}/(I \cap \mathbb{Z}_q[X]^{\leq \ell})) \leq \binom{n+\ell}{\ell} = \dim_{\mathbb{Z}_q}(\mathbb{Z}_q[X]^{\leq \ell}). \tag{127}$$

To avoid trivial relations over $\mathbb{Z}_q$, we will present here a modified version of $\mathcal{B}1$ – that we will call $\mathcal{B}2$ – that will always find a reduced algebraic relation of polynomials over $\mathbb{Z}_q$. For this sake, we set by abuse of notation

$$R_q[X] := \mathbb{Z}_q[X_1, \ldots, X_n]/(X_1^q - X_1, \ldots, X_n^q - X_n), \tag{128}$$
$$R_q[Y] := \mathbb{Z}_q[Y_1, \ldots, Y_m]/(Y_1^q - Y_1, \ldots, Y_m^q - Y_m). \tag{129}$$

As explained in Remark 4, the rings $R_q[X]$ and $R_q[Y]$ are filtrated.

Now let F be a PRG of degree d over $\mathbb{Z}_q$ and let $f_1, \ldots, f_m \in \mathbb{Z}_q[X]$ be the polynomials that make up F. Without loss of generality, we can assume that $f_1, \ldots, f_m$ are reduced modulo the field equations $X_1^q - X_1, \ldots, X_n^q - X_n$. Therefore, by abuse of notation, we interpret $f_1, \ldots, f_m$ as elements of $R_q[X]$. Now, the dual map $\phi : \mathbb{Z}_q[Y] \to \mathbb{Z}_q[X]$ descends well-defined to a ring homomorphism

$$\phi_q : R_q[Y] \longrightarrow R_q[X] \tag{130}$$
$$Y_i \longmapsto f_i(X). \tag{131}$$

For the kernel of ϕ_q, we have

$$\ker \phi_q = (\ker \phi + (Y_1^q - Y_1, \ldots, Y_m^q - Y_m))/(Y_1^q - Y_1, \ldots, Y_m^q - Y_m). \tag{132}$$

I.e., $\ker \phi_q$ contains all algebraic relations of $\ker \phi$ modulo the trivial ones from the field equations of $\mathbb{Z}_q$. In particular, a non-zero element of $\ker \phi_q$ is now guaranteed to not vanish everywhere on $\mathbb{Z}_q^m$.

To find a non-zero element of $\ker \phi_q$, the algorithm $\mathcal{B}2$ will proceed similarly as $\mathcal{B}1$: For increasing $\ell = 1, \ldots, \dim_{\mathbb{Z}_q} R_q[Y]$, the algorithm $\mathcal{B}2$ computes a basis of the $\mathbb{Z}_q$-vector space $\ker \phi_q \cap R_q[Y]^{\leq \ell}$. If $\ker \phi_q \cap R_q[Y]^{\leq \ell}$ is non-zero, $\mathcal{B}2$ returns a non-zero element of it and terminates. Otherwise, $\mathcal{B}2$ increments ℓ and repeats these computations. Formally, $\mathcal{B}2$ is given by:

Definition 12. *The algorithm $\mathcal{B}2$ gets as input numbers $n, m, d, q \in \mathbb{N}$, and a description of a polynomial map $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$. It has to output a non-zero element of $\ker \phi_q$.*

For $\ell = 1, \ldots, (q-1)m$, $\mathcal{B}2$ does the following:

1. *$\mathcal{B}2$ computes $N := \dim_{\mathbb{Z}_q} \left(R_q[X]^{\leq d\ell}\right)$ and $M := \dim_{\mathbb{Z}_q} \left(R_q[Y]^{\leq \ell}\right)$.*
2. *$\mathcal{B}2$ computes a finite list*

$$(Y_1^{a_1} \cdots Y_m^{a_m} \mid a_1, \ldots, a_m \in \{0, \ldots, q-1\}, a_1 + \ldots + a_m \leq \ell) \tag{133}$$
$$= (Y^{\alpha_1}, \ldots, Y^{\alpha_M}) \tag{134}$$

   of all monomials in $R_q[Y]$ of degree $\leq \ell$.
3. *$\mathcal{B}2$ applies ϕ_q to each Y^{α_i} and computes a second list $(\phi_q(Y^{\alpha_1}), \ldots, \phi_q(Y^{\alpha_M}))$ of polynomials in $R_q[X]$ of degree $\leq d\ell$.*
4. *Let $X^{\beta_1}, \ldots, X^{\beta_N}$ be the set of all monomials in $R_q[X]$ of degree $\leq d\ell$ where each variable appears at most $q-1$ times. For each $\phi_q(Y^{\alpha_i})$ let $w_i = (w_{i,1}, \ldots, w_{i,N}) \in \mathbb{Z}_q^N$ be the unique column vector s.t.*

$$\phi_q(Y^{\alpha_i}) = \sum_{j=1}^{N} w_{i,j} \cdot X^{\beta_j}. \tag{135}$$

   These vectors give us the matrix

$$W_\ell := \left(w_1 | \ldots | w_M\right) \in \mathbb{Z}_q^{N \times M}. \tag{136}$$

5. *$\mathcal{B}2$ uses Gaussian elimination over $\mathbb{Z}_q$ to compute the kernel of W_ℓ*

$$K_\ell := \left\{ r \in \mathbb{Z}_q^M \mid W_\ell \cdot r = 0 \right\}. \tag{137}$$

6. *If K_ℓ is the trivial null-space, $\mathcal{B}2$ increases ℓ by one. If $\ell \leq (q-1)m$, $\mathcal{B}2$ goes back to step 1.*

7. *Otherwise, if $\ell > (q-1)m$, $\mathcal{B}2$ exhausted the whole vector space $R_q[Y]$. In this case, $\mathcal{B}2$ knows that $\ker \phi_q$ is trivial and aborts.*
8. *If K_ℓ is not the null-space, $\mathcal{B}2$ chooses an arbitrary non-zero vector $r \in K_\ell$, computes the polynomial*

$$h := r_1 \cdot Y^{\alpha_1} + \ldots + r_M \cdot Y^{\alpha_M} \in R_q[Y] \tag{138}$$

of total degree $\leq \ell$ and outputs it.

We have for $\mathcal{B}2$ similar time and space bounds as for $\mathcal{B}1$:

Lemma 9. *Assume that $\mathcal{B}2$ terminates after D iterations. Then, its space complexity can be bounded by $O(NM)$ and its time complexity can be bounded by $O(DN^2M)$ for $N \leq \binom{n+dD}{dD}$ and $M = \binom{m+D}{D}$.*

Similarly, as in Section 3, one can show that $\mathcal{B}2$ will return an algebraic relation of minimal degree, if such a relation exists:

Lemma 10. *Let $n, m, d \in \mathbb{N}$, $m > n$. Let $f_1, \ldots, f_m \in R_q[X]$ be polynomials of degree $\leq d$ and set $D := \min\{\deg h \mid h \in \ker \phi_q, h \neq 0\}$. Then, $\mathcal{B}2$ terminates after D iterations and outputs a non-zero element of $\ker \phi_q$ of degree D.*

The inequality in Lemma 7 has a pendant that states that for almost all n we have

$$\dim_{\mathbb{Z}_q}(R_q[Y]^L) > \dim_{\mathbb{Z}_q}(R_q[X]^{dL}) \tag{139}$$

where $L(n) = \left\lceil \left(2^d n^d / m\right)^{\frac{1}{d-1}} \right\rceil$.

Lemma 11 (Main Inequality for Small q). *Let $d \in \mathbb{N}, d \geq 2$. Let $q \in \mathbb{N}$ and let $m : \mathbb{N} \to \mathbb{N}$ be a function with $m(n) \geq d^{d-1} 2^d n$. Further, set $c = 2^{\frac{d}{d-1}}$. Then, we have for almost all integers n*

$$\dim_{\mathbb{Z}_q}(R_q[Y]^L) > \dim_{\mathbb{Z}_q}(R_q[X]^{dL}) \tag{140}$$

where $L(n) = \left\lceil \left(2^d n^d / m\right)^{\frac{1}{d-1}} \right\rceil$.

Proof. We can lower-bound the left-hand side of the equation by $\binom{m}{L}$ and upper-bound the left-hand side by $\binom{n+dL}{dL}$. Therefore, it suffices to show

$$\binom{m}{L} > \binom{n+dL}{dL}. \tag{141}$$

Rolling out both sides, we need to show

$$\frac{m \cdots (m-L+1)}{L \cdots 2 \cdot 1} > \frac{(n+dL) \cdots (n+1)}{(dL) \cdots 2 \cdot 1}. \tag{142}$$

We can rewrite this as

$$m \cdots (m-L+1) \cdot (dL) \cdots (L+1) > (n+dL) \cdots (n+1). \tag{143}$$

For n large enough, $n + dL$ is smaller than $2n$, since $m \geq d^{d-1}2^d n$. Therefore, it suffices to show

$$m \cdots (m-L+1) \cdot (dL) \cdots (L+1) \geq 2^{dL} \cdot n^{dL}. \tag{144}$$

We claim that the right-hand side is bigger than $m^L \cdot L^{L(d-1)}$. In fact, we have the equivalences

$$\begin{align} & m \cdots (m-L+1) \cdot (dL) \cdots (L+1) \geq m^L \cdot L^{L(d-1)} \tag{145} \\ \iff & \frac{(dL) \cdots (L+1)}{L^{L(d-1)}} \geq \frac{m^L}{m \cdots (m-L+1)} \tag{146} \\ \iff & \frac{dL}{L} \cdot \frac{dL-1}{L} \cdots \frac{L+1}{L} \geq \frac{m}{m} \cdots \frac{m}{m-L+1}. \tag{147} \\ & \tag{148} \end{align}$$

The last inequality does hold if $m > 2L$, since in this case we have that the last L factors $\frac{L+L}{L}, \ldots, \frac{L+1}{L}$ on the left-hand side are bigger than the corresponding L factors $\frac{m}{m-L+1}, \ldots, \frac{m}{m}$ on the right-hand side.

Therefore, for n large enough, the inequality Eq. (144) is implied by

$$m^L \cdot L^{L(d-1)} \geq 2^{dL} \cdot n^{dL}. \tag{149}$$

By computing L-th roots on both sides, we get the equivalent inequality

$$m \cdot L^{(d-1)} \geq 2^d \cdot n^d \tag{150}$$

which does hold. This finishes the proof.

Now, let $e > 0$ and $d \in \mathbb{N}$ be constants and assume $m \geq n^{1+e}$. In this case, the above lemma gives us $L = \lceil c \cdot n^{1-\frac{e}{d-1}} \rceil$.

It follows that $\mathcal{B}2$'s complexity is subexponential for $m \geq n^{1+e}$ polynomials $f_1, \ldots, f_m$:

Theorem 5. *Let $d \in \mathbb{N}$ be constant and $m \in \omega(n)$. Let $f_1, \ldots, f_m \in R_q[X]$ be polynomials of degree $\leq d$.*

Then, the algorithm $\mathcal{B}2$ in Definition 12 outputs a non-trivial element of $\ker \phi_q$ *of degree $O\left((n^d/m)^{\frac{1}{d-1}}\right)$. Its space and time complexities lie in $n^{O((n^d/m)^{\frac{1}{d-1}})}$.*

If $m \geq n^{1+e}$ for some constant $e > 0$, then the complexities of $\mathcal{B}2$ are bounded by $n^{O(n^{1-\frac{e}{d-1}})}$ and the degree of its output lies in $O(n^{1-\frac{e}{d-1}})$

A.2 Attacks on Constant-Degree PRGs over *Small* Moduli

$\mathcal{B}2$ gives rise to the following attacker $\mathcal{A}3$ on degree-d PRGs over $\mathbb{Z}_q$:

Definition 13. *The algorithm $\mathcal{A}3$ receives as input a description of a PRG $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ of degree d, which includes the numbers $n, m, d, q \in \mathbb{N}$, and an element $y^* \in \mathbb{Z}_q^m$. The goal of $\mathcal{A}3$ is to output 0, if y^* lies in the image of F, and 1, otherwise.*

$\mathcal{A}3$ proceeds in two simple steps:

1. *$\mathcal{A}3$ executes the algorithm $\mathcal{B}2$ from Definition 12 on the input n, m, d, F and receives a non-zero polynomial $h \in R_q[Y]$ as output.*
2. *$\mathcal{A}3$ outputs 0 if $h(y^*) = 0$. Otherwise, $\mathcal{A}3$ outputs 1.*

It is clear that $\mathcal{A}3$'s space and time complexities are comparable to the space and time complexities of $\mathcal{B}2$. However, since the degree D of h will be much higher than the cardinality of $\mathbb{Z}_q$, we cannot apply the Schwartz-Zippel Lemma any more. Since h is not zero in $R_q[Y]$, we can only guarantee that h vanishes on at most $q^m - q^{m-D}$ points of $\mathbb{Z}_q^m$:

Lemma 12. *Let $f \in R_q[Y_1, \ldots, Y_m]$ be a non-zero polynomial of degree d. Then, we have*

$$\#\left\{y \in \mathbb{Z}_q^m \mid f(y) = 0\right\} \leq q^m - q^{m-d}. \tag{151}$$

Proof. Since f is non-zero modulo $(Y_1^q - Y_1, \ldots, Y_m^q - Y_m)$, we can interpret it as a non-zero polynomial in $\mathbb{Z}_q[Y]$ that has degree at most $q-1$ in each variable Y_i.

First assume that no linear polynomial of the form $(Y_m - c)$ for $c \in \mathbb{Z}_q$ divides f (over $\mathbb{Z}_q[Y]$). In that case, f can be written as

$$f(Y_1, \ldots, Y_m) = \sum_{i=0}^{q-1} \frac{Y_m^q - Y_m}{Y_m - i} \cdot g_i(Y_1, \ldots, Y_{m-1}) \tag{152}$$

where each $g_i \in \mathbb{Z}_q[Y_1, \ldots, Y_m]$ is reduced, non-zero and of degree $\leq d$ (in fact, g_i is a scalar multiple of $f(Y_1, \ldots, Y_{m-1}, i)$). Then, we have

$$\begin{aligned} &\#\left\{y \in \mathbb{Z}_q^m \mid f(y) = 0\right\} && (153)\\ =&\#\left\{y \in \mathbb{Z}_q^{m-1} \mid g_0(y) = 0\right\} + \ldots + \#\left\{y \in \{0,1\}^{m-1} \mid g_{q-1}(y) = 0\right\}. && (154)\end{aligned}$$

By an inductive argument, the claim now follows.

On the other side, assume that f is divisible by a linear term $(Y_m - c)$. W.l.o.g. $c = 0$, ergo f decomposes as $f = f' \cdot Y_m$ where $\deg f' \leq d - 1$. We have

$$\begin{aligned} &\#\left\{y \in \mathbb{Z}_q^m \mid f(y) = 0\right\} && (155)\\ =&\#\left\{y \in \mathbb{Z}_q^m \mid y_m = 0\right\} && (156)\\ &+ \#\left\{y \in \mathbb{Z}_q^{m-1} \mid f'(y_1, \ldots, y_{m-1}, 1) = 0\right\} && (157)\\ &+ \ldots + \#\left\{y \in \mathbb{Z}_q^{m-1} \mid f'(y_1, \ldots, y_{m-1}, q-1) = 0\right\}. && (158)\end{aligned}$$

By an inductive argument, the right-hand side is smaller than

$$q^{m-1} + (q-1)(q^{m-1} - q^{m-d}) = q^m - (q-1)q^{m-d} \leq q^m - q^{m-d}. \tag{159}$$

This finishes the proof of the lemma.

This gives us the following theorem:

Theorem 6. *Let $d \in \mathbb{N}$ be constant and let $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ be a PRG of degree d and $m \in \omega(n)$.*

Then, there is an attack algorithm $\mathcal{A}3$ whose time and space complexities are bounded from above by $n^{O((n^d/m)^{\frac{1}{d-1}})}$. Further, there exists a constant $c > 0$ s.t. $\mathcal{A}3$'s advantage in the security game in Definition 5 against F is lower bounded by

$$\mathsf{adv}_F(\mathcal{A}3) \geq q^{-c\cdot(n^d/m)^{\frac{1}{d-1}}}. \tag{160}$$

If $m \geq n^{1+e}$, this gives us an attack algorithm of complexity $n^{O(n^{1-\frac{e}{d-1}})}$ and minimum advantage $q^{-c\cdot n^{1-\frac{e}{d-1}}}$.

Theorem 6 is unsatisfying, since $\mathcal{A}3$'s advantage can only be guaranteed to be at least subexponential. One solution for this problem is to look at a multi-challenge security game for the PRG F where the adversary receives Q challenges $y_1^*, \ldots, y_Q^* \in \mathbb{Z}_q^m$ and has to guess if all $y_1^*, \ldots, y_Q^*$ have been drawn uniformly and independently at random from $\mathbb{Z}_q^m$ or if all $y_1^*, \ldots, y_Q^*$ lie in the image of F.

If the number of challenges is $Q \in q^{\Omega(n^{1-\frac{e}{d-1}})}$, for $m \geq n^{1+e}$, then the advantage of $\mathcal{A}3$ can be amplified to a positive constant.

To prove this, we first give a formal definition of a security game for PRGs in the multi-challenge setting.

Definition 14 (Multi-Challenge Security Game for Pseudrandom Number Generators). *Let k be a finite field and let $F : k^n \to k^m$ be a PRG. Let $Q = Q(n)$ be the number of challenges that are given to the adversary.*

We describe here a non-interactive security game between a probabilistic challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The game is parametrized by n and proceeds in the following steps:

1. *$\mathcal{C}$ draws a bit $b \leftarrow \{0,1\}$. If $b = 0$, it samples preimages $x_1, \ldots, x_Q \leftarrow k^n$ uniformly at random, computes $F(x_1), \ldots, F(x_Q)$ and sends $(F, F(x_1), \ldots, F(x_Q))$ to $\mathcal{A}$. If $b = 1$, it samples $y_1, \ldots, y_Q \leftarrow k^m$ and sends $(F, y_1, \ldots, y_Q)$ to $\mathcal{A}$.*
2. *$\mathcal{A}$ receives $(F, y_1^*, \ldots, y_Q^*)$ for some $y_1^*, \ldots, y_Q^* \in k^m$ and must decide which bit b has been drawn by $\mathcal{C}$. It makes some computations on its own without interacting with $\mathcal{C}$ and finally sends a bit b' to $\mathcal{C}$.*

$\mathcal{A}$ wins an instance of this game iff $b = b'$ holds at the end. We define $\mathcal{A}$'s advantage against F by

$$\begin{aligned}
\mathsf{adv}_F^Q(\mathcal{A}) &:= 2\Pr[\mathcal{A}\ \textit{wins}] - 1 && (161)\\
&= \Pr_{x_1,\dots,x_Q \leftarrow k^n}[\mathcal{A}(F, F(x_1), \dots, F(x_Q)) = 0] && (162)\\
&+ \Pr_{y_1,\dots,y_Q \leftarrow k^m}[\mathcal{A}(F, y_1, \dots, y_Q) = 1] - 1 && (163)
\end{aligned}$$

where we take the probability over the randomness of $\mathcal{A}$ and $\mathcal{C}$.

We define $\mathcal{A}$'s space complexity to be the number of bits and elements of k it stores simultaneously in step 2, and we define its time complexity by the number of bit-operations and arithmetical operations over k it performs in step 2.

Theorem 7 (Multi-Challenge Attack). *Let $d \in \mathbb{N}, e > 0$ be constants. Let $q \leq n$. Let $m \geq n^{1+e}$ and let $F : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ be of degree d. Let $Q \in q^{\Theta(n^{1-\frac{e}{d-1}})}$.*

Then, there is an attack algorithm $\mathcal{A}3^{\text{multi}}$ whose time and space complexity is bounded from above by $n^{O(n^{1-\frac{e}{d-1}})}$. Further, there exists a constant $c > 0$ s.t. $\mathcal{A}3^{\text{multi}}$'s advantage in the multi-challenge security game in Definition 14 of F is lower bounded by c.

Before we can show Theorem 7, we first show a technical lemma that will prove to be helpful.

Lemma 13. *Let $a, b > 0$. We have for almost all $t \in \mathbb{N}$*

$$\left(1 - \frac{1}{t^a}\right)^{t^b} \leq \frac{1}{e}. \tag{164}$$

Proof. First note, that we have for each $x \geq 1$

$$\left(1 - \frac{1}{x}\right)^{x} \leq \frac{1}{e}. \tag{165}$$

We distinguish three cases:

Case 1: $a < b$.
In this case, we have

$$\left(1 - \frac{1}{t^a}\right)^{t^b} \leq \left(1 - \frac{1}{t^b}\right)^{t^b}, \tag{166}$$

since $t^a \leq t^b$. Since $t^b \geq 1$, we ergo have

$$\left(1 - \frac{1}{t^a}\right)^{t^b} \leq \left(1 - \frac{1}{t^b}\right)^{t^b} \leq \frac{1}{e}. \tag{167}$$

Case 2: $a = b$.

In this case, we have

$$\left(1 - \frac{1}{t^a}\right)^{t^b} = \left(1 - \frac{1}{t^b}\right)^{t^b} \leq \frac{1}{e} \tag{168}$$

since $t^b \geq 1$.

Case 3: $a > b$.

In this case, we have

$$\left(1 - \frac{1}{t^a}\right)^{t^b} = \left(1 - \frac{1}{t^a}\right)^{t^a \cdot t^{b-a}} = \left(\left(1 - \frac{1}{t^a}\right)^{t^a}\right)^{t^{b-a}}. \tag{169}$$

Since $t^{b-a} \geq 1$ for t big enough, we get

$$\left(1 - \frac{1}{t^a}\right)^{t^b} = \left(\left(1 - \frac{1}{t^a}\right)^{t^a}\right)^{t^{b-a}} \leq \left(1 - \frac{1}{t^a}\right)^{t^a} \leq \frac{1}{e}. \tag{170}$$

Proof. The algorithm $\mathcal{A}3^{\text{multi}}$ receives as input the numbers $n, m, d, q \in \mathbb{N}$, a description of F and the challenges $y_1^*, \ldots, y_Q^* \in \mathbb{Z}_q^m$. The goal of $\mathcal{A}3^{\text{multi}}$ is to output 0, if each y_i^* lies in the image of F, and 1, otherwise.

$\mathcal{A}3^{\text{multi}}$ works similar as $\mathcal{A}3$ and will first use $\mathcal{B}2$ to compute an algebraic relation h of F. If h vanishes on each y_i^*, $\mathcal{A}3^{\text{multi}}$ will output 0, otherwise it will output 1. $\mathcal{A}3^{\text{multi}}$ proceeds as follows:

1. $\mathcal{A}3^{\text{multi}}$ uses $\mathcal{B}2$ with input F to compute a non-zero algebraic relation $h \in R_q[Y]$.
2. $\mathcal{A}3^{\text{multi}}$ evaluates h one each y_i^*. If we have $h(y_1^*) = \ldots = h(y_Q^*) = 0$, then $\mathcal{A}3^{\text{multi}}$ outputs 0.
3. Otherwise, $\mathcal{A}3^{\text{multi}}$ outputs 1.

We first analyse the space and time complexities of $\mathcal{A}3^{\text{multi}}$: In its first step, $\mathcal{A}3^{\text{multi}}$'s complexity is bounded by the complexity of $\mathcal{B}2$, which is upper-bounded by $n^{O(n^{\frac{e}{d-1}})}$. Let $a > 0$ be constant s.t. the degree D of h outputted by $\mathcal{B}2$ is smaller than $\leq a \cdot n^{1-\frac{e}{d-1}}$. Evaluating h on one y_i^* costs less than

$$\lfloor a \cdot n^{1-\frac{e}{d-1}} \rfloor \cdot \left(\binom{m + \lfloor a \cdot n^{1-\frac{e}{d-1}} \rfloor}{\lfloor a \cdot n^{1-\frac{e}{d-1}} \rfloor} + 1 \right) \tag{171}$$

$$\leq a \cdot n^{1-\frac{e}{d-1}} \cdot m^{a \cdot n^{1-\frac{e}{d-1}}} \leq n^{v \cdot n^{1-\frac{e}{d-1}}} \tag{172}$$

operations for some constant $v > 0$. Let $\overline{u} > 0$ be a constant s.t. $Q \leq q^{\overline{u} \cdot n^{1-\frac{e}{d-1}}}$. Then, the cost of evaluating h on Q points can be upper-bounded by

$$Q \cdot n^{v \cdot n^{1-\frac{e}{d-1}}} \leq n^{\overline{u} \cdot n^{1-\frac{e}{d-1}}} \cdot n^{v \cdot n^{1-\frac{e}{d-1}}} \tag{173}$$

$$= n^{(\overline{u}+v) \cdot n^{1-\frac{e}{d-1}}} \in n^{O(n^{1-\frac{e}{d-1}})} \tag{174}$$

operations. Therefore, the complexity of $\mathcal{A}3^{\text{multi}}$ lies in $n^{O(n^{1-\frac{e}{d-1}})}$.

Now, we want to analyse the advantage of $\mathcal{A}3^{\text{multi}}$ in the multi-challenge security game:

1. If $b = 0$, then the points $y_1^*, \ldots, y_Q^*$ all lie in the image of F. In this case, h will vanish on all $y_1^*, \ldots, y_Q^*$ and $\mathcal{A}3^{\text{multi}}$ will output 0.
2. If $b = 1$, then each point y_i^* has been sampled uniformly from $\mathbb{Z}_q^m$. Lemma 12 bounds the number of roots of zeros of h by $\leq q^m - q^{m-D}$ where D is the degree of h. Therefore, for $i \in [m]$, we have

$$\Pr_{y_i^* \leftarrow \mathbb{Z}_q^m}[h(y_i^*) = 0] \leq \frac{q^m - q^{m-D}}{q^m} = 1 - \frac{q^{m-D}}{q^m} = 1 - q^{-D}. \tag{175}$$

The probability, that h vanishes on each y_i^* can be upper-bounded by

$$\begin{aligned} &\Pr_{y_1^*,\ldots,y_Q^* \leftarrow \mathbb{Z}_q^m}[h(y_1^*) = \ldots = h(y_Q^*) = 0] && (176)\\ =\ &\Pr_{y_1^* \leftarrow \mathbb{Z}_q^m}[h(y_1^*) = 0]^Q \leq (1 - q^{-D})^Q. && (177)\end{aligned}$$

Let $\underline{u} > 0$ be constant s.t. $Q \geq q^{\underline{u} \cdot n^{1-\frac{e}{d-1}}}$. Note, that $D \leq a \cdot n^{1-\frac{e}{d-1}}$ for $a > 0$ constant. Then, we have

$$\begin{aligned} &\Pr_{y_1^*,\ldots,y_Q^* \leftarrow \mathbb{Z}_q^m}[h(y_1^*) = \ldots = h(y_Q^*) = 0] && (178)\\ \leq\ &(1 - q^{-D})^Q && (179)\\ \leq\ &\left(1 - \frac{1}{q^{a \cdot n^{1-\frac{e}{d-1}}}}\right)^{q^{\underline{u} \cdot n^{1-\frac{e}{d-1}}}} && (180)\\ \leq\ &\left(1 - \frac{1}{\left(q^{n^{1-\frac{e}{d-1}}}\right)^a}\right)^{\left(q^{n^{1-\frac{e}{d-1}}}\right)^{\underline{u}}} . && (181)\end{aligned}$$

Lemma 13 states now that we have for almost all $n \in \mathbb{N}$

$$\begin{aligned} &\Pr_{y_1^*,\ldots,y_Q^* \leftarrow \mathbb{Z}_q^m}[h(y_1^*) = \ldots = h(y_Q^*) = 0] && (182)\\ \leq\ &\left(1 - \frac{1}{\left(q^{n^{1-\frac{e}{d-1}}}\right)^a}\right)^{\left(q^{n^{1-\frac{e}{d-1}}}\right)^{\underline{u}}} \leq \frac{1}{e}. && (183)\end{aligned}$$

Therefore, the probability that $\mathcal{A}3^{\text{multi}}$ will output 1 if each y_i^* has been sampled from $\mathbb{Z}_q^m$ uniformly at random is at least $1 - \frac{1}{e}$ for n big enough.

Ergo, we can lower-bound the advantage of $\mathcal{A}3^{\text{multi}}$ for almost all n by

$$\begin{aligned} \mathsf{adv}_F^Q(\mathcal{A}3^{\text{multi}}) &= \Pr_{x_1,\dots,x_Q \leftarrow \mathbb{Z}_q^n}[\mathcal{A}(F, F(x_1), \dots, F(x_Q)) = 0] && (184)\\ &+ \Pr_{y_1,\dots,y_Q \leftarrow \mathbb{Z}_q^m}[\mathcal{A}(F, y_1, \dots, y_Q) = 1] - 1 && (185)\\ &\leq 1 + (1 - \frac{1}{e}) - 1 = 1 - \frac{1}{e}. && (186) \end{aligned}$$

A.3 Transcendence and Function Fields

Let k be any field. In Section 3, we claimed that each set of $m > n$ polynomials $f_1, \dots, f_m \in k[X] = k[X_1, \dots, X_n]$ must be algebraically dependent. In this subsection, we will prove this claim formally. For this end, we will study some properties of extensions of function fields and introduce the notion of *transcendental* field extensions. For more background on transcendental field extensions, we refer the reader to the book of Serge Lang [33, Chapter 8].

We will first introduce the notion of *function fields*:

Definition 15. *Let k be any field and $k[X] = k[X_1, \dots, X_n]$. Since $k[X]$ is an integral domain (i.e. commutative and zero divisor-free), it can be embedded into its quotient field that is given by*

$$k(X) := \left\{ \frac{f}{g} \mid f, g \in k[X], g \neq 0 \right\}. \qquad (187)$$

*$k(X)$ is called the **function field** of n variables over k.*

Elements of function fields are called *rational functions.*

The extension $k \subset k(X)$ gives us a prime example of a transcendental field extension.

Definition 16. *The inclusion $k \subset k(X)$ gives us an extension of fields. Let L be an intermediate field i.e. $k \subset L \subset k(X)$.*

*We call an element $f \in k(X)$ **transcendental** over L if the morphism of L-algebras*

$$\begin{aligned} L[T] &\longrightarrow k(X) && (188)\\ T &\longmapsto f(X) && (189) \end{aligned}$$

*is injective (where T is a fresh new variable). If f is not transcendental, we call it **algebraic** over L.*

*Given elements $f_1, \dots, f_m \in k(X)$ we call them a **transcendence basis** for the extension $k \subset k(X)$ if the following things hold:*

1. *For each $i \in [m]$, f_i is transcendental over $k(f_1, \dots, f_{i-1})$ (where $k(f_1, \dots, f_{i-1})$ is the smallest field in $k(X)$ that contains k and $f_1, \dots, f_{i-1}$).*
2. *$k(X)$ is algebraic over $k(f_1, \dots, f_m)$, i.e.*

$$\dim_{k(f_1,\dots,f_m)}(k(X)) < \infty. \qquad (190)$$

As we will see, transcendence bases are very similar to vector space bases. In fact, one defines the *degree of transcendence* of a field extension $k \subset L$ as the number of elements of a transcendence basis for this extension. We will see later that this notion is well-defined i.e. independent of the choice of the basis.

Remark 5. It can be shown that $f_1, \dots, f_m$ is a transcendence basis iff each of its permutation $f_{\pi(1)}, \dots, f_{\pi(n)}$, for $\pi \in S_n$, is a transcendence basis. Therefore, we can consider unordered sets as transcendence bases.

The following lemma shows how transcendence bases are related to the notion of algebraically independent polynomials:

Lemma 14. *Let $f_1, \dots, f_m \in k[X]$. The following are equivalent:*

1. *$f_1, \dots, f_m$ can be extended to a transcendence basis for $k \subset k(X)$.*
2. *The polynomials $f_1, \dots, f_m$ are algebraically independent.*

Proof. We prove each direction separately:

1. Let $f_1, \dots, f_m$ be s.t. they can be extended to a transcendence basis for $k \subset k(X)$. Let $g_1, \dots, g_r$ be a set of elements of $k(X)$ s.t. $f_1, \dots, f_m, g_1, \dots, g_r$ is a transcendence basis for $k(X)$ (note that r may be zero). Then, we have for each $i \in [m]$ that the map

$$\psi_i : L_i[T] \longrightarrow k(X) \tag{191}$$

$$T \longmapsto f_i \tag{192}$$

is injective where $L_i = k(f_1, \dots, f_{i-1}, f_{i+1}, \dots, f_m)$.
Assume – for the sake of contradiction – that $f_1, \dots, f_m$ are not algebraically independent. Set

$$K := \{h \in k[Y_1, \dots, Y_m] \mid h(Y) \neq 0, h(f_1(X), \dots, f_m(X)) = 0\} \tag{193}$$

and let $v \in K$ be of minimal degree. Let $i \in [m]$ be s.t. we can write v as

$$v(Y) = \sum_{j=0}^{d} c_j(Y) \cdot Y_i^j \tag{194}$$

with $c_j(Y) \in k[Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_m]$, $d > 0$ and $c_d \neq 0$.
We claim that the polynomial

$$l(T, X_1, \dots, X_n) := v(f_1(X), \dots, f_{i-1}(X), T, f_{i+1}(X), \dots, f_m(X)) \tag{195}$$

$$= \sum_{j=0}^{d} c_j(f_1(X), \dots, f_m(X)) \cdot T^j \tag{196}$$

is a non-zero element of $L_i[T]$. In fact, $c_d(f_1(X), \dots, f_m(X))$ cannot vanish, since c_d would be an element of K in that case. However, we picked v to be an element of minimal degree, and the degree of c_d is by $d > 0$ lower than the degree of v. Therefore, $l(T, X)$ is non-zero. Now, $l(f_i, X) = v(f_1, \dots, f_m) = 0$, therefore l lies in the kernel of ψ_i, therefore ψ_i is not injective. A contradiction!

2. Let $f_1, \dots, f_m$ be algebraically independent. Set $S_0 := \{f_1, \dots, f_m\}$. For $i = 1, \dots, n$, we proceed inductively as follows: If X_i is transcendental over $k(S_{i-1})$, set $S_i := S_{i-1} \cup \{X_i\}$. Otherwise, set $S_i := S_{i-1}$.
   By construction, we will end with a set S_n s.t. each X_i is algebraic over $k(S_n)$. In particular, the extension $k(S_n) \subseteq k(X)$ will be algebraic. On the other side, by construction, if $X_{a_1}, \dots, X_{a_l}$ are elements added to S_0, we have for each $i = 1, \dots, l$ that X_{a_l} is transcendental over $k(f_1, \dots, f_m, X_{a_1}, \dots, X_{a_{l-1}})$. So, it is left to show that for $i = 1, \dots, m$ the element f_i is transcendental over $k(f_1, \dots, f_{i-1})$. Assume – for the sake of contradiction – that this would not be the case for one $i \in [m]$. Let $h(T) \in k(f_1, \dots, f_{i-1})[T]$ s.t. $h(T) \neq 0$ and $h(f_i) = 0$. The function $h(T)$ can be written as

   $$h(T) = \sum_{i=0}^{d} c_i \cdot T^i \tag{197}$$

   with $c_1, \dots, c_d \in k(f_1, \dots, f_{i-1})$. Since we can multiply each c_j with the smallest common denominator of $c_0, \dots, c_d$, we can – without loss of generality – assume that each c_j lies in $k[f_1, \dots, f_{i-1}]$. In particular, there is a non-zero polynomial $v \in k[Y_1, \dots, Y_i]$ s.t.

   $$v(f_1, \dots, f_{i-1}, T) = h(T). \tag{198}$$

   Now, v is a non-trivial element of the kernel of

   $$\phi : k[Y] \longrightarrow k[X] \tag{199}$$

   $$Y_i \longmapsto f_i. \tag{200}$$

   Ergo, the elements $f_1, \dots, f_m$ are not algebraically independent. A contradiction!

The previous lemma justifies to extend the notion of algebraically independence to elements of $k(X)$ in the following way:

Definition 17. *We call a list of elements $f_1, \dots, f_m \in k(X)$* ***algebraically independent*** *or* ***transcendental*** *over k iff $f_1, \dots, f_m$ can be extended to a transcendence basis of $k \subset k(X)$.*

The next lemma is a well-known fact in the study of transcendent extension fields. It shows that, for a given extension $k \subset L$, all transcendence bases must have the same cardinality. We show this fact only in the case of finite transcendence bases of the extension $k \subset k(X)$:

Lemma 15. *Let $f_1, \dots, f_m, g_1, \dots, g_l \in k(X)$ be s.t. $\{f_1, \dots, f_m\}$ and $\{g_1, \dots, g_l\}$ are transcendence bases for $k \subset k(X)$. Then, we have*

$$m = l. \tag{201}$$

Proof. W.l.o.g., we assume $m > l$. We claim that there is an $i \in [l]$ s.t. the elements

$$g_i, f_2, \ldots, f_m \tag{202}$$

are algebraically independent. In fact, if each $g_1, \ldots, g_l$ would be algebraic over $k(f_2, \ldots, f_m)$, then f_1 would be algebraic over $k(f_2, \ldots, f_m)$ and $\{f_1, \ldots, f_m\}$ couldn't be algebraically independent.

We repeat this procedure for the first l elements of $f_1, \ldots, f_m$ and have finally that the elements

$$g_1, \ldots, g_l, f_{l+1}, \ldots, f_m \tag{203}$$

must be algebraically independent. However, this is not possible, since the elements $f_{l+1}, \ldots, f_m$ must be algebraic over $k(g_1, \ldots, g_l)$, since $\{g_1, \ldots, g_l\}$ is a transcendence basis for $k \subset k(X)$. A contradiction!

Since $X_1, \ldots, X_n$ is a transcendence basis of $k \subset k(X)$, it follows that the degree of transcendency of $k \subset k(X)$ must be n and each transcendence basis must have cardinality n. We can combine this insight with Lemma 14 to deduce the following corollary:

Corollary 2. *Let $f_1, \ldots, f_m \in k[X]$. If $f_1, \ldots, f_m$ are algebraically independent, then we must have*

$$m \leq n. \tag{204}$$

This corollary now explains why algorithm $\mathcal{B}1$ in Definition 7 must stop after a finite number of iterations if $m > n$.