On the minimal Hardware Complexity of Pseudorandom Function Generators

– Full Paper, November 28, 2000–

Matthias Krause and Stefan Lucks[⋆]

Theoretische Informatik, Univ. Mannheim, 68131 Mannheim, Germany
e-mail: {krause, lucks}@informatik.uni-mannheim.de

Abstract. A set F of Boolean functions is called a pseudorandom function generator (PRFG) if communicating with a randomly chosen secret function from F cannot be efficiently distinguished from communicating with a truly random function. We ask for the minimal hardware complexity of a PRFG. This question is motivated by design aspects of secure secret key cryptosystems. Such cryptosystems should be efficient in hardware, but often are required to behave like PRFGs. By constructing efficient distinguishing schemes we show for a wide range of basic nonuniform complexity classes, induced by depth restricted branching programs and several types of constant depth circuits (including TC_2^0), that they do not contain PRFGs. On the other hand we show that the PRFG proposed by Naor and Reingold in [24] consists of TC_4^0-functions. The question if TC_3^0-functions can form PRFGs remains as an interesting open problem. We further discuss relations of our results to previous work on cryptographic limitations of learning (see, e.g., [13]) and Natural Proofs [27].

Keywords: Cryptography, Pseudorandomness, Boolean Complexity Theory, Computational Distinguishability

1 Basic Definitions

Function Generators

A *function generator* F is an efficient (i.e., polynomial time) algorithm which for specific values of plaintext block length n computes for each plaintext block $x \in \{0,1\}^n$ and each key s from a predefined key set $S_n^F \subseteq \{0,1\}^{k(n)}$ a corresponding ciphertext output block $y = F_n(x,s) \in \{0,1\}^{l(n)}$. $k(n)$ and $l(n)$ are called key length and output length of F. The efficiency of F implies that $k(n)$ and $l(n)$ are polynomially bounded in n. Observe that the encryption mechanism of a secret key block cipher can be thought of as a function generator in a straightforward way. Clearly, cryptographic algorithms occuring in practice are usually designed for one specific input length n. However, in many cases the definition can be generalized to infinitely many values of admissible input length n in a more or less natural way. Correspondingly, we consider function generators to be sequences $F = (F_n)_{n\in\mathbb{N}}$ of sets of Boolean functions

$$F_n = \left\{ f_{n,s} : \{0,1\}^n \longrightarrow \{0,1\}^{l(n)};\ s \in S_n^F \right\},$$

where, if n is admissible, we define $f_{n,s}(x) = F_n(x,s)$.

A function generator F is **pseudorandom** if it is infeasible to distinguish between a (pseudorandom) function, which is randomly chosen from F_n, n admissable, and a truly random function $f \in B_n^{l(n)}$. (For $l,n \in \mathbb{N}$ let B_n^l denote the set of all $2^{2^{ln}}$ functions $f : \{0,1\}^n \longrightarrow \{0,1\}^l$.) In the sequel, we concentrate on functions $f : \{0,1\}^n \longrightarrow \{0,1\}^1$

[⋆] Supported by DFG grant Kr 1521/3-1.

and define $B_n = B_n^1$. Note that a truly random function in $B_n^l(n)$ is just a tuple of $l(n)$ independent random functions in B_n.

For giving the formal definition of pseudorandomness we introduce the notion of an H**-oracle**, where $H \subseteq B_n$. An H-oracle chooses randomly, via the uniform distribution on H, a secret function $h \in H$ and answers membership queries for inputs $x \in \{0,1\}^n$ immediately with $h(x)$. A **distinguishing algorithm** for a function generator $F = F_n$ is a randomized oracle Turing machine D which knows the definition of F, which gets an admissible input parameter n and which communicates via membership queries with an H-oracle, where either $H = B_n^{l(n)}$ (the truly random source) or $H = F_n$ (the pseudorandom source). The aim of D is to find out whether $H = B_n$ (in this case, D outputs 0) or $H = F_n$ (in this case, D outputs 1). Let us denote by $Pr_D(f)$ the probability that D accepts if the unknown oracle function is f.

The relevant cost parameters of a distinguishing algorithms D are the **worst case running time** $t_D = t_D(n)$ and the **advantage** $\varepsilon_D = \varepsilon_D(n)$, which is defined as

$$\begin{aligned}\varepsilon_D(n) &= \left|Pr[D \text{ outputs } 1|H = F_n] - Pr[D \text{ outputs } 1|H = B_n]\right| \\ &= \left|\mathbf{E}_{f\in F_n} Pr_D(f) - \mathbf{E}_{f\in B_n^{l(n)}} Pr_D(f)\right|.\end{aligned}$$

The **ratio** $r_D = r_D(n)$ of a distinguishing algorithm D is defined to be $r_D(n) = t_D(n) \cdot \varepsilon_D^{-1}(n)$.

Observe further that for any function generator F, there are two trivial strategies to distinguish it from a truly random source, which achieve ratio $O(|F_n|\log(|F_n|))$, the trivial upper bound. In both cases the distinguisher fixes a set X of inputs, where $|X|$ is the minimal number satisfying $2^{|X|} \geq 2|F_n|$. The first strategy is to fix a function $f \in F_n$ and to accept if the oracle coincides with f on X. This gives running time $O(|X|) = O(\log|F_n|)$ and advantage $\frac{1}{2}|F_n|^{-1}$. The second strategy is to check via exhaustive search whether there is some $f \in F_n$ which coincides with the oracle function on X. This implies advantage at least $\frac{1}{2}$ but running time $O(|F_n|\log(|F_n|))$.

We will call F to be a **pseudorandom function generator** (for short: **PRFG**) if for all distinguishing algorithms D for F it holds that $r_D \in 2^{n^{\Omega(1)}}$. Observe that this definition is similar to that in [7]. The difference is that in [7] only superpolynomiality is required.

Given a complexity measure M we denote by $P(M)$ the complexity class containing all sequences of (multi-output) Boolean functions which have polynomial size representations with respect to M. We say that a function generator F has M-complexity bounded by a function $c : \mathbb{N} \longrightarrow \mathbb{N}$ if for all n and all keys $s \in S_n^F$ it holds that $M(f_{n,s}) \leq c(n)$, and that F belongs to $P(M)$ if the M-complexity of F is bounded by some $c(n) \in n^{O(1)}$. We will call a complexity class **cryptographically strong** if it contains a PRFG, and **cryptographically weak** otherwise.

It is widely believed that there exist PRFGs (we will present a candidate in section 4), i.e., P/poly is supposed to be cryptographically strong. Pseudorandom function generators are of great interest in cryptography, e.g. as building blocks for block ciphers [20, 21], for remotely keyed encryption schemes [22, 3], for message authentication [2], and others. As the existence of PRFGs obviously implies $P \neq NP$, recent pseudorandomness proofs refer to unproven cryptographic hardness assumptions. In the following we will detect cryptographical strength – or weakness – for most of the basic nonuniform complexity classes.

A distinguishing algorithm $D = D(n,m)$, depending on the two input parameters n (input length) and m (complexity parameter), is called a **polynomial distinguishing scheme** with respect to M (resp. P(M)) if there are functions $t(n,m), \varepsilon^{-1}(n,m) \in (n+m)^{O(1)}$ such that for all polynomial bounds $m = m(n) \in n^{O(1)}$ and all (single output) functions $g \in B_n$ with $M(g) \leq m(n)$ it holds that $D(n,m)$ runs in time $t(n,m)$ and

$$Pr_D(g) - \mathbf{E}_{f\in B_n} Pr_D(f) \geq \varepsilon(n,m).$$

The definition of a **quasipolynomial distinguishing scheme** with respect to M can be obtained by replacing $t(n,m), \varepsilon^{-1}(n,m) \in (n+m)^{O(1)}$ by $t(n,m), \varepsilon^{-1}(n,m) \in (n+m)^{\log^{O(1)}(n+m)}$. We call a distinguishing scheme **efficient** if it is quasipolynomial or polynomial.

If there is an efficient distinguishing scheme D w.r.t. such a complexity measure M then, obviously, P(M) is cryptographically weak as each output bit of a function generator in P(M) can be efficiently distinguished via D. Consequently, as the **efficiency of key length** is a central design criterion for modern secret key encryption algorithms, these algorithms should have nearly maximal complexity w.r.t. to such complexity measures M. As cryptographers are searching for encryption mechanisms having hardware implementations which are very efficient with respect to time and energy consumption, there is a **low complexity danger** to get into the sphere of influence of one of the distinguishing schemes presented in this paper.

We consider several types of constant depth circuits over unbounded fan-in MOD$_m$, AND-, OR-, as well as bounded and unbounded weight threshold gates. The gate function MOD$_m$ is defined by MOD$_m(x_1, \ldots, x_n)$=1 if and only if $x_1 + \ldots + x_n \not\equiv 0 \mod m$. Unweighted treshold gates $T^n_{\geq r}$, resp. $T^n_{\leq r}$, are defined by the relations

$$T^n_{\geq r}(x_1, \ldots, x_n) = 1 \iff x_1 + \ldots + x_n \geq r$$

and $T^n_{\leq r}(x_1, \ldots, x_n) = 1 \iff x_1 + \ldots + x_n \leq r$. A weighted treshold gate $T^{\vec{a}}_{\geq r}$, where $\vec{a} \in \mathbb{Z}^n$, is defined by the relation

$$T^{\vec{a}}_{\geq r}(x_1, \ldots, x_n) = 1 \iff a_1 x_1 + \ldots + a_n x_n \geq r.$$

The inputs for the circuits are the constants 0 and 1 and literals from the set $\{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$. The definition of the mode of computation as well as the definition of AND- and OR-gates should be known. As usual, by AC^0_k, $AC^0_k[m]$, TC^0_k we denote the complexity classes consisting of all problems having polynomial size depth k circuits over AND-,OR-, resp. AND-, OR-, MOD$_m$-, resp. unweighted threshold gates.

We further consider branching programs, alternatively called binary decision diagrams (BDDs). A branching program for a Boolean function $f \in B_n$ is a directed acyclic graph $G = (V, E)$ with l sources. Each sink is labeled by a Boolean constant and each inner node by a Boolean variable. Inner nodes have two outgoing edges, one labeled by 0 and the other by 1. Given an input a, the output $f(a)_j$ is equal to the label of the sink reached by the unique path consistent with a and starting at source j, $1 \leq j \leq l$. Relevant restricted types of branching programs are

- Ordered binary decision diagrams (OBDDs), where each computational path has to respect the same variable ordering. An OBDD which respects a fixed variable ordering π is called a π-OBDD.
- Read-k-BDDs, for which on each path each variable is forbidden to occur more than k times.

2 Related Work, Our Results

Cryptographic Weakness

In section 3 we present efficient distinguishing schemes for the following complexity measures,

- a quasipolynomial scheme for the size of read-k BDDs (Theorem 3),
- a quasipolynomial scheme for the size of weighted Threshold-MOD$_2$ circuits, i.e. depth 2 circuits with a layer of MOD$_2$-gates connected with one output layer consisting of weighted threshold gates (Theorem 1),

- a quasipolynomial scheme for the size of constant depth circuits consisting of AND-, OR-, and MOD$_p$-gates, p prime (Theorem 2),
- a polynomial scheme for the size of unweighted threshold circuits of depth 2 (Theorem 4)
- a quasipolynomial scheme for the size of constant depth circuits having a constant number of layers of AND-, OR-gates connected with one output layer of weighted threshold gates (Theorem 5).

Observe that the function generator $f_{\vec{a}}(x_1,\ldots,x_n) = \sum_{i=1}^{n} a_i x_i$, where $\vec{a} \in \mathbb{Z}^n$, $(x_1,\ldots,x_n) \in \{0,1\}^n$, corresponding to the NP-hard **Subset Sum Problem**, belongs to TC_2^0 [28], which emphasizes the cryptographic weakness of this operation.

The complexity measures M handled in Theorems 3,1,2,4,5 represent a ”frontline” in the sense that they correspond to the most powerful models for which we know effective lower bound arguments, i.e., methods to show $\Pi \notin P(M)$ for some explicitely defined problem Π. Indeed, all our distinguishing schemes are inspired by the known lower bound arguments for the corresponding models and can be seen as some ”algorithmic version” of these arguments. It seems that searching for effective lower bound arguments for a complexity measure M is the same problem as searching for methods to distinguish unknown $P(M)$-functions from truly random functions. Note that a similar observation, but with respect to another mode of distinguishing, was made already by *Razborov* and *Rudich* in [27]. For illustrating the difference of their approach with our paper let us review the results in [27] in some more detail and start with the following definition.

Distinguishing Schemes versus Natural Proofs

Let $\Gamma \subseteq P/poly$ denote a complexity class and $T = (T_n) \in \Gamma$ be a sequence of Boolean functions for which the input length of T_n is N=2^n. T is called an efficient Γ-test against a function generator $F = (F_n)_{n\in\mathbb{N}}$ (consisting of single output functions) if for all n

$$\left|Pr_f[T_n(f) = 1] - Pr_s[T_n(f_{n,s}) = 1]\right| \geq p^{-1}(N) \qquad (1)$$

for a polynomially (in N) bounded function $p : \mathbb{N} \longrightarrow \mathbb{N}$. Hereby, functions $f \in B_n$ are considered to be strings of length $N = 2^n$. The probability on the left side is taken w.r.t. the uniform distribution on B_n (the truly random case), the probability on the right side is taken w.r.t. the uniform distribution on F_n (the pseudorandom case). The following observation was made in [27].

(1) It seems that all complexity classes Λ for which we know a method for proving that $F \notin \Lambda$ for some explicitely defined problem F have a so called Γ-Natural Proof for some complexity classes $\Gamma \subseteq P/poly$. (the somewhat technical definition of Natural Proofs is omitted here).
(2) On the other hand (and this is the property of Natural Proofs which is important in our context), if Λ has a Γ-Natural Proof then all function generators $F = (F_n)$ belonging to Λ have efficient Γ-tests.

The main implication of [27] is that a $P/poly$-Natural Proof against $P/poly$ would imply the nonexistence of function generators which are pseudorandom w.r.t. $P/poly$-tests. But this implies the nonexistence of pseudorandom bit generators [27], contradicting widely believed cryptographic hardness assumptins.

Observe that, in contrast to our concept of pseudorandomness, the existence of an efficient Γ-test for a given PRFG F does not yield any feasible attack against the corresponding cipher. This is because the whole function table has to be processed, which is of exponential size in n. Thus, informally speaken, the message of [27] is that effective lower bound arguments for M, as a rule, imply low complexity circuits which efficiently distinguish P(M)-functions from truly random functions, where the complexity is measured in the size of the whole function table. Our message is that effective lower bound arguments for M, as a rule, imply even efficient distinguishing attacks against each secret key encryption mechanism which belongs to P(M), where the running time is measured in the input length of the function. Observe that our most complicated distinguishing scheme for the size of constant depth circuits over AND, OR, MOD$_p$, p prime, (Theorem 2) uses an idea from [27] for constructing an NC^2-Natural Proof for AC$^0[p]$, $p > 2$ prime.

Cryptographic Strongness

In section 4 we try to identify the smallest complexity classes which are powerful enough to contain PRFGs. In [7], a general method for constructing PRFGs on the basis of pseudorandom bit generators is given. The construction is inherently sequential, and at first glance it seems hopeless to build PRFGs with small parallel time complexity. *Naor* and *Reingold* [23, 24] used a modified construction, based on concrete number-theoretic assumptions instead of generic pseudorandom bit generators. They presented a function generator (which we shortly call NR-generator, the definition will be presented in section 4) which is pseudorandom under the condition that the **Decisional Diffie-Hellman Assumption**, a widely believed cryptographic hardness assumption, is true. Moreover, the NR-generator belongs to TC^0, in [24] it is claimed (without proof) that it consists of TC^0_5-functions.

We show in Theorem 6 that the NR-generator even consists of TC^0_4-functions, i.e. TC^0_4 seems to be cryptographic strong while TC^0_2 has proved to be weak. It is an interesting open question if TC^0_3 is strong enough to contain PRFGs. Observe that TC^0_3 seems to contain pseudorandom bit generators, take hardcore bits of cryptographic one-way functions in TC^0_3 like discrete logarithm or squaring modulo the product of two prime numbers [28].

Some Remarks on Learning versus Distinguishing

Clearly, a successful distinguishing attack against a secret key encryption algorithm does not automatically imply that relevant information about the secret key can be efficiently computed. Observe that breaking the cipher corresponds to efficiently learning an unknown function from a known concept class. It is intuitively clear and not hard to prove that, with respect to any reasonable model of algorithmically learning Boolean concept classes from examples, any efficient learning algorithm for functions from a given complexity class Λ gives an efficient distinguishing scheme for Λ. (Use the learning algorithm to compute a low complexity hypothesis h of the unknown function f and test if h really approximates f.) Observe on the other hand that under the condition that membership queries are forbidden, each efficient distinguishing algorithm (which poses oracle queries only for randomly chosen inputs) can be simulated by an efficient weak learning algorithm, which computes a $\frac{1}{2} + \varepsilon$-approximator for the unknown function [4]. I.e., efficient **known plaintext** distinguishing attacks can be used to really break a cipher. There is some evidence that in the general case, if **chosen plaintext**, i.e., membership queries are allowed, this is not the case. It is not hard to see that there is a polynomial distinguishing scheme for polynomial size OBDDs.[1] On the other hand, there are several results proved in [17] which strongly support the following conjecture: it is impossible to efficiently learn the optimal variable ordering of a function with small OBDDs from examples.

In a certain sense the results of this paper can be considered as cryptographic limitations of proving lower bounds for complexity classes containing TC^0_4, while the results of [27] can be seen as cryptographic limitations of proving lower bounds against P/poly.

[1] Take disjoint random subsets of variables Y and Z of appropriate logarithmic size and test if the matrix $(f(y, z, \vec{0}))$, where y and z range over all assignments of Y and Z, resp., has small rank. As in the pseudorandom case with probability $1/poly(n)$, Y and Z are separated by the optimal variable ordering of the oracle function f. This gives an efficient test.

Observe that cryptographic limitations of learning were already detected by *Kearns* and *Valiant* in [13]. It is shown there that efficient learnability of TC^0_3-functions would contradict the existence of pseudorandom bit generators in TC^0_3 and thus to widely believed cryptographic hardness assumptions like the security of RSA or *Rabin*'s cryptosystem, see above.

Note that for all complexity classes Λ which are shown in section 3 to be cryptographically weak, it is unknown whether Λ- functions are efficiently learnable.

3 Distinguishing Schemes

Let us firstly consider the following basis test $T(p, \delta, N)$, where $\delta, p \in (0, 1)$, which accepts if

$$\frac{1}{N}\sum_{i=1}^{N} X_i \not\in [p - \delta, p + \delta],$$

where the X_i denote N mutually independent random variables defined by $Pr[X_i = 1] = p$ and $Pr[X_i = 0] = 1 - p$. Höffdings Inequality (see, e.g., [1], Appendix A) yields that

Lemma 1. *The probability that $T(p, \delta, N)$ accepts is smaller than $2e^{-2\delta^2 N}$.* □

Note that most of our distinguishing scheme will be tests T which first choose a random seed r from an appropriate set R, and then perform a corresponding test $T(r)$ on the oracle function. Such a test T is called a (p, q, ρ)-**test for a function** $f^* \in B_n$ if T accepts a random function with probability at most ρ (i.e., $\mathbf{E}_{r\in R}[Pr_{f\in B_n}[T(r) \text{ accepts } f]] \leq \rho$), but if the probability (taken over r) that $T(r)$ accepts $f^* \in F_n$ with probability at least q, is at least p.

Observe the following easy but useful fact.

Lemma 2. *If $pq > \rho$ then a (p, q, ρ)-test for f^* distinguishes f^* with advantage at least $pq - \rho$ from a truly random function.* □

Theorem 1. *There is a polynomial distinguishing scheme for polynomial size weighted threshold-MOD_2 circuits.*

Proof. The algorithm follows quite straightforwardly from a result from *Bruck* [6]. If m is the minimal number of MOD$_2$-nodes in a weighted threshold-MOD$_2$-circuit computing a given $f \in B_n$ then there is a MOD$_2$-function $p(x) = x_{i_1} \oplus \ldots \oplus x_{i_r}$ in B_n such that

$$\left|\mathbf{E}_{x\in\{0,1\}^n}[f \oplus p(x)] - \frac{1}{2}\right| \geq \frac{1}{2m}.$$

Let us fix a polynomial bound $m(n) \in n^{O(1)}$. Let the scheme D work as follows on n and $m = m(n)$. It chooses an appropriate number $\tilde{n}, \log(m) < \tilde{n} < n$, chooses a random MOD$_2$-function $\tilde{p}(x)$ over $\{x_1, \ldots, x_{\tilde{n}}\}$ and accepts if

$$\left|\mathbf{E}_{x\in\{0,1\}^{\tilde{n}}}[f(x, \vec{0}) \oplus \tilde{p}(x)] - \frac{1}{2}\right| \geq \frac{1}{4m}.$$

Observe that the running time is linear in $N = 2^{\tilde{n}}$ and that this test is a $(1/N, 1, 2e^{-2\frac{1}{16m^2}N})$-test on each function $f^* \in B_n$ having weighted threshold-MOD_2 circuits of size m. (Observe the above mentioned result [6] and the fact that the subfunction $f(\cdot, \vec{0})$ has size $\leq m$.) It is easy to see that we can find some $\tilde{n} \in O(\log(n))$ yielding advantage $\frac{1}{2N}$ (see Lemma 2). □

Theorem 2. *For all primes p and all constant depth bounds d there is a quasipolynomial distinguishing scheme for polynomial size depth d circuits over $\{AND, OR, MOD_p\}$.*

The proof is quite lengthy and can be found in the full paper [14]. As MOD_{p^k} belongs to $AC_2^0[p]$ [29], the proof for prime powers follows immediately.

Theorem 3. *For all $k \geq 1$ there is a quasipolynomial distinguishing scheme for nondeterministic read–k BDDs.*

Proof. The first exponential lower bounds on read k branching programs were independently proved in [5] and [26]. See also [12] for other interesting applications of the method. We use these methods for our distinguishing scheme. Let us fix an arbitrary natural constant $k \geq 1$, and a polynomial bound $m = m(n) \in n^{O(1)}$. Let us denote $X_n = \{x_1, \ldots, x_n\}$. In [12] *Jukna* shows the existence of a number $s \in m^{O(1)} = n^{O(1)}$ and a constant $\gamma \in (0,1)$ such that each $f \in B_n$ which is computable by a nondeterministic syntactic read–k times branching program of size $m(n)$ can be written as

$$f = \bigvee_{i=1}^{W} f_i, \tag{2}$$

where for all $i,\ 1 \leq i \leq W$, it holds that there is a partition $X_n = U_i \cup V_i \cup W_i$ of pairwise disjoint subsets U_i, V_i, W_i of X_n such that

$$f_i(X_n) = g_i(U_i, V_i) \wedge h_i(V_i, W_i),$$

where $|U_i| \geq \gamma n$ and $|W_i| \geq \gamma n$.

The distinguishing scheme D works on n and $m = m(n)$ as follows.

(0) Fix an appropriate $N \in n^{O(1)}$ and test via $T(\frac{1}{2}, \frac{1}{12}, N)$ if the probability that the oracle function outputs 1 is at least $\frac{1}{3}$. If not accept.

(1) Compute s and appropriate parameters $q, r \in \log^{O(1)} n$. Let $Q = 2^q$. Choose randomly disjoint subsets U, W from X_n with $|U| = |W| = q$, and a $\{0,1\}$-assignment b of $X \setminus (U \cup W)$. Finally, choose random $\{0,1\}$-assignments $a^1, \ldots, a^r$ of U.

(2) Accept iff $f(a^1, b, c) \wedge \ldots \wedge f(a^r, b, c) = 1$ for at least $\frac{Q}{6s}$ assignments c of W.

The parameters q, N, and r will be specified later. Observe that the running time is $O(rQ)$. Observe further that the probability that a truly random function will be accepted in Step 2 is bounded by $2e^{-2\delta^2 Q}$ for $\delta = \frac{1}{6s} - 2^{-r}$ (see (1)).

On the other hand, in the pseudorandom case it holds with probability $\frac{1}{s}(\gamma/2)^{2q}$ that $U \subseteq U_j$ and $W \subseteq W_j$ for some j for which $Pr_x[f_j(x) = 1] \geq \frac{1}{3s}$. Further, with probability $\frac{1}{2s}(\gamma/2)^{2q}$ we have b fixed in such a way that $Pr_{a,c}[f_j(a,b,c) = 1] \geq \frac{1}{6s}$, where a and c denote the assignments of U and W respectively. Observe that this implies that $Pr_a[g_j(a,b) = 1] \geq \frac{1}{6s}$ and that $Pr_c[h_j(b,c) = 1] \geq \frac{1}{6s}$. Consequently, with probability $p = \frac{1}{6s}^{r} \frac{1}{2s}(\gamma/2)^{2q}$ it holds that $g_j(a^1, b) = \ldots = g_j(a^r, b) = 1$. But, under this condition, it holds for all assignments c to W and $l, 1 \leq l \leq r$, that $f_j(a_l, b, c) = 1$ iff $h_j(b,c) = 1$ iff $f_j(a_i, b, c) = 1$ for all $l, 1 \leq l \leq r$. As $f_j(a_i, b, c) = 1$ implies $f(a_i, b, c) = 1$, the function is accepted in Step 2 with probability 1.

We obtain that Step 1 and 2 form a $(p, 1, 2e^{-2\delta^2 Q})$-test for each function f of size at most m. It can be easily verified that for $q = \lfloor \log_2(s^2 n) \rfloor$ and $r = \lfloor \log_2(12s) \rfloor$, we can find some $N \in n^{O(1)}$ such that $D(n,m)$ achieves advantage $\varepsilon(n,m)$ fulfilling $\varepsilon(n,m)^{-1} \in n^{O(\log n)}$. □

Theorem 4. *There is a polynomial distinguishing scheme for polynomial size unweighted depth 2 threshold circuits.*

Proof. For all distributed functions $f : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}$ consider the following invariants

$$\gamma(f) = \max\left\{ \left| \mathbf{E}_{x,y}[f(x,y) \oplus g(x) \oplus h(y)] - \frac{1}{2} \right| ;\ g, h \in B_n \right\}$$

$$\alpha(f) = \max\left\{ \left| \mathbf{E}_y[f(x,y) \oplus f(x',y)] - \frac{1}{2} \right| ; \; x \neq x' \in \{0,1\}^n \right\}.$$

The first exponential lower bound on the size of unweighted depth 2 threshold circuits was proved in [10]. The following two observations are implicitely contained there. Let us fix an arbitrary polynomial bound $m = m(n) \in n^{O(1)}$.

(I) There is a number $S \in m^{O(1)}$ such that if $f : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}$ has unweighted depth 2 threshold circuits of size $m(n)$ then $\gamma(f) \geq \frac{1}{S}$.
(II) For all distributed functions $f : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}$ it holds that $\gamma(f) \leq \sqrt{\frac{1}{2}(\alpha(f) + 2^{-n})}$.

The distinguishing scheme $D = D(n,m)$ is defined to do the following on n and m. It chooses an appropriate number $q \in O(\log(n))$ such that for $Q = 2^q$ the condition $Q \geq S^2$ is satisfied, and two random assignments $x \neq x'$ of $\{x_1, \ldots, x_q\}$. D accepts if

$$|\mathbf{E}_{y \in \{0,1\}^q}[f(x,y,\vec{0}) \oplus f(x',y,\vec{0})] - \frac{1}{2}| \geq \frac{1}{2S^2}.$$

Observe that the probability that this test accepts a truly random function is the same as the probability that test $T(\frac{1}{2}, \frac{1}{2S^2}, Q)$ accepts, i.e., at most $2e^{-Q/S^2}$.

On the other hand, observe that for all oracle functions of size $\leq m$ the following holds: if in Step 1 the pair x, x' determining $\alpha(f(\cdot,\cdot,\vec{0}))$ is chosen (and this occurs with probability $1/(Q(Q-1)))$ then Step 2 will accept with probability 1. In other words, we have a $(1/(Q(Q-1)), 1, 2e^{-Q/S^2})$-test. It is quite easy to verify that we can fix some $q \in O(\log(n))$ which gives advantage $\varepsilon(n,m)$ for $D(n,m)$ fulfilling that $\varepsilon^{-1}(n,m) \in n^{O(1)}$. □

Theorem 5. *For all $k \geq 1$ it holds that there is a distinguishing algorithm of quasipolynomially bounded ratio for depth $k+1$ circuits consisting of k levels of AND and OR gates connected with one weighted threshold gate as output gate.*

The proof exhibits the so called Switching Lemma [11] and can be found in the full paper [14].

4 Pseudorandom TC_4^0-Functions

We start with the definition of the NR-generator F. For all n the keys s for F have the form $s = (P, Q, g, r, a_1, \ldots, a_n)$, where all components are n-bit numbers fulfilling the following conditions. P and Q are primes and Q divides $P-1$, $g \in \mathbb{Z}_P^*$ has multiplicative order Q, and $a_1, \ldots, a_n$ are from $\mathbb{Z}_Q^*$. Define the corresponding function $f_s : \{0,1\}^n \to \mathbb{Z}_P \subseteq \{0,1\}^n$ by

$$f_s(x) = f_s(x_1, \ldots, x_n) = g^{y(x)} \bmod P,$$

where $y(x) = \prod_{i=1}^n a_i^{x_i}$. For our purpose it is obviously sufficient to show

Theorem 6. *The function $f = f_s$ has polynomial size depth 4 unweighted threshold circuits.*

Proof. We use the following terminology and facts about threshold circuits which are mainly based on results from [8, 9, 28].

Definition 1. *A Boolean function $g : \{0,1\}^n \longrightarrow \{0,1\}$ is called t-bounded if there are integer weights $w_1, \ldots, w_n$ and t pairwise disjoint intervals $[a_k, b_k]$, $1 \leq k \leq t$ of the real line such that*

$$g(x_1, \ldots, x_n) = 1 \quad \Longleftrightarrow \quad \exists k \text{ s.t. } \sum_{i=1}^{n} w_i x_i \ \in \ [a_k, b_k].$$

The function g is called polynomially bounded if g is t-bounded for some $t \in n^{O(1)}$. A multi-output function is called t-bounded if each output bit is a t-bounded Boolean function.

Fact 1: Suppose that a function $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ can be computed by a depth d circuit of polynomial size, where each gate of the circuit performs a function which can be written as a sum of at most $s \in n^{O(1)}$ polynomially bounded operations. Then f can be computed by a polynomial size depth $d+1$ unbounded weight threshold circuit.

Observe the following statements which can be easily proved.

Fact 2: If $g(x_1, \ldots, x_n)$ depends only on a linear combination $\sum_{i=1}^{n} w_i x_i$, where for all i, $1 \leq i \leq n$, it holds $|w_i| \in n^{O(1)}$, then g is a polynomially bounded operation.

Fact 3: If a Boolean function $g : \{0,1\}^n \longrightarrow \{0,1\}$ can be written as $g = h(g_1, \ldots, g_c)$, where c is a constant and the Boolean functions $g_1, \ldots, g_c : \{0,1\}^n \longrightarrow \{0,1\}$ are polynomially bounded operations, then g is a polynomially bounded operation.

As for many other efficient threshold circuit constructions, the key idea is to parallelize the computation of $f(x)$ via Chinese remaindering. Let us fix the first r prime numbers $p_1, \ldots, p_r$, where r is the smallest number such that $\varPi := \prod_{1 \leq k \leq r} p_k \geq \prod_{i=1}^{n} a_i$. Observe that $r \in O(n^2)$ and that all p_i, $1 \leq i \leq r$, are polynomially bounded in n, i.e., can be written as m-bit numbers for some $m \in O(\log n)$.

Consider the inverse Chinese remaindering transformation CRT^{-1} which assigns to each r-tupel of m bit numbers $(z^1, \ldots, z^r)$, $z^i = (z^i_{m-1}, \ldots, z^i_0)$ for $i = 1, \ldots, r$, the uniquely defined number $y < \varPi$ for which $y \equiv z^i \mod p_i$ for all $i = 1, \ldots, r$. Denote by CRT_P^{-1} the function

$$CRT_P^{-1} : (\{0,1\}^m)^r \longrightarrow \{0,1\}^{n^2}$$

defined as $\big(CRT^{-1}(z^1, \ldots, z^r) \mod P\big)$, and observe

Fact 4: CRT_P^{-1} can be written as the sum of polynomially (in n) many polynomially bounded operations.

The proof (see, e.g., [28]) is based on the fact that

$$CRT^{-1}(z^1, \ldots, z^r) = \sum_{i=1}^{r} E_i z^i \mod \varPi,$$

where for $i = 1 \ldots r$ the number E_i denotes the uniquely determined number smaller than $\varPi$ for which $(E_i \mod p_j) = \delta_{i,j}$ for all $i, j = 1, \ldots, r$. This implies

$$\begin{aligned} CRT^{-1}(z^1, \ldots, z^r) &= \sum_{i=1}^{r} E_i \left(\sum_{j=0}^{m-1} z^i_j 2^j \right) \mod \varPi \\ &= \sum_{i=1}^{r} \sum_{j=0}^{m-1} e_{i,j} z^i_j \mod \varPi, \end{aligned} \tag{3}$$

where $e_{i,j} = (E_i 2^j \mod \varPi)$.

The computation of $f(x)$ will be performed on 3 consecutive levels consisting of operations which are polynomially bounded (level 1,2) or which can written as polynomial length sums of polynomially bounded operations.

Level 1: Compute $z(x) = (z^1(x), \ldots, z^r(x))$, where for all $i = 1, \ldots, r$, the m-bit number z^i is defined to be $(y(x) \bmod p_i)$.

Observe that for all $i = 1, \ldots, r$, $z^i(x)$ can be written as

$$z^i(x) = \prod_{j=1}^{n} a_j^{x_j} \bmod p_i = \alpha_i^{\sum_{j=1}^{n} r_j^i x_j} \bmod p_i,$$

where α_i denotes a fixed element of order $p_i - 1$ in $\mathbb{Z}^*_{p_i}$ and r_j^i denotes for $j = 1, \ldots, n$ the discrete logarithm of a_j to the base α_i. Because all r_j^i are polynomially bounded in n, it follows by Fact 2 that $z(x)$ is a polynomially bounded operation.

For all inputs $z = (z^1, \ldots, z^r) \in (\{0,1\}^m)^r$ denote by $Y(z)$ the number

$$Y(z) = \sum_{i=1}^{r} \sum_{k=0}^{m-1} e_k^i z_k^i.$$

Observe that for all x it holds that $y(x) \equiv Y(z(x)) \bmod \Pi$ and $Y(z(x)) \leq mr\Pi$. Moreover, there exists exactly one k, $1 \leq k \leq mr - 1$, such that

$$y(x) = Y(z(x)) - k\Pi.$$

This k is characterized by $k\Pi \leq Y(z(x)) \leq (k+1)\Pi - 1$. Consequently, the equation $f = f_0 + \ldots + f_{mr-1}$ holds, where for each $k = 0, \ldots, mr - 1$, the function f_k is defined as

$$f_k(x) = \chi_k(z(x))(g^{Y(z(x)) - k\Pi} \bmod P),$$

where $\chi_k(z(x)) \in \{0,1\}$ is defined by $\chi_k(z(x)) = 1$ iff $k\Pi \leq Y(z(x)) \leq (k+1)\Pi - 1$.

Further observe that

$$g^{Y(z) - k\Pi} \bmod P \ = \ G_k(z) \bmod P,$$

where $G_k(z) = c_k \prod_{i=1}^{r} \prod_{j=0}^{m} (b_{i,j})^{z_j^i}$, and the c_k and $b_{i,j}$ are n-bit numbers defined by

$$c_k = (g^{-k\Pi} \bmod P) \quad \text{and} \quad b_{i,j} = (g^{e_{i,j}} \bmod P).$$

Observe that, in contrast to $g^{Y(z)-k\Pi}$, the number $G_k(z)$ has polynomially many, namely $n(mr+1)$, bits. Fix u to be the smallest number such that $\prod_{i=1}^{u} p_i \geq 2^{n(mr+1)}$. Observe further that by the same arguments as above (Level 1), the operation $(G_k(z) \bmod p_i)$ is for all $i = 1, \ldots, u$ polynomially bounded.

Level 2: For all $k = 0 \ldots mr - 1$ and $i = 1 \ldots u$ compute

$$H_k^i(z) = \chi_k(z)(G_k(z) \bmod p_i).$$

This is a polynomially bounded operation as each output bit depends only on two polynomially bounded operations (Fact 3).

Level 3: Compute $f_k(x) = CRT_P^{-1}(H_k^1(z(x)), \ldots, H_k^u(z(x)))$.

Due to Fact 4 and Fact 1 this yields polynomial size depth 4 unweighted threshold circuits for f. □

5 Open Problems

It would be nice if we could detect for each basic nonuniform complexity class $\Lambda = P(M)$ whether it has an efficient distinguishing scheme (then cryptodesigners should obey the low complexity danger w.r.t. M) or whether Λ contains a PRFG (then lower bound proofs for this model seem to be a very serious task). Unfortunately, there are classes like TC_3^0 and $AC_3^0[m]$, m composite, which up to now cannot be classified in the above way. It is an interesting open question if TC_3^0 is strong enough to contain PRFGs. Observe that TC_3^0 seems to contain pseudorandom bit generators. (Note that operations such as squaring modulo the product of two unknown primes is in TC_3^0 [28].)

Another open problem is the design of an efficient distinguishing scheme for polynomial size weighted threshold-MOD$_p$ circuits, p an odd prime power. This is the only example of a complexity measure for which we failed to transform the known effective lower bound method (see [15]) into a distinguishing algorithm.

A further interesting question is to determine the minimal hardware complexity of other cryptographic primitives like pseudorandom bit generators, pseudorandom permutation generators, one-way functions and cryptographically secure hash functions. Does TC_2^0 contain pseudorandom bit generators? Luby and Rackoff [20] showed how to construct *pseudorandom permutations* by three sequential applications of a pseudorandom function, each followed by an XOR-operation. Luby and Rackoff also showed how to construct *super pseudorandom permutations* by four such applications. Thus, as a corollary of our results, efficient pseudorandom permutations can be constructed in TC$_{10}^0$ and efficient super pseudorandom permutations can be constructed in TC$_{13}^0$. We conjecture that these results can be further improved, perhaps based on the results from [25].

References

1. N. Alon, J. Spencer, P. Erdös. The probabilistic method. Wiley & Sons 1992.
2. M. Bellare, S. Goldwasser. New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. Crypto '89, Springer LNCS, pp. 194–211, 1990.
3. M. Blaze, J. Feigenbaum, M. Naor. A Formal Treatment of Remotely Keyed Encryption. Eurocrypt '98, Springer LNCS, 1998.
4. A. Blum, M. Furst, M. Kearns, R.J. Lipton. Cryptographic primitives based on hard learning problems. Proc. CRYPTO 93, LNCS 773, 278-291.
5. A. Borodin, A. Razborov, R. Smolensky. On lower bounds for read k times branching programs. J. Computational Complexity 3, 1993, 1-13.
6. J. Bruck. Harmonic Analysis of polynomial threshold functions. SIAM Journal of Discrete Mathematics. 3:22, 1990, pp. 168-177.
7. O. Goldreich, S. Goldwasser, S. Micali. How to construct random functions. J. of the ACM, vol 33, pp. 792–807, 1986.
8. M. Goldmann, J. Hastad, A. A. Razborov. Majority gates versus general weighted Threshold gates. J. Computational Complexity 2, 1992, 277-300.
9. M. Goldmann, M. Karpinski. Simulating threshold circuits by majority circuits. Proc. 25th ACM Symp. on Theory of Computing (STOC), 1993, 551-560.
10. A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, G. Turan. Threshold circuits of bounded depth. FOCS'87, pp. 99-110.
11. J. Hastad. Almost optimal lower bounds for small depth circuits. STOC'86, pp. 6-20.
12. S. Jukna. A note on read-k time branching programs. Theoretical Informatics and Applications 29(1), 1995, 75-83.
13. M. Kearns, L. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. J. of the ACM, vol. 41(1), 1994, pp. 67-95.
14. M. Krause, S. Lucks. On the minimal Hardware Complexity of Pseudorandom Function Generators. "http://th.informatik.uni-mannheim.de/research/research.html".
15. M. Krause, P. Pudlak. On the computational power of depth-2 circuits with threshold and modulo gates. J. Theoretical Computer Science 174, 1997, pp. 137-156. Prel. version in STOC'94, pp. 49-59.

16. M. Krause, P. Pudlak. Computing Boolean functions by polynomials and threshold circuits. J. Comput. complex. 7 (1998), pp. 346-370. Prel. version in FOCS'95, pp. 682-691.
17. M. Krause, P. Savicky, I. Wegener. Approximation by OBDDs, and the variable ordering problem. Lect. Notes Comp. Science 1644, Proc. of ICALP'99, pp. 493-502.
18. M. Krause, S. Waack. Variation ranks of communication matrices and lower bounds for depth two circuits having symmetric gates with unbounded fan-in. J. Mathematical System Theory 28, 1995, 553–564.
19. N. Linial, Y. Mansour, N. Nisan. Constant depth circuits, Fourier transform, and learnability. J. of the ACM, vol. 40(3), 1993, pp. 607-620. Prel. version in FOCS'89, pp. 574-579.
20. M. Luby, C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. SIAM J. Computing, Vol. 17, No. 2, pp. 373–386, 1988.
21. S. Lucks. Faster Luby-Rackoff Ciphers. Fast Software Encryption 1996, Springer LNCS 1039, 189–203, 1996.
22. S. Lucks. On the Security of Remotely Keyed Encryption. Fast Software Encryption 1997, Springer LNCS 1267, 219–229, 1997.
23. M. Naor, O. Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. Proc. 36th IEEE Symp. on Foundations of Computer Science, pp. 170–181, 1995.
24. M. Naor, O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. Preliminary Version. Proc. 38th IEEE Symp. on Foundations of Computer Science, 1997.
25. M. Naor, O. Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. J. of Cryptology, Vol. 12, No 1, 29–66, 1999.
26. E. Okolshnikova. On lower bounds for branching programs. Siberian Advances in Mathematics 3(1), 1993, 152-166.
27. A. Razborov, S. Rudich. Natural Proofs. J. of Computer and System Science, vol. 55(1), 1997, pp. 24-35. Prel. version STOC '94, pp. 204-213.
28. K. Siu, J. Bruck, T. Kailath, T. Hofmeister. Depth efficient neural networks for division and related problems. IEEE Trans. of Inform. Theory, vol. 39, 1993, pp. 946-956
29. R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. STOC'87, pp. 77-82.
30. I. Wegener. The complexity of Boolean functions. John Wiley & Sons, 1987.

6 Appendix

6.1 The Proof of Theorem 2

Theorem 2 *For all primes p and all constant depth bounds d there is quasipolynomial distinguishing scheme for polynomial size depth d circuits over $\{AND, OR, MOD_p\}$.*
Proof. We start with some preliminaries: Let K denote an arbitrary field and $B = \{a, b\}$ an arbitrary two-element subset of K. Observe that each function $h : B^n \longrightarrow K$ has a unique representation as an n–variate multilinear polynomial over K. Let us denote by $\deg_K(h)$ the degree of this representation, i.e., the maximal length of a monomial occuring with nonzero coefficient in this representation. Fix a Boolean function $f : \{0, 1\}^n \longrightarrow \{0, 1\}$. The unique function $\hat{f} : B^n \longrightarrow B$, which is obtained from f by replacing all occurences of 0 by a and of 1 by b is said to be the (a, b)-variant of f. Now fix another two elements $a' \neq b'$ of K and denote by g the (a', b')-variant of f. Observe that for all $(y_1, \dots, y_n) \in \{a', b'\}^n$ the relation

$$g(y_1, \dots, y_n) = \frac{a' - b'}{a - b}\hat{f}(x_1, \dots, x_n) + \frac{ab' - a'b}{a - b} \tag{4}$$

holds, where $x_i = \frac{a-b}{a'-b'}y_i + \frac{a'b-ab'}{a'-b'} \in \{a, b\}$ for all $i = 1, \dots, n$. As this transformation is linear it follows that for all two elements $a \neq b \in K$ it holds that the K-degree of the (a,b)-variant of f is the same. We denote this value by $\deg_K(f)$.

For $K = \mathbf{F}_r$, $r = p^k$ prime power, we use the denotation $\deg_r(f)$. If the context is clear and some field K is fixed we identify Boolean functions with their $(0_K, 1_K)$-variants. We start now with the proof of Theorem 2.

Theorem 2 *For all primes p and all constant depth bounds d there is a uasipolynomial distinguishing scheme for polynomial size depth d circuits over AND, OR, MOD_p-gates.*

Let us fix a prime p and a depth bound d. The proof of the Theorem is based on the following result of *Smolensky* [29]:

Lemma 3. *Let $f, g_1, \dots, g_k \in B_n$ be given such that $f = \bigvee_{i=1}^k g_i$. Then for all $r < n$ there is a $\mathbf{F}_p$-polynomial $q = q(g_1, \dots, g_m)$ of degree at most $(p-1)r$ such that $Pr_x[f(x) \neq q(g_1(x), \dots, g_m(x))] \leq 2^{-r}$. The same statement holds if $f = \bigwedge_{i=1}^k g_i$.*

It is quite straightforward to derive

Corollary 1. *If $f \in B_n$ can be computed by a depth d AND,OR,MOD$_p$-circuit of size m then for each r, $p \leq r < n$, there is a function $\tilde{f} : \{0, 1\}^n \longrightarrow \mathbf{F}_p$ such that $\deg_p(f) \leq ((p-1)r)^d$ and $Pr_x[f(x) \neq \tilde{f}(x)] \leq ((m^d - 1)/(m - 1))2^{-r}$.*

Proof. The approximating function $\tilde{f}$ is obtained by replacing all AND- and OR- gates by $\mathbf{F}_p$-polynomials which approximate the gate with parameter r as in Lemma 3. Taking into account that the $\mathbf{F}_p$-degree of MOD$_p$ is p-1 and that the indegree of each AND- and OR-gate is bounded by m it is easy to see that the degree of $\tilde{f}$ is bounded by $\delta_d(m)$ and the error probability is bounded by $E_d(m)$, where $\delta_d(m)$ and $E_d(m)$ are defined via the recursion $\delta_1(m) = (p-1)r$, $E_1(m) = 2^{-r}$, $\delta_d(m) = (p-1)r\delta_{d-1}(m)$ and $E_d(m) = mE_{d-1}(m) + E_1(m)$. Evaluating this recursion gives the claim. □

Consequently, distinguishing $AC^0_d[p]$-functions from truly random functions can be reduced to testing that a given sample is induced by a function which can be well approximated by a low degree polynomial over $\mathbf{F}_p$. If $p \neq 2$ the idea for such a test can be derived from *Razborov's* and *Rudich's* Natural Proof against AC^0[3] [27]: Let us fix some odd number n. In the following, we do all arithmetic operations with respect to the field $\mathbf{F}_p$. For all Boolean functions $f : \{0, 1\}^n \longrightarrow \{0, 1\}$ we denote by $\hat{f}$ the (1,-1)-variant of f. As the characteristic of $\mathbf{F}_p$ is odd we have $1 \neq -1$.

Let us denote by V the $\mathbf{F}_p$-vector space of all functions $h : \{1,-1\}^n \longrightarrow \mathbf{F}_p$. It holds $\dim_p(V) = N := 2^n$. We denote further by L the subspace of all $h \in V$ with $\deg_p(h) < n/2$. As n is odd we have $\dim_p(L) = N/2$. The complexity parameter $D_p(\hat{f})$ which is essential for us is defined as

$$D_p(f) = \dim_p(L + \hat{f}L),$$

where $\hat{f}L$ denotes the subspace of functions which can be written as $\hat{f} \cdot h,\ h \in L$, where $\cdot$ denotes argumentwise multiplication. (Observe that the set of functions $\hat{f} : \{1,-1\}^n \longrightarrow \{1,-1\}$ is closed under argumentwise multiplication.)

Observe the following properties of the parameter D:

(i) If f coincides with a function $g : \{0,1\}^n \longrightarrow \mathbf{F}_p$ of degree $P \leq \gamma\sqrt{n},\ \gamma \in (0,1)$, outside a fixed input set $E \subseteq \{0,1\}^n$ then $D_p(f) \leq (1/2 + \gamma)N + |E|$.
In order to see this observe at first that there is a function $\hat{g} : \{1,-1\}^n \longrightarrow \mathbf{F}_p$ with degree P which coincides with $\hat{f}$ outside a fixed input set $E' \subseteq \{1,-1\}^n$, where $|E| = |E'|$.
Consequently, outside of E' all functions in $L + \hat{f}L$ coincide with a function of degree smaller than $n/2 + P$. Hence,

$$D_p(f) \leq \sum_{k=0}^{n/2+P} \binom{n}{k} + |E| \ \leq\ N(1/2 + P/\sqrt{n}) + |E|.$$

(The last calculation is a consequence of Stirling's Formula which gives that $\binom{n}{\lfloor n/2 \rfloor} \leq 2^n/\sqrt{n}$.)

(ii) For the parity function $\pi = x_1 \oplus \ldots \oplus x_n$ it holds that $D_p(\pi) = N$. This follows from the well-known fact that $\hat{\pi} = y_1 y_2 \ldots y_n$. Consequently, (over $\{1,-1\}^n$) for each monomial m of degree larger than $n/2$ there is a monomial m' of degree smaller than $n/2$ such that $m = \hat{\pi}m'$.

(iii) For all Boolean functions f it holds that $D_p(f) + D_p(\pi \oplus f) \geq 3/2N$. In order to see this observe that

$$D_p(\pi \oplus f) - N/2 \ =\ \dim_p(L + \hat{\pi}\hat{f}L/L) \ =$$

$$\dim_p(\hat{f}L + \hat{\pi}L/\hat{f}L) \ \leq\ \dim_p(\hat{f}L + \hat{\pi}L + L/(\hat{f}L + L)) \ =$$

$$\dim_p(V/(L + \hat{f}L)) \ =\ N - D_p(f).$$

The statement follows directly. As a consequence of (3) we obtain:

(iv) The amount of Boolean functions $f : \{0,1\}^n \longrightarrow \{0,1\}$ with $D_p(f) \geq 3/4N$ is at least 50%.

(v) In order to evaluate $D_p(f)$, one has to compute the $\mathbf{F}_p$-rank of an $N \times N$-matrix, i.e., it can be done in time $N^{O(1)}$.

We describe now the distinguishing algorithm D for $\{AND, OR, MOD_p\}$-circuits, where $p \neq 2$. Fix a polynomial $m = m(n) \in n^{O(1)}$. Given input parameters n and $m = m(n)$, D at first computes the minimal number r and the minimal odd number $\tilde{n}$ such that

$$64m^{d-1} \ <\ 2^r \quad \text{and} \quad (p-1)^d r^d \ <\ (1/8)\sqrt{\tilde{n}}.$$

Observe that $r \in O(\log(n))$, $n \in O(\log^{2d}(n))$ and let $\tilde{N} = 2^{\tilde{n}}$.

Then D chooses randomly an 0,1-assignment c to the set of variables $\{x_{\tilde{n}+1}, \ldots, x_n\}$ and accepts if $D_p(f^c) < (3/4)\tilde{N}$.

Observe that by (v), this computation can be done using $\tilde{N} := 2^{\tilde{n}}$ oracle queries in time $\tilde{N}^{O(1)} = \exp(\log^{O(1)} n)$.

In the truly random case, by (iv), the probability that D outputs 1 is at most 1/2.

Now consider the pseudorandom case and denote by f the secret function chosen by the oracle. By Corollary 1, there is a function $\tilde{f} : \{0,1\}^n \longrightarrow \mathbf{F}_p$ such that $\deg_p(\tilde{f}) \le ((p-1)r)^d$ such that the probability that f differs from $\tilde{f}$ is bounded by $((m^d-1)/(m-1))2^{-r}$.

Observe that for at least 75% of the 0,1-assignments c to the variables $\{x_{\tilde{n}+1}, \ldots, x_n\}$ it holds that the probability that f^c differs from $\tilde{f}^c$ is bounded by

$$4((m^d-1)/(m-1))2^{-r} \; < \; 8m^{d-1}2^{-r}. \tag{5}$$

This implies that f^c differs from $\tilde{f}^c$ on a set E of less than $8m^{d-1}2^{\tilde{n}-r} \;\le\; (1/8)2^{\tilde{n}}$ inputs, i.e., by (i) and as $\deg_p(\tilde{f}) < (1/8)\sqrt{\tilde{n}}$ we obtain

$$D_p(\tilde{f}) \; < \; (1/2+1/8)\tilde{N} + (1/8)\tilde{N} \; = \; (3/4)\tilde{N}.$$

Consequently, the probability that D accepts is at least 3/4. It follows directly that D distinguishes $AC_d^0[p]$-functions from truly random functions with quasipolynomially bounded ratio.

Now let us consider the case $p = 2$. Clearly, if a given Boolean function f coincides outside a set E with a function g with $\deg_2(g) = d$, then for all fields K of characteristic 2 and all $a \neq b \in K$ it holds that the (a,b)-variant of f coincides with a function $\hat{g}$ of K-degree d outside a set $\hat{E}$ with $|E| = |\hat{E}|$.

The problem is that $1 = -1$ holds for fields of characteristic 2.

We choose the field $K = \mathbf{F}_4 = \{0, 1, z, z+1\}$. Observe the relation $z^2 = z+1$ and the fact that $k^3 = 1$ for all $k \in \{1, z, z+1\}$. For a Boolean function f we denote by $\hat{f}$ the (1,z)-variant of f. As above, we fix an odd n, denote $N = 2^n$, denote by V the N-dimensional K-vector space of all functions from $\{1,z\}^n$ into K, and by L the $N/2$-dimensional subspace of all functions of K-degree smaller than $n/2$.

Further let for all functions $h : \{1,z\}^n \longrightarrow \{1, z, z+1\}$

$$D_2(h) = \dim_K(L + \hat{f}L).$$

For Boolean functions $f : \{0,1\}^n \longrightarrow \{0,1\}$ let $D_2(f) := D_2(\hat{f})$. Observe that property (i) of D_p holds in the same way for D_2. Consider further the function $\rho : \{1,z\}^n \longrightarrow \{1, z, z+1\}$ defined by

$$\rho(y_1, \ldots, y_n) = y_1 y_2 \ldots y_n.$$

Observe now the following properties of D_2:

(I) It holds that $\dim_K(L + \rho^2 L) = N$. In order to prove this it is sufficient to show that each monomial m of length larger than n/2 belongs to $\rho^2 L$. We can obviously find a monomial m' of length smaller n/2 such that $m^2 = \rho^2 m'$. On the other hand, using the fact that on $\{1,z\}$

$$y_i^2 = (z+1)y_i + z$$

it can be seen that $m^2 = (z+1)^t m + h$, where t denotes the length of m and h a function of degree smaller than t. Induction on the length of m yields the proof.

(II) The amount of functions $h : \{1,z\}^n \longrightarrow \{1, z, z+1\}$ for which $D_2(h) \ge (3/4)N$ is at least 50%. For proving this observe that for all $h : \{1,z\}^n \longrightarrow \{1, z, z+1\}$

$$D_2(\rho^2 h) - N/2 = dim_K(L + \rho^2 hL/L) = dim_K(h^2 L + \rho^2 L/h^2 L)$$

$$\ge dim_K(h^2 L + \rho^2 L + L/(h^2 L + L)) \; = \; N - D_2(h^2),$$

i.e., $D_2(\rho^2 h) + D_2(h^2) \ge (3/2)N$. As squaring and multiplication with ρ^2 are bijective mappings over the set of functions $h : \{1,z\}^n \longrightarrow \{1, z, z+1\}$ the claim follows.

In other words, if we take a truly random function $h : \{1, z\}^n \longrightarrow \{1, z, z+1\}$ then $D_2(h) \geq (3/4)N$ with significant probability. Unfortunately, we can not show this for the (1,z)-variants of random *Boolean* functions which would be necessary for our distinguishing algorithm. This is because we do not see any way for applying the above distinguishing algorithm straightforwardly in the case $p = 2$. The only way-out we see in the moment is to use the following (allmost complexity preserving) transformation of functions $f \in B_n$ into functions which map into $\{1, z, z+1\}$. We describe the transformation in a more general form which could also be usefull in other similar situations.

Generating random functions into $\{1, \ldots, k\},\ k > 2$

We describe here an operator $T_{n,k,m}$, where n, k, m are positiv natural numbers fulfilling $m < n$ and $k < 2^n$, which assigns to each Boolean function $f : \{0,1\}^n \longrightarrow \{0,1\}$ a k-nary function $T_{n,k,m}(f) : \{0,1\}^m \longrightarrow \{0,1\}$ such that the following holds:

– If f has low complexity w.r.t. to a large number of relevant nonuniform complexity measures then $T_{n,k,m}(f)$ has, too.

– If f is a random Boolean function then, for s large enough, $T_{n,k,m}$ looks ”sufficiently random”. The construction is based on the following technical

Lemma 4. *For each n and $k \leq 2^n$, and each partition $\pi = (s_1, \ldots, s_k)$ of 2^n, i.e., the s_i are positive natural numbers fulfilling $s_1 + \ldots + s_k = 2^n$, there is a function $h_\pi : \{0,1\}^n \longrightarrow \{0,1\}$ with the following properties:*

(a) For all $i,\ 1 \leq i \leq k$, it holds $|h_\pi^{-1}(i)| = s_i$.

(b) h has a Boolean decision tree with at most $(k-1)n+1$ leafs.

Proof. A decision tree for a function $h : \{0,1\}^n \longrightarrow \{1, \ldots, k\}$ is a usual Boolean decision tree for which the leafs are labelled by $1, \ldots, k$. The computation mode is straightforward. We identify partitions $2^n = s_1 + \ldots + s_k$ by multisets $\pi = (s_1, \ldots, s_k)$. For each n and $k \leq 2^n$, and each partition $\pi = (s_1, \ldots, s_k)$ we define the corresponding function h_π by giving a decision tree D_π^n for h_π of the appropriate size (=number of leafs). We do this by induction.

Clearly, for $k = 1$ this tree consists of a single leaf labelled by ”1”. The size is 1 and matches the statement of the lemma.

If $n = 1$ and $k = 2$ (partition 2=1+1) this tree consists of one inner node labelled by x_1 and two leafs labelled ”1” and ”2”.

If $k = 2$ and $n > 1$ and $\pi = (s, s'),\ s + s' = 2^n$, then the tree D_π^n can be (inductively) constructed as follows: Let $t = \max\{s, s'\}$ and observe that $t \geq 2^{n-1}$. D_π^n consists of a source labelled by x_n, one successor is a leaf, the other successor is $D^{n-1}_{(t-2^{n-1}, 2^n - t)}$. It follows easily by induction that the size of $D^n_{(s,s')}$ is at most $n+1$.

Now let us fix arbitrary $n > 1,\ k > 2$, and a partition $\pi = (s_1, \ldots, s_k)$ of 2^n. Let us fix the uniquely defined $l,\ 1 \leq l \leq k$, for which $s_1 + \ldots + s_{l-1} \leq 2^{n-1}$ and $s_1 + \ldots + s_l > 2^{n-1}$.

Let $s'_l = 2^{n-1} - (s_1 + \ldots + s_{l-1}),\ s"_l = s_l - s'_l,\ \pi' = (s_1, \ldots, s_{l-1}, s'_l)$, and $\pi" = (s"_l, s_{l+1} \ldots, s_k)$. Observe that both π' and $\pi"$ are partitions of 2^{n-1}.

D_π^n can be defined as a source labelled by x_n, the 0-successor of the source is $D_{\pi'}^{n-1}$, the 1-successor is a copy of $D_{\pi"}^{n-1}$ for which the leafs are labelled by $l, l+1, \ldots, k$ instead of $1, 2, \ldots, (k-l)+1$. By induction hypothesis the size of D_π^n is at most

$$(l-1)(n-1) + 1 + (k-l)(n-1) + 1 = (k-1)n + 3 - k \leq (k-1)n + 1.$$

□

We identify each function $h_\pi : \{0,1\}^n \longrightarrow \{1, \ldots, k\}$ with k Boolean functions $h_\pi^1, \ldots, h_\pi^k$ defined by

$$h_\pi^j(x) = 1 \iff h_\pi(x) = j.$$

We call $h^1, \ldots, h^k$ the *characteristic Boolean functions* of h. Observe

Corollary 2. *For all positive natural numbers n and k with $k \leq 2^n$, all partitions π of 2^n of length k, and all $j,\ 1 \leq j \leq k$, it holds that the Boolean functions $h_\pi^j,\ 1 \leq j \leq k$, can be written as the sum of S_j monomials with $S_1 + \ldots + S_k \leq (k-1)n + 1$.*

Proof. Take the monomials for h_π^j corresponding to the paths in D_π^n leading to leafs with label ”j”. □

Now, for all positive natural numbers n and k with $k \leq 2^n$ fix the *balanced* partition π of 2^n consisting of r times $\lceil 2^n/k \rceil$ and $k - r$ times $\lfloor 2^n/k \rfloor$, where $r = 2^n \mod k$. Denote by $h_{n,k}^1, \ldots, h_{n,k}^k$ the characteristic Boolean functions corresponding to π.

Fix a further positive natural number $m < n$, and let $S = 2^{n-m}$. We now define the operator $T_{n,k,m}$. For all Boolean functions $f : \{0,1\}^n \longrightarrow \{0,1\}$ let $T_{n,k,m}(f) : \{0,1\}^m \longrightarrow \{0,1\}$ be defined

$$T_{n,k,m}(f)(x_1, \ldots, x_m) = \sum_{j=1}^{k} j h_{S,k}^j(y_1, \ldots, y_S),$$

with $y_j = f(x_1, \ldots, x_m, b^{(j)})$, where $b^{(1)}, \ldots, b^{(S)}$ denote the S possible 0,1-assignments of $x_{n-m+1}, \ldots, x_n$ in the canonical order.

Now denote by $B_{m,k}$ the set of all functions $h : \{0,1\}^m \longrightarrow \{1, \ldots, k\}$. In the following lemma we estimate how much the distribution induced by $T_{n,k,m}(f)$ on $B_{m,k}$ deviates from the uniform distribution on $B_{m,k}$.

Lemma 5. *Fix an arbitrary subset E of $B_{m,k}$ and denote by p the probability of the event E w.r.t. the uniform distribution over $B_{n,k}$, and with $\tilde{p}$ the probability of the event E w.r.t. the distribution which is induced via $T_{n,k,m}(f)$ by uniformly distributed random Boolean functions $f : \{0,1\}^n \longrightarrow \{0,1\}$. Then*

$$|p - \tilde{p}| \ \leq \ pk2^{m-S}(1 + k2^{-S})^{2^m}.$$

Corollary 3. *If n, m are choosen in such a way that for $S = 2^{n-m}$ it holds that $2^S > ak2^m$ for some $a \geq 1$, then*

$$|p - \tilde{p}| \ \leq \ (p/a)e^{1/a}.$$

Proof. Let us denote $M = 2^m$. Observe that for all $x \in \{0,1\}^m$ and all $j,\ 1 \leq j \leq k$, the probability that $h(x) = j$, where h denotes a random function distributed according to $T_{n,k,m}(f)$, is in $(1/k - 2^{-S}, 1/k + 2^{-S})$. Consequently,

$$|p - \tilde{p}| \leq pk^M(1/k + 2^{-S})^M - p = p\left(1 + k2^{-S}\right)^M - 1 = pMk2^{-S}(1+z)^{M-1}$$

for some $z \in (1, 1 + k2^{-S})$. Hence, $|p - \tilde{p}| \leq pMk2^{-S}(1 + k2^{-S})^M$.

The Corollary follows by applying the well known inequality $(1 + (x/N))^N \leq e^x$ for all $x > 0$, which yields $(p/a)(1 + (1/aM))^M \ \leq \ (p/a)e^{1/a}$. □

The distinguishing algorithm for $p = 2$

For all $d \geq 2$, a distinguishing algorithm D for depth d circuits over $\{AND, OR, MOD_2\}$ can be designed as follows. Given input parameters n and $m(n) \in n^{O(1)}$, D fixes parameters r and $\tilde{n}$ as the minimal natural numbers fulfilling

$$192m^{2(d+1)} < 2^r \quad \text{and} \quad r^{d+2} < (1/8)\sqrt{\tilde{n}}.$$

Observe that $192 = 24 \cdot 8$, $r \in O(\log(n))$, and $\tilde{n} \in O(\log^{2(d+2)} n)$, and let $\tilde{N} = 2^{\tilde{n}}$.

At next, D computes a parameter $s \in O(\log\log(n))$ such that for $S = 2^s$ it holds that

$$2^S \geq 12\tilde{N} \quad \text{and} \quad S(m+2)+1 \leq m^2.$$

This is always possible for n, m large enough.

Then D chooses randomly a 0,1-assignment c to the variables $x_{\tilde{n}+1}, \ldots, x_{n-s}$.

D accepts iff $D_2(h^c) < (3/4)\tilde{N}$, where h denotes the (1,z,z+1)-variant of $T_{n,3,n-s}(f)$. Observe that the evaluation of one value of h needs S oracle queries and evaluations of $h^1_{S,3}, h^2_{S,3}$ and $h^3_{S,3}$, i.e. the running time of the algorithm is bounded by $(\tilde{N}S)^{O(1)}$ which is quasipolynomially bounded in n.

In the truly random case, h^c is a random function from $\{1,z\}^{\tilde{n}}$ into $\{1,z,z+1\}$ which is distributed according to that distribution on $B_{\tilde{n},3}$ which is induced by the uniform distribution on $B_{\tilde{n}+s,2}$ via $T_{\tilde{n}+s,3,\tilde{n}}$.

Remember that by (II) the probability that $D_2(h) \geq (3/4)\tilde{N}$ is at least $1/2$ w.r.t. the uniform distribution on $B_{\tilde{n},3}$. Consequently, by Corollary 3, and as $2^S > 4 \cdot 3 \cdot \tilde{N}$ we obtain that the probability that A accepts is at most

$$1/2 + (1/4)e^{1/4} \; < \; 11/16.$$

Now consider the pseudorandom case and denote by f the secret function fixed by the oracle. Observe that for all $u = 1,2,3$ the functions $h^u : \{0,1\}^n \longrightarrow \{0,1\}$ defined by

$$h^u(x) = h^u_{S,3}(y_1, \ldots, y_S)$$

with $y_j = f(x, b^j)$, where $b^{(1)}, \ldots, b^{(S)}$ denote the S possible assignments of $x_{n-s+1}, \ldots, x_n$ in the canonical order, can be computed by AND,OR,MOD$_2$-circuits of depth $d+2$ and size $Sm + 2S + 1 = S(m+2)+1 \leq m^2$. (see Corollary 2.)

Consequently, for the given r, there is a degree r^{d+2} polynomial g for which the probability that h differs from g is at most

$$3(m^{2(d+2)} - 1)/(m^2 - 1)2^{-r} \leq 6m^{2(d+1)}2^{-r},$$

for m large enough.

Hence, for an amount of at least 75% of all 0,1-assignments c to the variables $x_{\tilde{n}+1}, \ldots, x_{n-s}$ it holds that the error probability of h^c w.r.t. g^c is at least $24m^{2(d+1)}2^{-r}$, i.e. h^c and g^c differ with respect to at most

$$24m^{2(d+1)}2^{\tilde{n}-r} < (1/8)2^{\tilde{n}}$$

inputs. Suppose that we have choosen such a c. Then, as the degree of g^c is smaller than $(1/8)\sqrt{n}$, we get by (i) that $D_2(h^c) < (3/4)\tilde{N}$, i.e., D accepts with probability $3/4 > 11/16$. We obtain quasipolynomial distinguishing ratio. □

6.2 The Proof of Theorem 5

Theorem 5 *For all $k \geq 1$ it holds that there is a distinguishing algorithm of quasipolynomially bounded ratio for depth $k+1$ circuits consisting of k levels of AND and OR gates connected with one weighted threshold gate as output gate.*

Proof. Let us call an unbounded fanin depth k circuit Σ_k-circuit, resp. Π_k-circuit, if the circuit consists of k inner levels, which contain either only AND-gates, or only OR-gates, and if the top gate is an OR-gate, resp. an AND-gate.

We use the fact that for each Boolean function f with polynomial size weighted threshold-Π_k, or with polynomial size weighted threshold-Σ_k circuits the following holds. With high probability, a random subfunction of f Threshold-MOD$_2$ circuits of quasipolynomial size. According to [11] we consider the set $\{0,1,*\}^n$ of partial assignments to the set of variables $\{x_1, \ldots, x_n\}$ with respect to the probability distribution $R(p)$ which is defined by

$$Pr[\rho] = \Pi_{i=1}^{n} Pr[\rho_i],$$

where $Pr[\rho_i = *] = p$, and $Pr[\rho_i = 0] = Pr[\rho_i = 1] = (1-p)/2$.

We exhibit the *Switching Lemma* [11] saying that for all $f \in B_n$, $p \in (0,1)$ and $s, t \leq n$ it holds the following. If f has a Σ_2- (resp. Π_2-circuit) of bottom fan-in $\leq t$ than the probability that f^ρ has a Π_2-circuit (resp. Σ_2-circuit) of bottom fan-in $\leq s$ is at least $1 - \alpha^s$, where the partial assignment ρ is distributed according to $R(p)$ and the value α can be estimated by $\alpha < 5pt$ (see [30] pp. 325-331 for a nice presentation of the proof).

Moreover, it is shown in [19] that if f has a Σ_2-circuit of bottom fan-in $\leq t$ and a Π_2-circuit of bottom fan-in $\leq s$ then f has a decision tree of depth st, and, consequently, can be computed exactly by a real polynomial of degree st.

Let us fix a polynomial bound $m = m(n) \in n^{O(1)}$ and suppose that $f \in B_n$ can be computed by a threshold-Σ_k circuit S, where each level of the circuit consists of at most $m(n)$ nodes. The case of threshold-Π_k circuits can be treated in a similar way. Fix $s \in O(\log(n))$ to be the smallest number for which $2^s \geq m(n)$. The gates at level 1 of S can be seen as Σ_2- (resp. Π_2-) circuits of bottom fanin 1$\leq s$. Fix an appropriate probability p, which will be specified later, and consider partial assignments ρ of $\{x_1, \ldots, x_n\}$ to be distributed according to $R(p)$. Observe that a standard probability estimation shows that the probability that f^ρ depends on at least pn variables is at least 1/3. Consequently, the probability that each bottom gate of S can be replaced by an equivalent Π_2- (resp. Σ_2-) circuits of bottom fanin s is at least

$$1 - 2/3 - 2^s\alpha^s < 1/3 - (10ps)^s.$$

We fix a number r in such a way that for $p = 2^{-r}$ holds $(10ps)^s \leq 1/6$. Observe that $p^{-1} \in O(\log(n))$.

It follows that the probability that f^ρ depends on at least pn variables and has threshold-Σ_k circuits of width $m(n)$ and bottom fanin s is at least 1/6. This argument can be iteratively applied to f^ρ. It turns out that for ρ distributed according to $R(p^k)$, the probability that f^ρ depends on at least $p^k n$ variables and has a threshold-Π_1- or threshold-Σ_1 circuit of bottom fanin s^2 is at least $(1/6)^k$. Observe that this implies that f^ρ has threshold-MOD$_2$ circuits of size

$$\phi(n,s) = \sum_{i=0}^{s^2} \binom{n}{i} \in n^{O(\log^2 n)},$$

i.e., we can apply the distinguishing scheme for threshold-MOD$_2$ circuits. Let $m' = \phi(n,s) + 1$ and $\tilde{n}$ and $\tilde{N}$ be defined as above in the proof of Theorem 1. We suppose that n, s are large enough such that $12\ln(m')/m' < (1/6)^{k+1}$ and $p^k n > \tilde{n}$.

The distinguishing scheme for weighted threshold-Σ_k- and weighted threshold-Π_k-circuits of width $m(n)$ works as follows. Choose randomly a partial assignment ρ of $\{x_1, \ldots, x_n\}$, where ρ is distributed according to $R(p^k)$ and test whether f^ρ has weighted threshold-MOD$_2$ circuits of size $\phi(n,s)$ with the algorithm of Theorem 1. The choice of the internal parameters p, s, m' and $\tilde{n}$ yields that the advantage is at least $(1/6)^k - (1/6)^{k+1}$ and that the running time is quasipolynomially bounded in n. □