Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 76 (2011)

Block ciphers, pseudorandom functions, and Natural Proofs

Eric Miles* Emanuele Viola*

December 8, 2011

Abstract

This paper takes a new step towards closing the troubling gap between pseudorandom functions (PRF) and their popular, bounded-input-length counterpart: block ciphers. This gap is both quantitative, because block-ciphers are more efficient than PRF in various ways, and methodological, because block-ciphers usually fit in the substitution-permutation network paradigm (SPN) which has no counterpart in PRF.

We give several candidate PRF $\mathcal{F}_i$ that are inspired by the SPN paradigm. This paradigm involves a "substitution function" (S-box). Our main candidates are:

$\mathcal{F}_1 : \{0,1\}^n \to \{0,1\}^n$ is an SPN whose S-box is a random function on $b = O(\lg n)$ bits, given as part of the seed. We prove unconditionally that $\mathcal{F}_1$ resists attacks that run in time $\leq 2^{\epsilon b}$. Setting $b = \omega(\lg n)$ we obtain an inefficient PRF, which however seems to be the first such construction using the SPN paradigm.

$\mathcal{F}_2 : \{0,1\}^n \to \{0,1\}^n$ is an SPN where the S-box is (patched) field inversion, a common choice in block ciphers. $\mathcal{F}_2$ is computable with Boolean circuits of size $n \cdot \log^{O(1)} n$, and in particular with seed length $n \cdot \log^{O(1)} n$. We prove that this candidate has exponential security $2^{\Omega(n)}$ against linear and differential cryptanalysis.

$\mathcal{F}_3 : \{0,1\}^n \to \{0,1\}$ is a non-standard variant on the SPN paradigm, where "states" grow in length. $\mathcal{F}_3$ is computable with size $n^{1+\epsilon}$, for any $\epsilon > 0$, in the restricted circuit class TC^0 of unbounded fan-in majority circuits of constant-depth. We prove that $\mathcal{F}_3$ is almost 3-wise independent.

$\mathcal{F}_4 : \{0,1\}^n \to \{0,1\}$ uses an extreme setting of the SPN parameters (one round, one S-box, no diffusion matrix). The S-box is again (patched) field inversion. We prove that this candidate is a small-bias generator (for tests of weight up to $2^{0.9n}$).

Assuming the security of our candidates, our work also narrows the gap between the "Natural Proofs barrier" [Razborov & Rudich; JCSS '97] and existing lower bounds, in three models: unbounded-depth circuits, TC^0 circuits, and Turing machines. In particular, the efficiency of the circuits computing $\mathcal{F}_3$ is related to a result by Allender and Koucky [JACM '10] who show that a lower bound for such circuits would imply a lower bound for TC^0.

*Supported by NSF grant CCF-0845003. Email: {enmiles,viola}@ccs.neu.edu

ISSN 1433-8092

1 Introduction

This paper takes a new step towards closing the troubling gap between pseudorandom functions ([GGM86], cf. [Gol01, §3.6]) and their popular, bounded-input-length counterpart: block ciphers. This gap is both quantitative and methodological.

It is quantitative because all known candidate pseudorandom functions (hereafter, PRF) (e.g. [GGM86, HILL99, NR04, HRV10, VZ11]) have seed length at least quadratic in the input length n, which also implies a quadratic lower bound on the circuit size of such PRF. In contrast, block ciphers typically have seed length which *equals* the input length. This is for example the case with the 128-bit version of the Advanced Encryption Standard (AES) by Daemen and Rijmen [DR02].

It is methodological because modern block ciphers, including AES, typically follow the *substitution-permutation network* (SPN) paradigm. An SPN is computed over a number of rounds, where each round "confuses" the input by dividing it into bundles and applying a substitution permutation (S-box) to each bundle, and then "diffuses" the bundles by applying a matrix with certain "branching" properties. (Cf. [Sha49].) No piece of this structure appears to have been used to construct PRF. In fact, until the present paper no asymptotic analysis of the SPN structure was given. This is in stark contrast with the seminal work of Luby and Rackoff [LR88] that gave such an analysis for the so-called *Feistel network* structure (which in particular was the basis for the block cipher DES, the predecessor to AES). Moreover the SPN structure is tailored to resist two general attacks on block ciphers which appear to be ignored in the PRF literature, namely linear and differential cryptanalysis.

In this paper we give several candidate PRF that are inspired by the SPN structure. Some of our candidates have better parameters than previous candidates, where by parameters we refer to the seed length and the resources required to compute each function in various computational models.

1. We first consider an SPN with a random S-box (specified as part of the seed). We prove unconditionally that this resists attacks that run in time less than the seed length. For example we can set the seed length to n^c and withstand attacks running in time $n^{c'}$ for sufficiently large c and $c' = \Theta(c)$. (Note that being a PRF means that the seed length is n^c and that the function withstands all attacks running in time $n^{c'}$ for *any* c'.)

   This result is analagous to that of Luby and Rackoff, who analyzed the Feistel network structure when a certain component is instantiated with a random function, and indeed we prove the same level of security (exponential in the input size of the random function). The techniques used are similar to those in the work by Naor and Reingold [NR99] that followed Luby and Rackoff's. To our knowledge this is the first construction of a (provably secure, inefficient) PRF using the SPN structure.

2. Using the AES S-box and a strengthened version of the AES diffusion matrix, we give a candidate computable with Boolean circuits of size $n \cdot \log^{O(1)} n$, and in particular with seed length $n \cdot \log^{O(1)} n$. We prove that this candidate has exponential security

$2^{\Omega(n)}$ against linear and differential cryptanalysis by extending a result due to Kang et al. [KHL+01].

3. Again using the AES S-box and a different diffusion matrix, we give a candidate computable with size $n^{1+\epsilon}$, for any $\epsilon > 0$, in the restricted circuit class TC^0 of unbounded fan-in majority circuits of constant-depth. The diffusion matrix used here blows up the state to size $O(n)$, and we output a single bit by taking the inner product of this state with a random string. We prove that this candidate is almost 3-wise independent.

4. We give another single-bit output candidate which uses an extreme setting of the SPN parameters (one round, one S-box, no diffusion matrix). This can be viewed as a slightly modified version of the Even-Mansour cipher [EM97] that uses the AES S-box in place of a random permutation. We prove that this candidate is a small-bias generator.

5. Our final candidate is a straightforward generalization of AES, and may be folklore. We show that it is computable by size $O(n^2)$, depth $O(n)$ Boolean circuits, and we further show that for each fixed seed k it is computable in time $O(n^2)$ by a single-tape Turing machine with $O(n^2)$ states. We do not have any proof of security, but the (heuristic) arguments underlying AES's security also apply to this candidate.

Natural proofs. The landscape of circuit lower bounds remains bleak, despite exciting recent results [Wil11]. Researchers however have been successful in explaining this lack of progress by pointing out several "barriers," i.e. establishing that certain proof techniques will not give new lower bounds [BGS75, RR97, AW08].

Of particular interest to us is the Natural Proofs work by Razborov and Rudich [RR97]. They make the following two observations. First, most lower-bound proofs that a certain function $f : \{0,1\}^n \to \{0,1\}$ cannot be computed by circuits C (e.g., C = circuits of size n^2) entail an algorithm that runs in time polynomial in $N := 2^n$ and can distinguish truth-tables of n-bit functions $g \in C$ from truth-tables of random functions (i.e., a random string of length N). (For example, the algorithm corresponding to the restriction-based proof that Parity is not in AC^0, given $f : \{0,1\}^n \to \{0,1\}$, checks if there is one of the $2^{O(n)} = N^{O(1)}$ restrictions of the n variables that makes f constant.) Informally, any proof that entails such an algorithm is called "natural."

The second observation is that, under standard hardness assumptions, no algorithm such as the above one exists when C is a sufficiently rich class. This follows from the existence of PRF with security $2^{s^{\Omega(1)}}$ where s is the seed length (e.g. [GGM86, HILL99, NR04, HRV10, VZ11]) and by setting $s := n^c$ for a sufficiently large c.

The combination of the two observations is that no natural proof exists against circuits of size n^c, for some constant $c \geq 2$.

Moreover, the PRF construction [NR04] by Naor and Reingold is implementable in TC^0, pushing the above second observation "closer" to the frontier of known circuit lower bounds. For completeness we also mention that this PRF achieves seed length $s = O(n^2)$ and is a candidate to having hardness $2^{\Omega(n)}$ under elliptic-curve conjectures.

The gap between lower bounds and PRF. However, the natural proofs barrier still has a significant gap with known lower bounds, due to the lack of sufficiently strong PRF. For example, there is no explanation as to why one cannot prove superlinear-size circuit lower bounds. For this one would need a PRF $f_k : \{0,1\}^n \to \{0,1\}$ that is computable by linear-size circuits (hence in particular with $|k| = O(n)$) and with exponential hardness 2^n. (So that, given n, if one had a distinguisher running in time $2^{O(n)}$, one could pick a PRF on inputs of length bn for a large enough constant b, to obtain a contradiction.)

A recent work by Allender and Koucký [AK10] brings to the forefront another setting where the Natural Proofs barrier does not apply: proving lower bounds on TC^0 circuits of size $n^{1+\epsilon}$ and depth d, for any $\epsilon > 0$ and large enough $d = d(\epsilon)$. (As mentioned above, the Naor-Reingold PRF requires larger size.) This setting is especially interesting because [AK10] shows that such a lower bound for certain functions implies a "full-fledged" lower bound for TC^0 circuits of polynomial-size. Moreover even if the first lower bound were natural, the latter would not be, thus circumventing the Naor-Reingold PRF.

Another long-standing problem is to exhibit a candidate PRF in ACC^0.

Of course, circuit models such as the above ones are only some of the models in which the gap between candidate PRF and lower bounds is disturbing. Other such models include various types of Turing machines, and small-space branching programs. For example, there is no explanation as to why the lower bounds for single-tape Turing machines stop at quadratic time, cf. [KN97, §12.2].

Assuming the security of our candidates, our work narrows this gap in three ways. First, Candidate 2 is computable by quasilinear-size Boolean circuits. Second, Candidate 3 is computable by TC^0 circuits of size $n^{1+\epsilon}$ and depth $d = d(\epsilon)$ for any $\epsilon > 0$. Third, for each fixed seed k Candidate 5 is computable in time $O(n^2)$ by a single-tape Turing machine with $O(n^2)$ states (note that the fixed-seed setting suffices for the Natural Proofs connection).

1.1 Background on SPNs

To formally define our candidates, we begin by reviewing the SPN structure (refer to Figure 1).

An SPN $C_k : \{0,1\}^n \to \{0,1\}^n$ is indexed by a key $k = (k_0, \ldots, k_r) \in (\{0,1\}^n)^{r+1}$, and is specified by the following three parameters and two functions:

- $r \in \mathbb{N}$, the number of *rounds*
- $b \in \mathbb{N}$, the *S-box size*
- $m \in \mathbb{N}$, the *number of S-boxes*
- $S : \mathrm{GF}(2^b) \to \mathrm{GF}(2^b)$, the *S-box*
- $M : \big(\mathrm{GF}(2^b)\big)^m \to \big(\mathrm{GF}(2^b)\big)^m$, the *linear transformation.*

The input/output size of C_k is given by $n := mb$. Throughout this paper, we assume a fixed canonical mapping between $\{0,1\}^b$ and $\mathrm{GF}(2^b)$.

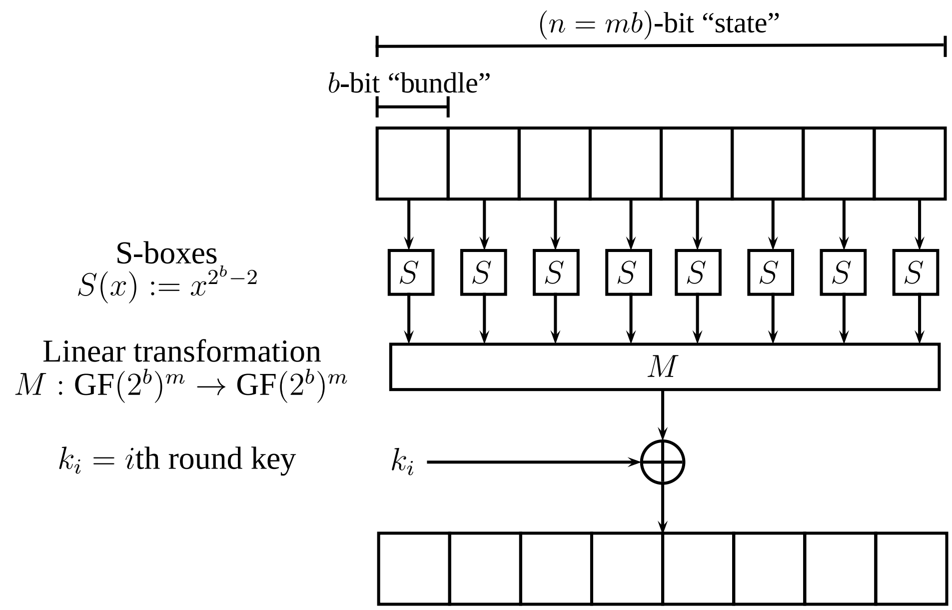

Figure 1: One round of an SPN

C_k is computed over r rounds. The ith round ($1 \leq i \leq r$) is computed over three steps: (1) m parallel applications of S; (2) application of M to the entire state; (3) XOR of the entire state with the round key k_i. Note that each round is identical except for step (3).[1]

On input x, $C_k(x)$ gives $x \oplus k_0$ as input to the first round; the output of round i becomes the input to round $i+1$ (for $1 \leq i < r$), and $C_k(x)$'s output is the output of the rth round.

We now briefly review how the security of an SPN is evaluated against two general attacks on block ciphers: linear and differential cryptanalysis. Resistance to these attacks is typically seen as the main security feature of SPNs. Full details are deferred to Appendix A.

For both linear and differential cryptanalysis, a crucial property in the security proof is that the linear transformation M has maximal *branch number*, defined as follows.

Definition 1.1. Let $M : \mathbb{F}^m \to \mathbb{F}^m$ be a linear transformation acting on vectors over a field $\mathbb{F}$. The *branch number* of M is

$$\mathrm{Br}(M) = \min_{\alpha \neq 0^m} \left(w(\alpha) + w(M(\alpha)) \right) \leq m + 1$$

where $w(\cdot)$ denotes the number of non-zero elements.

Linear cryptanalysis [Mat94] exploits the existence of linear correlations to attack a block cipher C_k. For a function $f : \{0,1\}^n \to \{0,1\}^n$ and input/output parities $\Gamma_x, \Gamma_y \in \{0,1\}^n$, define the *correlation* of f with respect to Γ_x and Γ_y as

$$\mathrm{Cor}_{\Gamma_x,\Gamma_y}(f) := 2 \cdot \Pr_x[\langle \Gamma_x, x \rangle = \langle \Gamma_y, f(x) \rangle] - 1.$$

[1]SPNs are sometimes defined more generally, e.g. by allowing the S-box to vary across rounds or by allowing a more complex interaction with k than XOR.

For a block cipher C_k, the parameter of interest for linear cryptanalysis is

$$p_{\mathrm{LC}}(C_k) := \max_{\Gamma_x, \Gamma_y \neq 0} \left(\mathbb{E}_k \left[\mathrm{Cor}_{\Gamma_x, \Gamma_y} (C_k)^2 \right] \right).$$

Specifically, the attack requires an expected number of plaintext/ciphertext pairs proportional to $1/p_{\mathrm{LC}}(C_k)$.

Differential cryptanalysis [BS91] attacks a block cipher C_k by exploiting the relationship between the XOR difference of two inputs to C_k and the XOR difference of the corresponding outputs. For a function $f_k : \{0,1\}^n \to \{0,1\}^n$ parameterized by a key k, and input/output differences $\Delta_x, \Delta_y \in \{0,1\}^n$, define the *difference propagation probability* (DPP) of f_k with respect to Δ_x and Δ_y as

$$\mathrm{DPP}_{\Delta_x, \Delta_y}(f_k) := \Pr_{x,k}[f_k(x) \oplus f_k(x \oplus \Delta_x) = \Delta_y].$$

(If f is not parameterized by a key, k is ignored in this definition). For a block cipher C_k, the parameter of interest for differential cryptanalysis is

$$p_{\mathrm{DC}}(C_k) := \max_{\Delta_x, \Delta_y \neq 0} \left(\mathrm{DPP}_{\Delta_x, \Delta_y} (C_k) \right).$$

Specifically, the attack requires an expected number of plaintext/ciphertext pairs proportional to $1/p_{\mathrm{DC}}(C_k)$.

The following theorem, due to Kang et al. [KHL+01], gives a bound on p_{LC} and p_{DC} for 2-round SPNs with maximal branch number.

Theorem 1.2. ([KHL+01], Thms. 5 & 6) Let $C_k : \{0,1\}^n \to \{0,1\}^n$ be an SPN with $r = 2$ rounds and S-box S. Let $q := \max_{\Gamma_x, \Gamma_y \neq 0} \left(\mathrm{Cor}_{\Gamma_x, \Gamma_y} (S)^2 \right)$ denote the maximum squared correlation of S, and let $p := \max_{\Delta_x, \Delta_y \neq 0} \left(\mathrm{DPP}_{\Delta_x, \Delta_y} (S) \right)$ denote the maximum DPP of S. If $\mathrm{Br}(M) = m + 1$, then $p_{\mathrm{LC}}(C_k) \leq q^m$ and $p_{\mathrm{DC}}(C_k) \leq p^m$.

Intuitively, the S-box provides security q (resp. p) against linear (resp. differential) cryptanalysis, and this security is raised to the branch number. For typical S-boxes, such as the one used in AES, one can have $q = p = 2^{-b+2}$, and so the theorem guarantees security exponentially small in $bm = n$. (For completeness we note that one cannot directly apply the above theorem to AES because it is a more complicated SPN.)

While resistance to linear and differential cryptanalysis is the main security feature of the SPN structure (and indeed, "the most important criterion in the design" of AES [DR02, p. 81]), considerations are usually also taken to prevent attacks that would exploit algebraic structure in the cipher. For instance the S-box that is used in AES, which we also adopt for our candidates 2-5, is defined by $S(x) := x^{2^b-2}$ and was chosen to allow the computation to have high degree when considered as a multivariate polynomial over GF(2). The use of $x \mapsto x^{2^b-2}$ results in each of S's output bits having (near-maximum) degree $b - 1$. Using instead $x \mapsto x^3$ would not diminish resistance to linear and differential cryptanalysis, but it would result in degree (only) 2 [Pie91, Nyb93] (cf. [Kop11, Lemma 9]).

Finally, although a block cipher's security is often measured against *key-recovery* attacks, we share many researchers' viewpoint that *distinguishing* attacks are the correct model. We also point out that the linear and differential cryptanalysis techniques can be seen as falling in the latter type of attacks, as they do construct a distinguishing algorithm (which is then used to select the correct round key from a set of potential keys).

1.2 Our candidates

We now describe our candidates. Candidates 1, 2, and 5 output n bits, while Candidates 3 and 4 output 1 bit. For $i \geq 2$ we use $\mathcal{F}_i$ to refer to the function computing Candidate i.

Continuing the discussion above, we need the degree of each of our candidates (as a multivariate GF(2)-polynomial) to be $\geq \epsilon n$, for some constant ϵ, to resist attacks that exploit the degree of this polynomial. (For completeness we present such an attack in Appendix B, showing that a PRF which has degree $o(n)$ cannot have hardness 2^n.) The only non-linear operation in the entire cipher is the S-box, which has degree $b-1$, and hence the maximum possible degree of each output bit is at most $(b-1)^r$. Hence we make sure that

$$b^r \geq n$$

in each of our candidates. (The distinction between $(b-1)^r \geq \epsilon n$ and $b^r \geq n$ is unimportant, as in our candidates we can always increase r by a constant factor, except in Candidate 4 where we have $b = n$ and $r = 1$.)

Candidate 1. Our first candidate $\mathcal{F}_1$ is an r-round SPN with an S-box that is chosen uniformly at random (i.e. specified as part of $\mathcal{F}_1$'s key) from the set of all functions mapping $\mathrm{GF}(2^b)$ to itself. (Analyzing this candidate when S is a random *permutation* is a natural research direction which we do not address here.) We show that any adversary A has small advantage in distinguishing $\mathcal{F}_1$ from a random function F.

Theorem 1.3. *If A makes at most q total queries to its oracle, then*

$$\left| \Pr_F \left[A^F = 1\right] - \Pr_{\mathcal{F}_1} \left[A^{\mathcal{F}_1} = 1\right] \right| < O(r^2 m^3 q^2) \cdot 2^{-b}.$$

The bound achieved here is similar to that of Luby and Rackoff [LR88] in the sense that it is exponentially small in the size of the random function, with a quadratic loss in the number of queries. The proof of this theorem is very similar to that of [NR99, Thm. 3.2], and proceeds by bounding the collision probability between any two inputs to S in the final round. However we face an additional hurdle, namely that the inputs to the random function S in the final round depend on outputs of S in previous rounds.

By setting $b = \omega(\log n)$ and $r = \log n$, we get an inefficient PRF (with security $n^{\omega(1)}$). We also note that by setting $b = c \log n$ for some sufficiently large constant c, $\mathcal{F}_1$ is computable in time $n^{O(c)}$ and has security $n^{c'}$ for some $c' = \Omega(c)$.

Finally, note that Theorem 1.3 implies corresponding bounds on $p_{\mathrm{LC}}(\mathcal{F}_1)$ and $p_{\mathrm{DC}}(\mathcal{F}_1)$.

Candidate 2. In this candidate we set $b = \Theta(\log n)$, and we use the same S-box as in AES except for this increase in input/output size (recall that the AES S-box maps $x \mapsto x^{2^b-2}$). We use a linear transformation M with maximal branch number, and M is constructed from an error-correcting code in a similar manner to the linear transformation in AES. (AES's linear transformation does not have maximal branch number however, a choice that was made to reduce computation time.) We set the number of rounds $r = \Theta(\log n)$ (observe that $b^r \geq n$).

We prove that Candidate 2 is computable by Boolean circuits of quasilinear-size $\widetilde{O}(n) := n \cdot \log^{O(1)} n$. To show this, note that since r is logarithmic it is enough to show how to compute each round with these resources. Moreover, since b is logarithmic, computing the S-boxes comes at little cost.

Our main technical contribution in this candidate is to show how to efficiently compute the linear transformation M; specifically, we show that it can be computed with size $\widetilde{O}(n)$, for a total circuit size of $r \cdot \left(b^{O(1)} + \widetilde{O}(n)\right) = \widetilde{O}(n)$. A common method for constructing maximal-branch-number linear transformations is to use the generator matrix G of a $m \to 2m$ maximum distance separable (MDS) code; specifically, if $G^T = [I \mid A]$, then $M := A$ has maximal branch number. Our method for computing M efficiently has two parts. First, we use a result by Roth and Seroussi [RS85] that if G generates a Reed-Solomon code (which is well-known to be MDS), then M forms a $t \times t$ *Cauchy matrix* (a type of matrix specified by $O(t)$ elements). We then use a result by Gerasoulis [Ger88] to compute the product of a vector (consisting of bundles of the state) and a Cauchy matrix in quasilinear time; this requires a simple adaptation of the algorithm in [Ger88] to fields of characteristic 2.

By proving a simple inductive extension of Theorem 1.2 and combining it with a theorem of Nyberg [Nyb93], we show that this candidate has exponential security against linear and differential cryptanalysis.

Theorem 1.4. 1. $p_{\mathrm{LC}}(\mathcal{F}_2) \leq 2^{-\Omega(n)}$. 2. $p_{\mathrm{DC}}(\mathcal{F}_2) \leq 2^{-\Omega(n)}$.

We do not know how to get a candidate computable by circuits of size $O(n)$.

Candidate 3. In the previous candidate, the components S and M remain essentially unchanged from AES. In Candidate 3, we also keep S the same (aside from the increase in input/output size), but we modify the linear transformation M.

Our observation is that the rationale for using a linear transformation with maximal branch number is just that it allows one to lower bound the number $\mathcal{A}$ of so-called "active" S-boxes, which can be defined as follows. Let C be an SPN which uses the identity permutation for S and which has $k_i := 0$ for $0 \leq i \leq r$. Let $w_b : \left(\{0,1\}^b\right)^m \to \mathbb{N}$ be the function that counts the number of non-zero b-bit bundles in its input. Then,

$$\mathcal{A} := \min_{0^n \neq x \in \{0,1\}^n} \sum_{i=1}^{r} w_b(\text{state of } C(x) \text{ at the beginning of round } i).$$

This number $\mathcal{A}$ is crucial in evaluating the security of SPNs against linear and differential cryptanalysis (cf. [KHL+01, DR02]). With a simple modification to M, we get that a constant fraction of the S-boxes in each round are active. Specifically we use the full generator matrix of an error correcting code with minimum distance $\Omega(n)$, which comes at the expense of expanding the state from n bits to $O(n)$ bits at each round. To counteract the fact that such codes may have some output positions fixed to constant values (leading to a simple distinguishing attack), the computation of Candidate 3 concludes by taking the inner product of the state with a uniform $O(n)$-bit vector that is given as part of the seed. Candidate 3 therefore outputs a single bit.

We take $b = n^\epsilon$ and $r = O(1/\epsilon)$ for arbitrarily small $\epsilon > 0$, and so each round is computable in size

$$\frac{n}{b} \cdot \mathrm{poly}(b) = n^{1+O(\epsilon)},$$

and the whole circuit also in size $n^{1+O(\epsilon)}$.

We further show that Candidate 3 is computable even by TC^0 circuits of size $n^{1+O(\epsilon)}$ for any $\epsilon > 0$ (with depth depending on ϵ), cf. § "The gap between lower bounds and PRF" above. The main technical difficulty in implementing this candidate with the required resources is that the S-box requires computing *inversion* in a field of size 2^b (recall $b = n^{\Omega(1)}$). To implement this in TC^0 we note (cf. [HV06]) that inverting the field element $\alpha(x)$ can be accomplished as:

$$\alpha(x)^{2^b-2} = \alpha(x)^{\sum_{i=1}^{b-1} 2^i} = \prod_{i=1}^{b-1} \alpha(x)^{2^i} = \prod_{i=1}^{b-1} \alpha\left(x^{2^i}\right)$$

where the last equality follows from the fact that we are working in characteristic 2. By hard-wiring the $\leq b$ powers $x, x^2, \ldots, x^{2^{b-1}}$ of x in the circuit, and using the fact that the iterated product of $\mathrm{poly}(n)$ field elements is computable by $\mathrm{poly}(n)$-size TC^0 circuits (see e.g. [HAB02, Corollary 6.5] and cf. [HV06]), we obtain a TC^0 circuit.

Because Candidate 3 deviates somewhat from the SPN structure, we cannot use Theorem 1.2, and indeed it is not clear how to define differential cryptanalysis for functions which output only one bit. However, we are able to leverage a technique from differential cryptanalysis to prove that Candidate 3 is almost 3-wise independent.

Definition 1.5. A function $f : \{0,1\}^n \to \{0,1\}$ parameterized by a key k is (d, ϵ)*-wise independent* if for any distinct $x_1, \ldots, x_d \in \{0,1\}^n$, the distribution $(f(x_1), \ldots, f(x_d))$ induced by a uniform choice of k is ϵ-close to U_d in statistical distance.

Theorem 1.6. *$\mathcal{F}_3$ is $(3, 2^{-\Omega(n)})$-wise independent.*

Finally, we mention that implicit in the assumption that Candidate 3 is indeed hard is the assumption that field inversion cannot be computed by unbounded fan-in constant depth circuits with parity gates $\mathrm{AC}^0[\oplus]$. For otherwise, it can be shown that the whole candidate would be in that class, in contradiction with an algorithm in [RR97, §3.2.1] which distinguishes truth tables of $\mathrm{AC}^0[\oplus]$ functions from random ones in quasipolynomial time. (M can be seen to be a linear operation over GF(2), hence it can be computed easily with parity gates.) Some evidence that field inversion is not in $\mathrm{AC}^0[\oplus]$ comes from the fact that related functions can indeed be shown not to be in $\mathrm{AC}^0[\oplus]$; see [HV06] and the recent work by Kopparty [Kop11].

Candidate 4. In this candidate, we use the extreme setting of parameters $b = n$ and $r = 1$. In other words, Candidate 4 consists of one round, and this round contains only a single S-box (and in particular no linear transformation). This construction can be seen as a concrete instantiation of the Even-Mansour block cipher [EM97], using the AES S-box in place of the random permutation oracle. While this setting does indeed preserve resistance to linear and differential cryptanalysis, we exhibit a simple attack, inspired by Jakobsen and Knudsen [JK01], in which we exploit the algebraic structure to recover the key with just 4 queries.

This attack was somewhat unexpected to us, because intuitively increasing the S-box size should increase security, and we ask whether there is any PRF of the form $f_{k_0,k_1}(x) := g(x + k_0) + k_1$ where g is a fixed function. (Even and Mansour (see also [GR04]) show this f is secure when g is a random function, even if all parties have oracle access to g.)

We then put forth a related candidate $\mathcal{F}'_4$ where we only output the Goldreich-Levin bit [GL89]: $\mathcal{F}'_4(x) := \langle (x + k_0)^{2^b-2}, k_1 \rangle$. We prove that this candidate is a small-bias generator [NN93, AGHP92].

Theorem 1.7. *For any choice of $d \leq 2^n$, $\mathcal{F}'_4$ is a d-wise small-bias generator with error $d/2^n$: for any distinct $a_1, \ldots, a_d \in \{0,1\}^n$,*

$$\left| \Pr_{k_0,k_1} \left[\sum_{i=1}^{d} \mathcal{F}'_4(a_i) = 0 \right] - \frac{1}{2} \right| < \frac{d}{2^n}.$$

Using Braverman's result [Bra09] (cf. [Baz09, Raz09]) we obtain that this candidate also fools small-depth AC^0 circuits of any size $w = 2^{n^{o(1)}}$ (that look at only w fixed output bits of the candidate).

Using the same ideas for Candidate 3, this candidate is also computable by poly-size TC^0 circuits. For unbounded-depth circuits, a more refined size bound $\widetilde{O}(n^2)$ follows from the exponentiation algorithm in [GvzGPS00].

Candidate 5. Our final candidate is a straightforward generalization of AES, and may be folklore. We set $b = 8$ as in AES and we again use AES's S-box. We also use the same linear transformation as in AES (which is slightly different from that of Candidate 2, cf. §2.5), except for the necessary increase in the input/output size. We set the number of rounds $r = n$, and thus the size of the seed is $|k| = n(n + 1)$.

Candidate 5 is computable by size $O(n^2)$, depth $O(n)$ Boolean circuits. For each fixed seed k, Candidate 5 is also computable in time $O(n^2)$ by a single-tape Turing machine with $O(n^2)$ states.

We do not know how to get a candidate computable in time $O(n)$ on a 2-tape Turing machine.

2 New PRF candidates

In this section we give the details of the candidate PRFs described in §1.2. Recall that we denote Candidate i by $\mathcal{F}_i$.

2.1 Candidate 1

We first analyze the pseudorandomness of the SPN structure when the S-box is a uniformly random function. The results of this section are of a similar flavor, and use similar techniques, as those of Luby and Rackoff [LR88] and the following work by Naor and Reingold [NR99]. One notable difference is that we study SPNs as pseudorandom *functions*, and in particular we do not allow inverse queries to the SPN. (Indeed, if the S-box is not a permutation then the SPN may not be either, in which case inverse queries are not well-defined.) Adapting this proof to handle bidirectional queries is a natural research direction which is not addressed here.

Our analysis in this section holds for SPNs in which the linear transformation M is invertible and has all entries $\neq 0$. We observe that this includes all matrices with maximal branch number.

Claim. *Let $M \in (GF(2^b))^{m \times m}$ be any matrix with maximal branch number $m+1$. Then, all entries of M are non-zero and M is invertible.*

Proof. Assume for contradiction that $M_{i,j} = 0$ for some $i, j \leq m$. Let $x \in (\mathrm{GF}(2^b))^m$ be the vector such that $x_j = 1$ and $x_{j'} = 0$ for $j' \neq j$. Then $(Mx)_i = 0$, and so $\mathrm{Br}(M) \leq w(x) + w(Mx) \leq m$.

To see that M is invertible, simply note that if $Mx = My$ for $x \neq y$, then $M(x+y) = 0^m$. Since $x + y \neq 0^m$, we would again have $\mathrm{Br}(M) \leq m$. □

For the remainder of this section, fix any invertible $M \in (\mathrm{GF}(2^b))^{m \times m}$ such that all entries are non-zero. For any function $S : \mathrm{GF}(2^b) \to \mathrm{GF}(2^b)$ and any set of round keys $(k_0, \ldots, k_{r-1}) \in (\{0,1\}^n)^r$, let $\mathcal{F}_1 = \mathcal{F}_1(S, k_0, \ldots, k_{r-1})$ be the r-round SPN on $n := mb$ bits defined by these components, where the final round consists only of S-boxes (i.e. the final round omits the linear transformation and the key addition).

Let $A : (\mathbb{N}, \mathbb{N}) \to \{0,1\}$ denote an adversary with oracle access to a function mapping $(\mathrm{GF}(2^b))^m$ to itself; A's input is simply $(1^m, 1^b)$ which we omit from now on. We will show that A has small advantage in distinguishing between the case when its oracle is a uniformly random function F, and when its oracle is $\mathcal{F}_1$ for a uniform choice of $(S, k_0, \ldots, k_{r-1})$.

Theorem 1.3. *If A makes at most q total queries to its oracle, then*

$$\left| \Pr_{F}\left[A^F = 1\right] - \Pr_{\mathcal{F}_1}\left[A^{\mathcal{F}_1} = 1\right] \right| < O(r^2 m^3 q^2) \cdot 2^{-b}.$$

Below, we will analyze the first $(r-2)$ rounds in a different way from the final 2 rounds, and to this end we define the following two functions. Let $\rho = \rho(S, k_0, \ldots, k_{r-3})$ compute everything in $\mathcal{F}_1$ before the XOR with k_{r-2}, and let $\rho' = \rho'(S, k_{r-2}, k_{r-1})$ compute the remainder of $\mathcal{F}_1$. So, $\mathcal{F}_1(x) = \rho'(\rho(x))$. As handling ρ' will be the more involved part of the analysis, we note that it can be written as

$$\rho'(x) := S^*\left(M \cdot S^*(x + k_{r-2}) + k_{r-1}\right)$$

where $S^*(x) := \left(S(x^{(1)}), \ldots, S(x^{(m)})\right)$ for any $x = x^{(1)} \cdots x^{(m)}$.

2.1.1 Proof overview

The proof proceeds in two stages. In the first stage, we consider any set of distinct queries $x_1, \ldots, x_q$, and we show that there is a low-probability event BAD over the choice of $(S, k_0, \ldots, k_{r-1})$ such that, conditioned on $\neg$BAD, $\{\mathcal{F}_1(x_i)\}_{i \leq q}$ is uniformly distributed. Essentially, BAD is the event that any two SPN queries induce the same query to some S-box in the final round.

In the second stage, we consider the distribution over transcripts of A's interaction with its oracle; we use the results of the first stage in a probability argument to show that the transcripts are distributed nearly identically in either setting, and thus that A's distinguishing advantage is small. This framework has been used in a number of other works, e.g. [NR99, RR00, GR04].

The first stage actually shows that $\mathcal{F}_1$ is almost q-wise independent, or alternatively that it is pseudorandom against adversaries that make $\leq q$ non-adaptive queries. The technique used in the second stage is a rather generic way of extending the proof to adaptive queries; however we note that it crucially relies on the existence of the event BAD, and indeed it is not the case that any almost q-wise independent function is pseudorandom against adversaries making q adaptive queries. To see this, consider the distribution over functions $f : [N] \to [N]$ in which each output is selected uniformly and independently with the restriction that $f(f(0)) := 0$. This distribution is almost pairwise independent, but is trivially distinguishable with two adaptive queries.

2.1.2 Stage 1

Fix distinct $x_1, \ldots, x_q \in (\mathrm{GF}(2^b))^m$. Our analysis will use the state of the SPN's computation immediately before the final round of S-boxes, and we denote these states by

$$z_i := M \cdot S^*(\rho(x_i) + k_{r-2}) + k_{r-1}.$$

(So, $\mathcal{F}_1(x_i) = S^*(z_i)$.) We view $(S, k_0, \ldots, k_{r-1})$ being chosen as follows:

1. Uniformly choose $k_0, \ldots, k_{r-3}$.

2. Run the computation of $\rho(x_i)$ for all $i \leq q$, and each time the S-box is evaluated on a previously-unseen input, choose the output uniformly at random. Let $H \subseteq \mathrm{GF}(2^b)$ be the set of at most $qm(r-2)$ inputs whose output is determined after this step.

3. Uniformly choose k_{r-2}.

4. Uniformly choose the output of S on each block of each $(\rho(x_i)+k_{r-2})$ whose output is not already determined.

5. Uniformly choose k_{r-1}.

6. Uniformly choose the output of S on all remaining inputs.

It is clear that, for any $x_1,\ldots,x_q$, this distribution is uniform.

We now define the event BAD mentioned in the previous subsection. Informally BAD holds if, after step 5, any of the S-inputs that need to be evaluated (i.e. the blocks of the z_i) collide either with each other or with one of the inputs selected in steps 2 and 4. To reduce notation, we use the following definition.

Definition 2.1. Let $x, y \in (\mathrm{GF}(2^b))^m$, and denote $x = x^{(1)}\cdots x^{(m)}$ and $y = y^{(1)}\cdots y^{(m)}$. Then, we say that x and y *collide* if $\exists \ell,\ell' :\ x^{(\ell)} = y^{(\ell')}$. Further, for any $T \subseteq \mathrm{GF}(2^b)$, we say that x and T *collide* if $\exists \ell \le m, t \in T :\ x^{(\ell)} = t$. Finally, we say that x *self-collides* if $\exists \ell \neq \ell' :\ x^{(\ell)} = x^{(\ell')}$.

Now, let BAD $=$ BAD$(x_1,\ldots,x_q)$ be the set of all $(S,k_0,\ldots,k_{r-1})$ such that at least one of the following holds:

(a) $\exists h,h' \in H : S(h) = S(h')$.

(b) $\exists i < q : (\rho(x_i)+k_{r-2})$ and H collide.

(c) $\exists i < q : z_i$ and H collide.

(d) $\exists i,i' \le q : z_i$ and $(\rho(x_i)+k_{r-2})$ collide.

(e) $\exists i \le q : z_i$ self-collides.

(f) $\exists i \neq i' \le q : z_i$ and $z_{i'}$ collide.

It is crucial for us that determining whether BAD holds can be checked after step 5 in choosing $(S,k_0,\ldots,k_{r-1})$.

We now prove two lemmas showing that BAD occurs with low probability, and that the query answers are uniformly distributed when conditioned on $\neg$BAD. In the remainder of this subsection, we will simply use BAD to mean $(S,k_0,\ldots,k_{r-1}) \in$ BAD.

Lemma 2.2. $\Pr\limits_{S,k_0,\ldots,k_{r-1}}[\text{BAD}] < O(r^2m^3q^2)\cdot 2^{-b}$.

Proof. We start by bounding the probability of items (a)-(e) individually.

First, we have $\Pr[(a)] < |H|^2 \cdot 2^{-b} \le (qm(r-2))^2 \cdot 2^{-b}$ by a union bound over all pairs of elements of H.

We analyze the remaining items starting after step 2, so in particular we now let $k_0,\ldots,$ k_{r-3}, H, and $S(H)$ be fixed arbitrarily.

By a union bound over each block of each $\rho(x_i)$ and each element of H, we have $\Pr_{k_{r-2}}[(\mathrm{b})] \le qm \cdot qm(r-2) \cdot 2^{-b}$.

Fix any k_{r-2}. Fix any outputs of S on the blocks of $(\rho(x_i) + k_{r-2})$ for all i, which fixes $\tilde{z}_i := M \cdot S^*(\rho(x_i) + k_{r-2})$; note that $z_i = \tilde{z}_i + k_{r-1}$. Then, $\Pr_{k_{r-1}}[(\mathrm{c})] \le qm \cdot qm(r-2) \cdot 2^{-b}$ by a union bound over each block of each $\tilde{z}_i$ and each element of H.

By the same argument as for (c), $\Pr_{k_{r-1}}[(\mathrm{d})] \le (qm)^2 \cdot 2^{-b}$, where the union bound is now over each block of each z_i and each block of each $(\rho(x_i) + k_{r-2})$.

Let the $\tilde{z}_i$ be defined as above; then for each i, $\Pr_{k_{r-1}}[(z_i = \tilde{z}_i + k_{r-1}) \text{ self collides}] < m^2 \cdot 2^{-b}$ by a union bound over pairs of blocks. So, $\Pr_{k_{r-1}}[(\mathrm{e})] < qm^2 \cdot 2^{-b}$.

We will now bound $\Pr[(\mathrm{f}) \mid \neg(\mathrm{a}) \wedge \neg(\mathrm{b})]$. Note that $\neg$(a) implies that each $\rho(x_i)$ is distinct, because if $\neg$(a) holds then each component of ρ is injective (this is where we use the fact that M is invertible). Also note that if $\neg$(b) holds, then each block of each $(\rho(x_i) + k_{r-2})$ will be a yet-undetermined S-box input.

Choose any $i \neq i' \le q$ and $\ell, \ell' \le m$. We will show that $\Pr[z_i^{(\ell)} = z_{i'}^{(\ell')} \mid \neg(\mathrm{a}) \wedge \neg(\mathrm{b})] < 4m \cdot 2^{-b}$, and then a union bound over i, i', ℓ, ℓ' gives $\Pr[(\mathrm{f}) \mid \neg(\mathrm{a}) \wedge \neg(\mathrm{b})] < 4m^3 q^2 \cdot 2^{-b}$. (We remark that if $\ell \neq \ell'$ then one can use the same argument that was used to bound (e); however more is needed when $\ell = \ell'$, and the following argument works for either case.) We have $z_i^{(\ell)} = z_{i'}^{(\ell')}$ iff

$$k_{r-1}^{(\ell)} + \sum_{s=1}^{m} M_{\ell,s} \cdot S(\rho(x_i)^{(s)} + k_{r-2}^{(s)}) \;=\; k_{r-1}^{(\ell')} + \sum_{s=1}^{m} M_{\ell',s} \cdot S(\rho(x_{i'})^{(s)} + k_{r-2}^{(s)}). \tag{1}$$

Let $t \in [m]$ be such that $\rho(x_i)^{(t)} \neq \rho(x_{i'})^{(t)}$, which must exist because $\rho(x_i) \neq \rho(x_{i'})$. Arbitrarily fix k_{r-1}. Arbitrarily fix the values of $\{k_{r-2}^{(s)}\}_{s \neq t}$ in a way which does not violate $\neg$(b). Arbitrarily fix the output of S on the inputs $\left\{(\rho(x_i)^{(s)} + k_{r-2}^{(s)}),\, (\rho(x_{i'})^{(s)} + k_{r-2}^{(s)})\right\}_{s \neq t}$. Then, there is a fixed $\alpha \in \mathrm{GF}(2^b)$ such that (1) holds iff

$$M_{\ell,t} \cdot S(\rho(x_i)^{(t)} + k_0^{(t)}) \;+\; M_{\ell',t} \cdot S(\rho(x_{i'})^{(t)} + k_0^{(t)}) \;=\; \alpha. \tag{2}$$

Provided that $x_i^{(t)} + k_{r-2}^{(t)}$ and $x_{i'}^{(t)} + k_{r-2}^{(t)}$ do not collide with the portion of S that has been fixed, this equation holds with probability $= 2^{-b}$. Indeed, let ξ denote the event such that

$$\left\{(x_i^{(t)} + k_{r-2}^{(t)}),\ (x_{i'}^{(t)} + k_{r-2}^{(t)})\right\} \bigcap \left(\left\{(x_i^{(s)} + k_{r-2}^{(s)}),\ (x_{i'}^{(s)} + k_{r-2}^{(s)})\right\}_{s \neq t} \bigcup H\right) \neq \emptyset$$

and note that $\Pr_{k_{r-2}^{(t)}}[\xi \mid \neg(\mathrm{a}) \wedge \neg(\mathrm{b})] \le 2 \cdot 2(m-1) \cdot 2^{-b}$ by a union bound, because $\neg$(b) ensures that the intersection with H is empty. Then, because $M_{\ell,t}$ and $M_{\ell',t}$ are non-zero, we have $\Pr_{S, k_{r-2}^{(t)}}[(2) \mid \neg\xi] = 2^{-b}$ (where the choice of S is over the outputs that are not fixed), and thus $\Pr_{S, k_{r-2}, k_{r-1}}[(1) \mid \neg(\mathrm{a}) \wedge \neg(\mathrm{b})] < 4m \cdot 2^{-b}$ (where S, k_{r-2}, and k_{r-1} are completely uniform) as promised.

Finally, we have

$$\Pr[(\mathrm{a}) \vee \cdots \vee (\mathrm{f})] \le \Pr[(\mathrm{a})] + \cdots + \Pr[(\mathrm{e})] + \Pr[(\mathrm{f}) \mid \neg(\mathrm{a}) \wedge \neg(\mathrm{b})] < O(r^2 m^3 q^2) \cdot 2^{-b}.$$

$\square$

Lemma 2.3. *For any $y_1, \ldots, y_q \in (GF(2^b))^m$:*

$$\Pr_{S,k_0,\ldots,k_{r-1}} \left[\forall i \leq q : \mathcal{F}_1(x_i) = y_i \;\middle|\; \neg\text{BAD}\right] = 2^{-qmb}.$$

Proof. After running steps 1 through 5 in the process of choosing $(S, k_0, \ldots, k_{r-1})$, if we condition on $\neg$BAD then the qm elements of the set $\{z_i^{(\ell)}\}_{i,\ell}$ are unique and were not used as inputs to S in steps 2 or 4. Thus, each element has a 2^{-b} probability (independent from the other elements) of being mapped by S to the corresponding output (i.e. a block of a y_i), and the lemma follows. □

2.1.3 Stage 2

We now show that even adversaries that make adaptive queries have small distinguishing advantage, i.e. we prove Theorem 1.3. We make the standard assumption that the adversary A is deterministic, computationally unbounded, and never queries an oracle twice with the same input.

To prove Theorem 1.3, we extend the results of the previous section by considering the distribution over *transcripts* of A's interaction with its oracles. A transcript is a sequence $\sigma = [(x_1, y_1), \ldots, (x_q, y_q)]$ that contains the query/answer pairs arising from A's interaction with its oracle. We use T_F to denote the transcript of A^F, and we use $A(\sigma)$ to denote A's output after seeing transcript σ. (So note for instance that $\Pr_F[A^F = 1]$ and $\Pr_F[A(T_F) = 1]$ are semantically equivalent.)

Because A is deterministic, there is a deterministic function Q_A that determines its next query from the partial transcript so far. For a transcript σ, denote its prefixes by $\sigma_i := [(x_1, y_1), \ldots, (x_i, y_i)]$. We say a transcript σ is *possible* for A if for all $i < q$: $Q_A(\sigma_i) = x_{i+1}$. Clearly for any *im*possible transcript σ, $\Pr[T_F = \sigma] = 0$ regardless of the distribution from which F is chosen. Also note that the assumption that A never makes the same query twice implies that in any possible transcript, $x_i \neq x_j$ for all $i \neq j$.

We now prove Theorem 1.3 with a probability argument (similar to that of [NR99, Thm. 3.2]).

Proof of Theorem 1.3. Let Γ be the set of possible transcripts such that $A(\sigma) = 1 \Leftrightarrow \sigma \in \Gamma$. Then,

$$\begin{aligned}
&\left| \Pr_F \left[A^F = 1\right] - \Pr_{S,k_0,\ldots,k_{r-1}} \left[A^{\mathcal{F}_1} = 1\right] \right| \\
= \;& \left| \sum_{\sigma \in \Gamma} \left(\Pr_F[T_F = \sigma] - \Pr_{S,k_0,\ldots,k_{r-1}}[T_{\mathcal{F}_1} = \sigma] \right) \right| \\
\leq \;& \left| \sum_{\sigma \in \Gamma} \Pr_{S,k_0,\ldots,k_{r-1}}[\text{BAD}] \cdot \left(\Pr_F[T_F = \sigma] - \Pr_{S,k_0,\ldots,k_{r-1}}[T_{\mathcal{F}_1} = \sigma \mid \text{BAD}] \right) \right| \qquad (3) \\
&+ \left| \sum_{\sigma \in \Gamma} \Pr_{S,k_0,\ldots,k_{r-1}}[\neg\text{BAD}] \cdot \left(\Pr_F[T_F = \sigma] - \Pr_{S,k_0,\ldots,k_{r-1}}[T_{\mathcal{F}_1} = \sigma \mid \neg\text{BAD}] \right) \right|. \qquad (4)
\end{aligned}$$

Lemma 2.3 implies that (4) $= 0$, because $\Pr_F[T_F = \sigma] = 2^{-qmb}$ for any possible transcript σ. We rewrite (3) as

$$\left| \sum_{\sigma\in\Gamma} \left(\Pr_{S,k_0,\ldots,k_{r-1}}[\text{BAD}] \cdot \Pr_F[T_F = \sigma] \right) - \sum_{\sigma\in\Gamma} \left(\Pr_{S,k_0,\ldots,k_{r-1}}[\text{BAD}] \cdot \Pr_{S,k_0,\ldots,k_{r-1}}[T_{\mathcal{F}_1} = \sigma \mid \text{BAD}] \right) \right|.$$

Each of the two summations is bounded by $\alpha := \max_{\sigma\in\Gamma} \left(\Pr_{S,k_0,\ldots,k_{r-1}}[\text{BAD}] \right)$, since each is a convex combination of numbers that are bounded by α. Thus, the absolute value of their difference is bounded by α as well, and $\alpha < O(r^2m^3q^2) \cdot 2^{-b}$ by Lemma 2.2. $\square$

2.2 Candidate 2

Our next candidate PRF $\mathcal{F}_2 : \{0,1\}^n \to \{0,1\}^n$ is parameterized by a key k of length $O(n \log^2 n)$ and is computable by circuits of size $\leq n \log^{O(1)} n$. We will show that it has security $2^{-\Omega(n)}$ against linear and differential cryptanalysis via Theorem 2.5. The SPN defining $\mathcal{F}_2$ closely follows the Advanced Encryption Standard (AES).

Definition of $\mathcal{F}_2$. $\mathcal{F}_2$ is an SPN as defined in §1.1; our parameter choices are as follows. For any $b \in \mathbb{N}$ let $m := 2^{b-1}$, $r := \lceil b/10 \rceil$ and $n := mb$.

For the S-box, we use essentially the same function used in AES.[2] Namely, $S : \mathrm{GF}(2^b) \to \mathrm{GF}(2^b)$ is defined by $S(x) := x^{2^b-2}$. Note that $x \mapsto x^{2^b-2}$ is simply inversion in $\mathrm{GF}(2^b)$ with $0^{-1} := 0$. Recall that the bounds on p_{LC} and p_{DC} from Theorem 1.2 are stated in terms of bounds on the correlation and the DPP, respectively, of the S-box. The results of Nyberg [Nyb93] and the references therein establish these bounds, stated in following theorem.

Theorem 2.4. *Let $S : GF(2^b) \to GF(2^b)$ be defined by $S(x) := x^{2^b-2}$. Then:*

1. $\max_{\Gamma_x,\Gamma_y \neq 0} \left(\mathrm{Cor}_{\Gamma_x,\Gamma_y}(S)^2 \right) \leq 2^{b-2}$. [3]
2. $\max_{\Delta_x,\Delta_y \neq 0} \left(\mathrm{DPP}_{\Delta_x,\Delta_y}(S) \right) \leq 2^{b-2}$.

For the linear transformation $M : \mathrm{GF}(2^b)^m \to \mathrm{GF}(2^b)^m$, the crucial property is that it has maximal branch number $\mathrm{Br}(M) = m + 1$. Let G be the $2m \times m$ generator matrix of a Reed-Solomon code over $\mathrm{GF}(2^b)$. (Note that $2^b \geq 2m$ is sufficient to guarantee the existence of such a code.) Take G to be in reduced echelon form, i.e. take $G^T = [I \mid M]$ where I is the $m \times m$ identity matrix. Then, because G generates a maximum-distance-separable (MDS) code, it can be verified that the operation defined by left multiplication with M has branch number $m + 1$. This use of MDS codes to create maximal-branch-number transformations is widespread, and dates at least to [Dae95].

[2]Besides the obvious difference that in AES the value b is fixed to be 8, we omit the $\mathrm{GF}(2)^b$-affine function which is included in the AES S-box. Adding such a function would not affect the (asymptotic) circuit size of our candidates, and removing it does not affect resistance to linear/differential cryptanalysis.

[3][Nyb93] actually bounds a related quantity known as the *non-linearity* of S, but it translates directly to the stated result.

Security of $\mathcal{F}_2$. The security of $\mathcal{F}_2$ is given by the following theorem, restated for convienience.

Theorem 1.4. 1. $p_{\mathrm{LC}}(\mathcal{F}_2) \leq 2^{-\Omega(n)}$. 2. $p_{\mathrm{DC}}(\mathcal{F}_2) \leq 2^{-\Omega(n)}$.

We note that this security is "as good" as what is available for AES, i.e. AES on 128-bit inputs is believed to have security $\approx 2^{128}$. (In fact, AES's security relies on some heuristic arguments which we avoid.) Furthermore, resistance to linear and differential cryptanalysis is essentially the only type of security currently available for SPNs.

Given the choices of b, r, and m, Theorem 1.4 follows immediately from Theorem 2.4 and the following theorem which we obtain via an inductive extension of Theorem 1.2 (proof deferred to Appendix A). Moreover, by varying the constant 10 in $r := \lceil b/10 \rceil$, $p_{\mathrm{LC}}(\mathcal{F}_2)$ and $p_{\mathrm{DC}}(\mathcal{F}_2)$ can be bounded by $\leq 2^{-(1-\epsilon)\cdot n}$ for any fixed $\epsilon > 0$.

Theorem 2.5. *Let $C_k : \{0,1\}^n \to \{0,1\}^n$ be an SPN with $r = 2\ell$ rounds for some $\ell \geq 1$ and S-box S. Let $q := \max_{\Gamma_x, \Gamma_y \neq 0} \left(\mathrm{Cor}_{\Gamma_x,\Gamma_y}(S)^2\right)$ denote the maximum squared correlation of S, and let $p := \max_{\Delta_x, \Delta_y \neq 0} \left(\mathrm{DPP}_{\Delta_x,\Delta_y}(S)\right)$ denote the maximum DPP of S. If $\mathrm{Br}(M) = m+1$,*

1. $p_{\mathrm{LC}}(C_k) \leq q^{\ell m} \cdot 2^{(\ell-1)n}$. 2. $p_{\mathrm{DC}}(C_k) \leq p^{\ell m} \cdot 2^{(\ell-1)n}$.

Intuitively, the S-box provides security q (resp. p) against linear (resp. differential) cryptanalysis, and this security multiplies across "active" S-boxes (instances of S that are evaluated with a non-zero input). The branch number $\mathrm{Br}(M)$ guarantees that there exist $\geq m+1$ such active S-boxes in any pair of consecutive rounds, hence the term $q^{\ell m} = q^{(r/2)m}$. We note that the factor $2^{(\ell-1)n}$ seems to be an artifact of our extension of [KHL+01], and it is open to get a tighter bound on p_{LC} and p_{DC} for $r > 2$ rounds ([KHL+01] only consider $r = 2$). Such an extension has been considered before, for example by Keliher et al. [KMT01] and Cho et al. [CSK+04], but their results only apply in the fixed-parameter setting because they require extensive computer calculation. We are not aware of any other "closed form" bound for $r > 2$.

We also note that a bound of $p_{\mathrm{LC}}(C_k) \leq 2^{-\Omega(n)}$ incorporates the same bound on each Fourier coefficient of each output bit of C_k. In turn, this implies that each output bit depends on $\Omega(n)$ input bits. (Otherwise it can be verified that it would have too large a Fourier coefficient by Parseval's identity.)

Efficiency of $\mathcal{F}_2$. We now explain how to compute $\mathcal{F}_2$ in quasilinear size. The "tricky" component is multiplication by M. Roth and Seroussi [RS85, Theorem 1] show that when the Reed-Solomon matrix G is put into reduced echelon form, i.e. when $G^T = [I \mid M]$, then M is a generalized Cauchy matrix.

Definition 2.6. Let $\mathbb{F}$ be any field of characteristic 2. A matrix $C \in \mathbb{F}^{m\times m}$ is a *Cauchy matrix* if there exist $2m$ distinct values $\alpha_1, \dots, \alpha_m, \beta_1, \dots, \beta_m \in \mathbb{F}$ such that $C_{i,j} = (\alpha_i + \beta_j)^{-1}$. Furthermore, a matrix $M \in \mathbb{F}^{m\times m}$ is a *generalized Cauchy matrix* if it can be written as $M = BCD$, where C is a Cauchy matrix and $B, D \in \mathbb{F}^{m\times m}$ satisfy $B_{i,j} = 0 \Leftrightarrow i \neq j$ and $D_{i,j} = 0 \Leftrightarrow i \neq j$.

Gerasoulis [Ger88] shows that multiplication of a vector by an $m \times m$ Cauchy matrix can be done with $\widetilde{O}(m)$ operations when the underlying field is $\mathbb{C}$. (Multiplication with B and D in the above definition can be done with $O(m)$ operations, so we will focus on multiplication by C.) This algorithm can also be made to work over $\mathrm{GF}(2^b)$, as we now show. We stress that we are using the same algorithm from [Ger88]; the purpose here is to show that it works over $\mathrm{GF}(2^b)$.

Theorem 2.7. *Let $C \in GF(2^b)^{m\times m}$ be a Cauchy matrix defined by the (distinct) elements $\{\alpha_j, \beta_j\}_{j\in[m]}$. Then, given any vector $z \in GF(2^b)^m$, the product $C \cdot z$ can be computed with $O(m \cdot \log^2 m \cdot \log\log m)$ operations over $GF(2^b)$.*

Proof. Define the following polynomial.

$$f(x) := \sum_{j=1}^{m} z_j (x+\beta_j)^{2^b-2}$$

Then we have $C \cdot z = (f(\alpha_1), \ldots, f(\alpha_m))$, and so it suffices to evaluate f at the points $\{\alpha_i\}_i$. Now define the following three polynomials.

$$\begin{aligned}
g(x) &:= \prod_{j=1}^{m} (x+\beta_j) \\
h(x) &:= \sum_{i=1}^{m} \left[z_i (x+\beta_i)^{2^b-1} \cdot \prod_{j\neq i} (x+\beta_j) \right] \\
h_*(x) &:= \sum_{i=1}^{m} z_i \prod_{j\neq i} (x+\beta_j)
\end{aligned}$$

Then we have $f(x) = h(x)/g(x)$ as formal polynomials. Furthermore, for any $y \notin \{\beta_j\}_j$ we have $h(y) = h_*(y)$, using the identity $y^{2^b-2} = 1$ valid for any $y \neq 0$. Since our goal is to evaluate $f(\alpha_i)$ for all i, this is now seen to be equivalent to evaluating $h_*(\alpha_i)/g(\alpha_i)$ because $\alpha_i \neq \beta_j$ for all i, j.

Notice that, for each β_j, we have $h_*(\beta_j) = z_j \cdot g'(\beta_j)$, where $g'(x) = \sum_{i\in[m]} \prod_{j\neq i} (x+\beta_j)$ is the derivative of g. So, another way to view $h_*(x)$ is that it is the unique degree $\leq m-1$ polynomial interpolating the points $\{(\beta_j,\ z_j \cdot g'(\beta_j))\}_{j\in[m]}$. The algorithm is now the following:

1. Compute $g(x)$ and $g'(x)$ in coefficient form.
2. Evaluate $g'(\beta_j)$ for each β_j.
3. Compute all values of $z_j \cdot g'(\beta_j)$.
4. Interpolate the points $\{(\beta_j, z_j \cdot g'(\beta_j))\}$ to obtain $h_*(x)$ in coefficient form.
5. Evaluate both $g(\alpha_j)$ and $h_*(\alpha_j)$ for each α_j.

6. Compute each value of $f(\alpha_j) = h_*(\alpha_j)/g(\alpha_j)$.

We note that steps 1 and 2 do not involve the vector z and thus can be pre-processed, and that steps 3 and 6 can easily be done with m operations over $\mathrm{GF}(2^b)$ each. For the remaining steps, we use the following results which can be found in (for example) [vzGG03, Ch. 10] and which hold for any commutative ring with unity R.

Theorem 2.8 ([vzGG03], Corollary 10.8)**.** *Evaluation of a polynomial in $R[x]$ of degree $< m$ at m points can be done with $O(m \cdot \log^2 m \cdot \log\log m)$ operations in R.*

Theorem 2.9 ([vzGG03], Corollary 10.12)**.** *Given m distinct values $u_1, \ldots, u_m \in R$ and m arbitrary values $v_1, \ldots, v_m \in R$, the unique polynomial in $R[x]$ of degree $< m$ which interpolates $\{(u_i, v_i)\}_i$ can be computed in coefficient form with $O(m \cdot \log^2 m \cdot \log\log m)$ operations in R.*

As a result, steps 4 and 5, and thus the entire multiplication by C, can be performed with the stated number of operations in $\mathrm{GF}(2^b)$. □

One round of $\mathcal{F}_2$ consists of the following three steps:

(1) m parallel instances of exponentiation in $\mathrm{GF}(2^b)$ (i.e. $x \mapsto x^{2^b-2}$).

(2) One instance of multiplication by $M \in \mathrm{GF}(2^b)^{m\times m}$.

(3) One instance of the round key XOR.

Because finite field arithmetic and affine transformations are computable by polynomial size circuits, step (1) can be computed by a circuit with at most $m \cdot b^{O(1)}$ wires. For step (2), we have size at most $m \cdot \log^3 m \cdot b^{O(1)}$ by Theorem 2.7. Step (3) can clearly be done with $O(mb)$ wires. Thus, given the choices of m, b, and r above, the r rounds of $\mathcal{F}_2 : \{0,1\}^n \to \{0,1\}^n$ are computable by a circuit of size $n \cdot \log^{O(1)} n$, and the key size is $|k| = mbr = O(n \log^2 n)$.

2.3 Candidate 3

In this section we define a candidate PRF $\mathcal{F}_3 : \{0,1\}^n \to \{0,1\}$ parameterized by a key of length $O(n)$ and computable by TC^0 circuits of size $O(n^{1+\epsilon})$ for arbitrarily small constant $\epsilon > 0$. The construction is again inspired by the SPN structure, and the S-box S is defined identically to that of $\mathcal{F}_2$, but the linear transformation M takes a somewhat different form.

The linear transformation M. M is constructed using a good error correcting code as before; specifically, we use codes given by the following theorem, which follows from [GHK+11, Theorem 1].

Theorem 2.10. *For any constant $\epsilon > 0$, there exist constants $c, \delta > 0$ such that for sufficiently large ℓ, there exists a linear code $C_\epsilon : \{0,1\}^{\ell/c} \to \{0,1\}^\ell$ which has distance $\geq \delta \cdot \ell$ and is computable by a TC^0 circuit of size $O(\ell^{1+\epsilon})$.*

Rather than using a portion of C_ϵ's generator matrix as with $\mathcal{F}_2$ however, M consists of the entire matrix that generates C_ϵ. As a result, the internal state grows by a factor of c during each round, and thus the M used at round i will be a $c^i n \times c^{i-1} n$ matrix.

To see the advantage that this has over the previous choice of M, consider an input vector to M in which all b-bit bundles are non-zero. If we use only the fact that M has maximal branch number (Definition 1.1), then we are only guaranteed that M's output will have *one* non-zero bundle. However if we instead take $M = C_\epsilon$, then we are guaranteed that at least $\delta \cdot m$ of the output bundles will be non-zero (where $n = mb$), even if all input bundles were non-zero.

Definition of $\mathcal{F}_3$. Let $m, b, r \in \mathbb{N}$ be arbitrary for now, and set $n := mb$. Fix any $\epsilon > 0$; let c, δ, C_ϵ be given by Theorem 2.10, and for $1 \leq i \leq r$ let $M^{(i)}$ be the matrix that generates C_ϵ when $\ell := c^i n$ in Theorem 2.10. Let $k = (k_0, \ldots, k_{r+1})$ denote the key of $\mathcal{F}_3$, where $|k_i| = c^i n$ for $0 \leq i \leq r$ and $|k_{r+1}| = |k_r| = c^r n$.

$\mathcal{F}_3 : \{0,1\}^n \to \{0,1\}$ is computed over r rounds. For $1 \leq i \leq r$, round i maps $\{0,1\}^{c^{i-1}n}$ to $\{0,1\}^{c^i n}$, and is computed over three steps: (1) $c^{i-1}m$ parallel applications of S; (2) application of $M^{(i)}$ to the entire state; (3) XOR of the entire state with the round key k_i. (Note that this is exactly the same structure as an SPN, except that M is no longer a permutation, though it is still injective.)

On input x, $\mathcal{F}_3(x)$ gives $x \oplus k_0$ as input to the first round; the output of round i becomes the input to round $i+1$ (for $1 \leq i < r$), and $\mathcal{F}_3(x)$ outputs $\langle y, k_{r+1} \rangle \in \{0,1\}$ where y denotes the output of round r. (The inner product with k_{r+1} is the second way in which this candidate deviates from the SPN structure.)

Efficiency of $\mathcal{F}_3$. We now show that $\mathcal{F}_3$ can be computed by TC^0 circuits of size $O(n^{1+\epsilon})$. For $1 \leq i \leq r$, round i consists of the following:

(1) $c^{i-1}m$ parallel instances of exponentiation in $GF(2^b)$ (i.e. $x \to x^{2^b-2}$).

(2) One instance of multiplication by $M^{(i)}$.

(3) One instance of the round key XOR.

Step (1) is computable by a TC^0 circuit of size $c^{i-1} \cdot m \cdot b^{O(1)}$, using the technique described in §1.2. Step (2) is computable with size $O(c^{(1+\epsilon)i} \cdot n^{1+\epsilon})$ by Theorem 2.10. Step (3) can be computed with size $O(c^i n)$. The final inner product with k_{r+1} can be computed with size $O(c^r n)$.

Putting it together, there exists a constant κ such that the entire function can be computed by a threshold circuit of size $O\left(r \cdot (c^r m b^\kappa + c^{(1+\epsilon)r} n^{1+\epsilon})\right)$ and depth $O(r)$. Let $b \in \mathbb{N}$ be sufficiently large, and set $m := \lceil b^{(\kappa-\epsilon-1)/\epsilon} \rceil$ and $r := \lceil \kappa/\epsilon \rceil$. With $n := mb$, this ensures that $mb^\kappa \leq n^{1+\epsilon}$ (and also that $b^r \geq n$). Thus, the entire function is indeed computable by a TC^0 circuit of size $O(n^{1+\epsilon})$, where both the depth and the hidden constant depend on ϵ.

Security of $\mathcal{F}_3$. Here we are able to leverage techniques from differential cryptanalysis to prove that $\mathcal{F}_3$ is almost 3-wise independent. (Specifically, the proof below uses a technique from Nyberg's proof of Theorem 2.4. [Nyb93])

Theorem 1.6. *$\mathcal{F}_3$ is $(3, 2^{-\Omega(n)})$-wise independent.*

Proof. We will show that $\mathcal{F}_3$ is a 3-wise $2^{-\Omega(n)}$-bias generator, i.e. that for every $d \leq 3$ and any distinct $x_1, \ldots, x_d \in \{0,1\}^n$, $|\Pr_k[\sum_i \mathcal{F}_3(x_i) = 0] - 1/2| < 2^{-\Omega(n)}$. By a well-known fact (cf. [AGHP92, Lemma 1]) this implies the theorem.

For any input x, let $\mathcal{F}_3^*(x) \in \{0,1\}^{c^r n}$ denote the state just before the final inner product; that is, $\mathcal{F}_3(x) = \langle \mathcal{F}_3^*(x), k_{r+1}\rangle$. Then, $\sum_i \mathcal{F}_3(x_i) = \langle \sum_i \mathcal{F}_3^*(x_i), k_{r+1}\rangle$. So, we will show that for any $d \leq 3$ and any distinct $x_1, \ldots, x_d \in \{0,1\}^n$

$$\Pr_{(k_0,\ldots,k_r)}\left[\sum_i \mathcal{F}_3^*(x_i) = 0^{c^r n}\right] < 2^{-\Omega(n)}$$

which will complete the proof because $z \neq 0 \Rightarrow \Pr_{k_{r+1}}[\langle z, k_{r+1}\rangle = 0] = 1/2$.

For $d = 2$, this probability is 0 simply because $x_1 \neq x_2 \Rightarrow \mathcal{F}_3^*(x_1) \neq \mathcal{F}_3^*(x_2)$ because each component of $\mathcal{F}_3$ is injective.

Fix distinct $x_1, x_2, x_3 \in \{0,1\}^n$. Fix any values for $(k_0, \ldots, k_{r-2})$, the round keys used prior to round $r-1$. Let $y_i \in \{0,1\}^{c^{r-1}n}$ denote the state of the computation of $\mathcal{F}_3(x_i)$ immediately prior to the XOR with round key k_{r-1} in round $r-1$, and let $\Delta_i := y_1 \oplus y_i$ denote the differences of the y_i. Let z_1, z_2, z_3 be jointly-distributed random variables, over a uniform choice of k_{r-1}, defined by $z_i := y_i \oplus k_{r-1}$; note that (z_1, z_2, z_3) is uniformly distributed over all tuples with differences Δ_i.

Fix any $j \leq m$, and let $\overline{z}_1$ denote the jth bundle of z_1 and $\overline{\Delta}_i$ denote the jth bundle of Δ_i. We wish to bound

$$\Pr_{k_{r-1}}\left[(\overline{z_1})^{2^b-2} + \left(\overline{z_1} + \overline{\Delta}_2\right)^{2^b-2} + \left(\overline{z_1} + \overline{\Delta}_3\right)^{2^b-2} \;=\; 0\right] \tag{5}$$

the probability that the outputs of the jth S-box in round r sum to 0. If $\overline{\Delta}_1 = \overline{\Delta}_2 = 0$, then the equation is satisfied iff $\overline{z}_1 = 0$, in which case (5) $= 2^{-b}$. Now assume that at least one of $\overline{\Delta}_1, \overline{\Delta}_2$ are not zero. If we assume that $\overline{z}_1 \notin \{0, \overline{\Delta}_1, \overline{\Delta}_2\}$, then we may multiply both sides of the equation by $\prod_i \left(\overline{z}_1 + \overline{\Delta}_i\right) \neq 0$ to get a quadratic polynomial in $\overline{z}_1$. Thus, there are at most 5 values of $\overline{z}_1$ for which the equation is satisfied (including $\{0, \overline{\Delta}_1, \overline{\Delta}_2\}$), so we can bound (5) $< 6/2^b$.

Finally, because each bundle of k_{r-1} is chosen independently, and because the remaining steps in round r are linear, we have

$$\Pr_{(k_0,\ldots,k_r)}\left[\sum_{i\leq 3} \mathcal{F}_3^*(x_i) = 0^{c^r n}\right] < \left(\frac{6}{2^b}\right)^{c^{r-1}m} = 2^{-\Omega(n)}. \quad \square$$

Note that this proof does not use any properties of the code C_ϵ aside from injectivity. We remark why this proof does not show that $\mathcal{F}_3$ is almost d-wise independent for $d \geq 4$. When the number of inputs d is even, the equation in (5) is satisfied for all values of $\overline{z}_1$ iff $\overline{\Delta}_1, \ldots, \overline{\Delta}_d$ can be partitioned into $d/2$ pairs such that the two values in each pair are equal. Indeed, for even $d \in (2, 2^b]$ it is possible to construct a set $\{\Delta_1, \ldots, \Delta_d\}$ which admits such a partition for all j and yet satisfies the minimum-distance property of C_ϵ (which guarantees that $\geq \delta c^{r-1} m$ bundles of each Δ_i are non-zero for $i > 1$, and further that $\geq \delta c^{r-1} m$ bundles of $(\Delta_i \oplus \Delta_j)$ are non-zero for all $i \neq j$). However, it seems counterintuitive that that the differences at round r would satisfy such a specialized property with noticeable probability, and we believe that this proof can be extended to higher values of d.

2.4 Candidate 4

For our next candidate, we choose the extreme setting of $b = n$ and $r = 1$, which means that the function is computed over one round and essentially consists of just a single S-box. More specifically, the function is indexed by a seed $(k_0, k_1) \in \{0,1\}^{2n}$, and is computed as

$$\mathcal{F}_4(x) := (x + k_0)^{2^n - 2} + k_1.$$

Though $\mathcal{F}_4$ does indeed preserve resistance to differential and linear cryptanalysis, we note that the seed can be recovered with four known plaintext/ciphertext pairs, using an attack similar in spirit to the so-called interpolation attack of [JK01].

Claim. *Let $\mathcal{F}_4$ be the above function indexed by $k_0, k_1 \in \{0,1\}^n$. Let $\{(p_i, c_i)\}_{1 \leq i \leq 4}$ be any set such that $c_i = \mathcal{F}_4(p_i)$ for all i and $p_i \neq p_j$ for $i \neq j$. Then, with probability $(1 - 1/2^{n-2})$ over k_0, the values of k_0 and k_1 can be recovered from $\{(p_i, c_i)\}_i$.*

Proof. The attack is performed by using the four pairs to create two equations over $\mathrm{GF}(2^n)$ that are linear in the seed, as follows. Assume that $k_0 \notin \{p_i\}_i$, which happens with probability $(1 - 1/2^{n-2})$. Then the equation

$$(c_i + k_1) \cdot (p_i + k_0) = 1$$

holds for $1 \leq i \leq 4$. We can rewrite these equations as

$$k_0 k_1 + c_i k_0 + p_i k_1 + c_i p_i = 1. \tag{6}$$

If we sum (6) for $i = 1, 2$, the quadratic terms cancel and we obtain

$$(c_1 + c_2) k_0 + (p_1 + p_2) k_1 + (c_1 p_1 + c_2 p_2) = 0.$$

Summing (6) for $i = 3, 4$ gives another linear equation in k_0, k_1. The attack concludes by solving the two linear equations. $\square$

The function $\mathcal{F}_4$ can be seen as a concrete instantiation of the Even-Mansour cipher [EM97] where the random permutation is replaced with (the asymptotic version of) the AES S-box. This cipher is easily breakable as we have just observed, but we now consider a slight modification to $\mathcal{F}_4$ that is not susceptible to this simple attack, and furthermore is a small-bias generator. The modified function $\mathcal{F}'_4 : \{0,1\}^n \to \{0,1\}$ is defined as follows:

$$\mathcal{F}'_4(x) := \left\langle (x+k_0)^{2^n-2}, k_1 \right\rangle.$$

In other words, we combine the AES S-box with the Goldreich-Levin hardcore predicate [GL89]. Note that we now output only a single bit. This modification – replacing the second XOR with an inner product – can also be applied to the Even-Mansour cipher. We consider it an interesting question to what extent the assumptions necessary for the pseudorandomness of Even-Mansour can be relaxed in this setting. (In their setting, the assumption is that all parties have oracle access to a truly random permutation.)

The next theorem shows that $\mathcal{F}'_4$ is a small-bias generator. This result is reminiscent of the "exponentiation" small-bias generator in [AGHP92], where the x-th output bit is $\langle k_0^x, k_1 \rangle$. Indeed, our proof is inspired by theirs. However we face the extra difficulty that the polynomials we work with are not of low degree.

Theorem 1.7. *For any choice of $d \le 2^n$, $\mathcal{F}'_4$ is a d-wise small-bias generator with error $d/2^n$: for any distinct $a_1, \ldots, a_d \in \{0,1\}^n$,*

$$\left| \Pr_{k_0,k_1}\left[\sum_{i=1}^{d} \mathcal{F}'_4(a_i) = 0 \right] - \frac{1}{2} \right| < \frac{d}{2^n}.$$

Proof. Fix any distinct choices of $a_1, \ldots, a_d$. Then, identifying elements of $GF(2^n)$ with elements of $\{0,1\}^n$, we have

$$\begin{aligned} \sum_{i \le d} \mathcal{F}'_4(a_i) &= \sum_{i\le d} \left\langle (a_i+k_0)^{2^n-2}, k_1 \right\rangle \\ &= \left\langle p(x) := \sum_{i \le d} (a_i + k_0)^{2^n-2},\ k_1 \right\rangle. \end{aligned}$$

We now show that the polynomial $p(x) = \sum_{i\le d}(a_i + x)^{2^n-2}$ has at most $2d-1$ distinct roots. This will conclude the proof because when k_0 is not a root of $p(x)$, we have $\Pr_{k_1}\left[\langle p(k_0), k_1\rangle = 0\right] = 1/2$. Therefore,

$$\left| \Pr_{k_0,k_1}\left[\sum_{i=1}^{d} \mathcal{F}'_4(a_i) = 0 \right] - \frac{1}{2} \right| \le \frac{1}{2} \Pr_{k_0}[p(k_0) = 0] < \frac{d}{2^n}.$$

To show the bound on the number of roots, define the following polynomials:

$$\begin{aligned} \overline{p}(x) &:= p(x) \cdot \prod_{i \le d}(a_i + x) = \sum_{i \le d}\left[(a_i + x)^{2^n-1} \prod_{j \ne i}(a_j + x) \right], \\ \overline{p}_*(x) &:= \sum_{i \le d} \prod_{j \ne i} (a_j + x). \end{aligned}$$

Observe that any root y of $p(x)$ is also a root of $\overline{p}(x)$. Moreover, note for any $y \notin \{a_j : j \le d\}$, $\overline{p}(y) = \overline{p}_*(y)$, using the identity $y^{2^b-2} = 1$ valid for any $y \ne 0$.

Also observe that $\overline{p}_*(x)$ is not identically zero. Indeed, by inspection, the constant term of the polynomial $\overline{p}_*(x + a_1)$ is $\prod_{j \ne 1}(a_j + a_1)$, which is non-zero because the a_j are distinct; therefore $\overline{p}_*(x + a_1)$ is not identically zero, and so neither is $\overline{p}_*(x)$. Since $\overline{p}_*(x)$ is a non-zero polynomial of degree $d - 1$, it has at most $d - 1$ distinct roots.

So, if $p(x)$ has r roots, also $\overline{p}$ has r roots. At least $r - d$ of these do not belong to $\{a_j : j \le d\}$, and so they are also roots of $\overline{p}_*(x)$. Therefore, $r - d \le d - 1$, or $r \le 2d - 1$. $\square$

By Braverman's result [Bra09] (cf. [Baz09, Raz09]), we obtain that $\mathcal{F}_4'$ also fools small-depth AC^0 circuits of any size $w = 2^{n^{o(1)}}$ (that look at only w fixed output bits of the candidate).

Indeed, fix any function $w = 2^{n^{o(1)}}$ and any constant $d = O(1)$; let $N := 2^n$. By Theorem 1.7, any w output bits have bias $< w/N$. By [AGM03], for any $k \le w$, the output distribution on those w bits is $w^k w/N$-close to a k-wise independent distribution. By [Bra09], $k = \lg^{O(d^2)} w \le n^{o(1)}$ is sufficient to fool circuits of depth d with error $1/w$. Hence the overall error will be $1/w = 1/2^{n^{o(1)}}$ plus

$$\frac{w^k w}{N} = \frac{\left(2^{n^{o(1)}}\right)^{n^{o(1)}}}{N} \le \frac{1}{\sqrt{N}},$$

for a total of $1/w + 1/\sqrt{N} = O(1/w)$.

Efficiency. As noted in §1.2, $\mathcal{F}_4'$ is computable by Boolean circuits of size $\widetilde{O}(n^2)$ and TC^0 circuits of size $n^{O(1)}$.

2.5 Candidate 5

Our final candidate $\mathcal{F}_5$ preserves the structure of AES almost exactly. For any n that is a multiple of 32, we set $b = 8$, $m = n/8$, and $r = n$, and we use $S(x) := x^{2^b-2}$. The linear transformation M is of a slightly different form than that of the previous candidates, which we explain now.

M is computed in two (linear) steps. In the first step, a permutation $\pi : [m] \to [m]$ is used to shuffle the b-bit bundles of the state; namely, bundle i moves to position $\pi(i)$. π is computed as follows. First, the m bundles are placed column-wise into a $4 \times m/4$ matrix. Then row i of the matrix ($0 \le i < 4$) is shifted circularly to the left by i places, and finally the bundles are extracted column-wise from the new matrix.

In the second step, a maximal-branch-number matrix $\phi \in \mathrm{GF}(2^8)^{4\times4}$ is applied in parallel to each consecutive group of 4 bundles.

Efficiency: small circuits. In each round, the $O(n)$ instances of S and ϕ each perform computations on a constant number of bits; because permuting the bundles and adding the round key can also be done with $O(n)$ wires, each round of $\mathcal{F}_5$ can be computed by a circuit of depth $d = O(1)$ and size $w = O(n)$. Thus the entire (r-round) circuit for $\mathcal{F}_5$ has depth $d = O(n)$ and size $w = O(n^2)$.

Efficiency: fast Turing machines. Similarly, for any fixed seed k, each round of $\mathcal{F}_5$ can be computed in time $O(n)$ on a single-tape Turing machine with $O(n^2)$ states. To do so, we encode the bundles on the tape so that the matrix used by π is written column-wise. As before, the $O(n)$ instances of S and ϕ in a single round can be done in time $O(n)$. To see that π can also be computed in time $O(n)$, note that due to the column-wise representation each bundle needs to move ≤ 3 places away, except for the 6 bundles which are shifted circularly to the other end of the tape. Finally, encoding the $O(n^2)$-bit seed in the TM's state transitions, the addition of each round key also takes time $O(n)$. Therefore, the $r = n$ rounds of $\mathcal{F}_5$ can be computed in time $O(n^2)$.

Alternatively, consider the Turing machine variant with two tapes, in which the first tape is read-only and contains the n-bit input followed by the $n(n+1)$-bit seed, the second tape is read/write, and the TM has $O(1)$ states. Then $\mathcal{F}_5$ can again be computed in time $O(n^2)$ exactly as described above, because in round i only bits $in + 1, \ldots, in + n$ of the seed are used.

3 Conclusion

We believe a good candidate PRF should be the *simplest* candidate that resists known attacks. As noted in [DR02], some of the choices in the design of AES are not motivated by any known attack, but are there as a safeguard (for example, one can reduce the number of rounds and still no attack is known). While this is comprehensible when having to choose a standard that is difficult to change, one can argue that a better way to proceed is to put forth the *simplest* candidate PRF, possibly break it, and iterate until hopefully converging to a PRF. We view this paper as a step in this direction.

Abstracting from the SPN structure, one may arrive to the following paradigm for constructing PRF: alternate the application of (1) an error-correcting code and (2) a bundle-wise application of any local map that has high degree over GF(2) and resists attacks corresponding to linear and differential cryptanalysis. This viewpoint may lead to a PRF candidate computable in ACC^0, since for (1) one just needs parity gates, while, say, taking parities of suitable mod 3 maps one should get a map that satisfies (2). However a good choice for this latter map is not clear to us at this moment.

Acknowledgments. We thank Guevara Noubir for helpful discussions, and Salil Vadhan for mentioning AES. We also thank the anonymous referees for very helpful feedback, including pointing us to [KHL+01].

References

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992.

[AGM03] Noga Alon, Oded Goldreich, and Yishay Mansour. Almost k-wise independence versus k-wise independence. *Inf. Process. Lett.*, 88(3):107–110, 2003.

[AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3), 2010.

[AW08] Scott Aaronson and Avi Wigderson. Algebrization: a new barrier in complexity theory. In *40th ACM Symposium on the Theory of Computing (STOC)*, pages 731–740, 2008.

[Baz09] Louay M. J. Bazzi. Polylogarithmic independence can fool dnf formulas. *SIAM J. Comput.*, 38(6):2220–2272, 2009.

[BGS75] Theodore Baker, John Gill, and Robert Solovay. Relativizations of the $P{=}?NP$ question. *SIAM J. Comput.*, 4(4):431–442, 1975.

[Bra09] Mark Braverman. Poly-logarithmic independence fools AC^0 circuits. In *24th Conference on Computational Complexity (CCC)*. IEEE, 2009.

[BS91] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

[CSK+04] Hong-Su Cho, Soo Hak Sung, Daesung Kwon, Jung-Keun Lee, Jung Hwan Song, and Jongin Lim. New method for bounding the maximum differential probability for SPNs and ARIA. In *ICISC*, pages 21–32, 2004.

[Dae95] Joan Daemen. *Cipher and hash function design strategies based on linear and differential cryptanalysis*. PhD thesis, K.U.Leuven, 1995.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer Verlag, 2002.

[EM97] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *J. Cryptology*, 10(3):151–162, 1997.

[Ger88] A. Gerasoulis. A fast algorithm for the multiplication of generalized Hilbert matrices with vectors. *Mathematics of Computation*, 50:179–188, 1988.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[GHK+11] Anna Gál, Kristoffer Arnsfelt Hansen, Michal Koucký, Pavel Pudlák, and Emanuele Viola. Tight bounds on computing error-correcting codes by bounded-depth circuits with arbitrary gates. 2011.

[GL89] Oded Goldreich and Leonid Levin. A hard-core predicate for all one-way functions. In *21st ACM Symposium on the Theory of Computing (STOC)*, pages 25–32, 1989.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.

[GR04] Craig Gentry and Zulfikar Ramzan. Eliminating random permutation oracles in the Even-Mansour cipher. In *ASIACRYPT*, pages 32–47, 2004.

[GvzGPS00] Shuhong Gao, Joachim von zur Gathen, Daniel Panario, and Victor Shoup. Algorithms for exponentiation in finite fields. *J. Symb. Comput.*, 29(6):879–889, 2000.

[HAB02] William Hesse, Eric Allender, and David A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *J. Comput. System Sci.*, 65(4):695–716, 2002. Special issue on complexity, 2001 (Chicago, IL).

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HRV10] Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *42nd ACM Symposium on the Theory of Computing (STOC)*, pages 437–446, 2010.

[HV06] Alexander Healy and Emanuele Viola. Constant-depth circuits for arithmetic in finite fields of characteristic two. In *23rd Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 672–683. Springer, 2006.

[JK01] T. Jakobsen and L.R. Knudsen. Attacks on block ciphers of low algebraic degree. *Journal of Cryptology*, 14:197–210, 2001.

[KHL+01] Ju-Sung Kang, Seokhie Hong, Sangjin Lee, Okyeon Yi, Choonsik Park, and Jongin Lim. Practical and provable security against differential and linear cryptanalysis for substitution-permutation networks. *ETRI Journal*, 23(4):158–167, 2001.

[KMT01] Liam Keliher, Henk Meijer, and Stafford E. Tavares. New method for upper bounding the maximum average linear hull probability for SPNs. In *EUROCRYPT*, pages 420–436, 2001.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[Kop11] Swastik Kopparty. On the complexity of powering in finite fields. In *Symposium on the Theory of Computing (STOC)*, 2011.

[LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.

[Mat94] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology, Proc. Eurocrypt'93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer-Verlag, 1994.

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993.

[NR99] Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *J. Cryptology*, 12(1):29–66, 1999.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudorandom functions. *J. ACM*, 51(2):231–262, 2004.

[Nyb93] Kaisa Nyberg. Differentially uniform mappings for cryptography. In *EUROCRYPT*, pages 55–64, 1993.

[Pie91] Josef Pieprzyk. On bent permutations. In *Proceedings of the International Conference on Finite Fields, Coding Theory, and Advances in Communications and Computing, Las Vegas*, August 1991.

[Raz09] Alexander A. Razborov. A simple proof of Bazzi's theorem. *ACM Transactions on Computation Theory (TOCT)*, 1(1), 2009.

[RR97] Alexander Razborov and Steven Rudich. Natural proofs. *J. of Computer and System Sciences*, 55(1):24–35, August 1997.

[RR00] Zulfikar Ramzan and Leonid Reyzin. On the round security of symmetric-key cryptographic primitives. In *CRYPTO*, pages 376–393, 2000.

[RS85] Ron M. Roth and Gadiel Seroussi. On generator matrices of MDS codes. *IEEE Transactions on Information Theory*, 31:826–830, 1985.

[Sha49] Claude Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949.

[VZ11] Salil P. Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. *Electronic Colloquium on Computational Complexity (ECCC)*, (TR11–141), November 2011.

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2003.

[Wil11] Ryan Williams. Non-uniform ACC lower bounds. In *Conference on Computational Complexity (CCC)*, 2011.

A Security against linear/differential cryptanalysis

In this section we fill in the missing details from §1.1 on how the security of an SPN is evaluated against linear and differential cryptanalysis, and we prove Theorem 2.5 via an inductive extension of the results of Kang et al. [KHL+01].

A.1 Linear cryptanalysis

Recall the following two definitions from §1.1.

$$\mathrm{Cor}_{\Gamma_x,\Gamma_y}(f) := 2 \cdot \Pr_x[\langle \Gamma_x, x\rangle = \langle \Gamma_y, f(x)\rangle] - 1$$

$$p_{\mathrm{LC}}(C_k) := \max_{\Gamma_x,\Gamma_y \neq 0} \left(\mathbb{E}_k\left[\mathrm{Cor}_{\Gamma_x,\Gamma_y}(C_k)^2\right]\right)$$

To bound $p_{\mathrm{LC}}(C_k)$, the concept of a *linear trail* is used. Let ρ_k^i denote the ith round function of an SPN C_k, i.e. $C_k(x) = \rho_k^r(\rho_k^{r-1}(\cdots(\rho_k^1(x \oplus k_0))\cdots))$. A linear trail is a vector $\Gamma = (\Gamma_0, \ldots, \Gamma_r) \in (\{0,1\}^n)^{r+1}$, and the correlation of C_k with respect to Γ is (cf. [DR02, Eqn. 7.59])

$$\mathrm{Cor}_\Gamma(C_k) := \prod_{i=1}^{r} \mathrm{Cor}_{\Gamma_{i-1},\Gamma_i}(\rho_k^i).$$

This equation is defined for a fixed key k, but in fact for SPNs only the sign of this product is affected by the value of the key [DR02, §7.9.2]. In particular, $\mathrm{Cor}_\Gamma(C_k)^2$ is the same for every key k.

For any pair of input/output parities Γ_x, Γ_y, we have the following theorem.

Theorem A.1 ([DR02], Thm. 7.9.1)**.**

$$\mathbb{E}_k\left[\mathrm{Cor}_{\Gamma_x,\Gamma_y}(C_k)^2\right] = \sum_{\Gamma:\,\Gamma_0=\Gamma_x,\Gamma_r=\Gamma_y} \mathrm{Cor}_\Gamma(C_k)^2.$$

A naïve evaluation of this sum would lead to a useless bound on p_{LC} (i.e. a bound ≥ 1) due to the large number of vectors Γ that have the specified first and final elements. Kang et al. [KHL+01] give an exponentially small bound on this sum (Theorem 1.2) in the case where $r = 2$ and the linear transformation M has maximal branch number.

A.2 Differential cryptanalysis

Recall the following two definitions from §1.1.

$$\mathrm{DPP}_{\Delta_x,\Delta_y}(f_k) := \Pr_{x,k}[f_k(x) \oplus f_k(x \oplus \Delta_x) = \Delta_y]$$

$$p_{\mathrm{DC}}(C_k) := \max_{\Delta_x,\Delta_y \neq 0} \left(\mathrm{DPP}_{\Delta_x,\Delta_y}(C_k)\right)$$

Similarly to how linear trails were used in the previous subsection, *differential trails* are used to bound $p_{\mathrm{DC}}(C_k)$. A differential trail is a vector $\Delta = (\Delta_0, \ldots, \Delta_r) \in (\{0,1\}^n)^{r+1}$. For any SPN C_k, again let ρ_k^i denote its ith round function, and let $C_k^{(i)}(x)$ denote the output of the ith round of $C_k(x)$, with $C_k^{(0)}(x) := x \oplus k_0$. That is, for any $i \leq r$, $C_k^{(i)}(x) = \rho_k^i(\rho_k^{i-1}(\cdots(\rho_k^1(x \oplus k_0))\cdots))$.

Then for any Δ_0, Δ_r, we have

$$\mathrm{DPP}_{\Delta_0,\Delta_r}(C_k) = \sum_{\Delta_1,\ldots,\Delta_{r-1}} \Pr_{x,k}\left[\bigwedge_{i=1}^{r}\left[\rho_k^i\left(C_k^{(i-1)}(x)\right) \oplus \rho_k^i\left(C_k^{(i-1)}(x \oplus \Delta_0)\right) = \Delta_i\right]\right].$$

This can be seen by noting that for any fixed values of x, k, Δ_0 and Δ_r there is at most one tuple $(\Delta_1, \ldots, \Delta_{r-1})$ for which the conjunction evaluates to true. To simplify this equation, we use the following two facts.

- The independence of the round keys ensures that, conditioned on two inputs to round i having XOR difference Δ_{i-1}, the inputs are uniformly distributed over all pairs with difference Δ_{i-1}, and are independent of the inputs to all previous rounds.

- XORing the round key does not affect the DPP of a given round. That is, letting ρ denote the round function without the key XOR, we have $\mathrm{DPP}_{\Delta_x,\Delta_y}(\rho) = \mathrm{DPP}_{\Delta_x,\Delta_y}(\rho_k^i)$ for all i, Δ_x, Δ_y.

Using these facts and an application of the chain rule, we have

$$\mathrm{DPP}_{\Delta_0,\Delta_r}(C_k) = \sum_{\Delta_1,\ldots,\Delta_{r-1}} \prod_{i=1}^{r} \mathrm{DPP}_{\Delta_{i-1},\Delta_i}(\rho).$$

Kang et al. again give an exponentially small bound on this sum when $r = 2$ (Theorem 1.2).

A.3 Proof of Theorem 2.5

We now prove Theorem 2.5 via an inductive extension of Theorem 1.2. We restate both theorems for convienience.

Theorem 1.2. ([KHL+01], Thms. 5 & 6) Let $C_k : \{0,1\}^n \to \{0,1\}^n$ be an SPN with $r = 2$ rounds and S-box S. Let $q := \max_{\Gamma_x,\Gamma_y \neq 0} \left(\mathrm{Cor}_{\Gamma_x,\Gamma_y}(S)^2\right)$ denote the maximum squared correlation of S, and let $p := \max_{\Delta_x,\Delta_y \neq 0}\left(\mathrm{DPP}_{\Delta_x,\Delta_y}(S)\right)$ denote the maximum DPP of S. If $\mathrm{Br}(M) = m+1$, then $p_{\mathrm{LC}}(C_k) \leq q^m$ and $p_{\mathrm{DC}}(C_k) \leq p^m$.

Theorem 2.5. *Let $C_k : \{0,1\}^n \to \{0,1\}^n$ be an SPN with $r = 2\ell$ rounds for some $\ell \geq 1$ and S-box S. Let $q := \max_{\Gamma_x,\Gamma_y \neq 0}\left(\mathrm{Cor}_{\Gamma_x,\Gamma_y}(S)^2\right)$ denote the maximum squared correlation of S, and let $p := \max_{\Delta_x,\Delta_y \neq 0}\left(\mathrm{DPP}_{\Delta_x,\Delta_y}(S)\right)$ denote the maximum DPP of S. If $\mathrm{Br}(M) = m+1$,*

1. $p_{\mathrm{LC}}(C_k) \leq q^{\ell m} \cdot 2^{(\ell-1)n}$.
2. $p_{\mathrm{DC}}(C_k) \leq p^{\ell m} \cdot 2^{(\ell-1)n}$.

Proof. We prove part 1; part 2 is essentially identical.

We proceed inductively on ℓ. The base case $\ell = 1$ is given by Theorem 1.2. Fix $\ell > 1$, and let $\Gamma_0, \Gamma_{2\ell}$ be any non-zero input/output parities. Then,

$$
\begin{aligned}
p_{\mathrm{LC}}(C_k) &= \sum_{\Gamma=(\Gamma_0,\ldots,\Gamma_{2\ell})} \mathrm{Cor}_{\Gamma}(C_k)^2 \\
&= \sum_{\Gamma_1,\ldots,\Gamma_{2\ell-1}} \prod_{i=1}^{2\ell} \mathrm{Cor}_{\Gamma_{i-1},\Gamma_i}(\rho_{k_i})^2 \\
&= \sum_{\Gamma_1,\ldots,\Gamma_{2\ell-2}} \prod_{i=1}^{2\ell-2} \mathrm{Cor}_{\Gamma_{i-1},\Gamma_i}(\rho_{k_i})^2 \sum_{\Gamma_{2\ell-1}} \prod_{i=2\ell-1}^{2\ell} \mathrm{Cor}_{\Gamma_{i-1},\Gamma_i}(\rho_{k_i})^2 \\
&\leq q^m \cdot \sum_{\Gamma_1,\ldots,\Gamma_{2\ell-2}} \prod_{i=1}^{2\ell-2} \mathrm{Cor}_{\Gamma_{i-1},\Gamma_i}(\rho_{k_i})^2 && (7) \\
&= q^m \cdot \sum_{\Gamma_{2\ell-2}} \left(\sum_{\Gamma_1,\ldots,\Gamma_{2\ell-3}} \prod_{i=1}^{2\ell-2} \mathrm{Cor}_{\Gamma_{i-1},\Gamma_i}(\rho_{k_i})^2 \right) \\
&\leq q^m \cdot \sum_{\Gamma_{2\ell-2}} q^{(\ell-1)m} \cdot 2^{(\ell-2)n} && (8) \\
&= q^{\ell m} \cdot 2^{(\ell-1)n} && (9)
\end{aligned}
$$

where (7) is by Theorem 1.2, (8) is by the inductive hypothesis and (9) is by the fact that there are 2^n choices for $\Gamma_{2\ell-2}$. ∎

B Distinguishing $o(n)$-degree PRFs

In this section, we show that any PRF $f_k : \{0,1\}^n \to \{0,1\}$ which is computable by an $o(n)$-degree polynomial over GF(2) cannot have hardness 2^n. This just follows from the fact that in time 2^n one can write down the polynomial representation of f restricted to $\Omega(n)$ input bits. Details follow.

For simplicity, we instead show that any such PRF can be broken in time $2^{O(n)}$. This implies the desired goal, for if we had a PRF $f_k : \{0,1\}^n \to \{0,1\}$ with hardness 2^n we could consider it over bn input bits, note that the degree would still be $o(n) = o(bn)$, and obtain a contradiction.

To start, let $f : \{0,1\}^n \to \{0,1\}$ be any function, and define the following three values:

- $T_f \in \{0,1\}^{2^n}$ is the truth table of f; i.e. $(T_f)_i := f(i)$, identifying a natural number with its binary representation.

- $C_f \in \{0,1\}^{2^n}$ is the coefficient vector of f, defined as follows. Fix some ordering on the 2^n possible multilinear monomials in n variables. Then, $(C_f)_i = 1$ iff the ith monomial appears in the polynomial representation of f over GF(2).

- $A \in \{0,1\}^{2^n \times 2^n}$ is the matrix with rows indexed by the set $\{0,1\}^n$ and columns indexed by the set of degree $\leq n$ multilinear monomials (as with C_f), defined by $A_{ij} := 1$ iff monomial j has value 1 under input i.

Note that A is independent of the function f. Furthermore, A is invertible because it has full rank, which follows from the fact that any two linear combinations of A's columns give the truth tables of two distinct polynomials. We now show how to distinguish a low-degree PRF using the fact that $A \cdot C_f = T_f$ for all f.

Theorem B.1. *Let $\{f_k : \{0,1\}^n \to \{0,1\}\}_k$ be a PRF such that, for each key k, the polynomial representation of f_k over GF(2) has degree $o(n)$. Then, there is an adversary that runs in time $\leq 2^{O(n)}$ and distinguishes f_k from random with advantage $\geq 1 - 2^{-2^{\Omega(n)}}$.*

Proof. For any function $f : \{0,1\}^n \to \{0,1\}$, we can use C_f to check if the polynomial representation of f contains a monomial of degree $\geq n/2$. Clearly this will be false for any f_k drawn from the PRF, and for a uniformly random function F we have

$$\Pr_F[F \text{ has a monomial of degree } \geq n/2] \;\geq\; 1 - 2^{-\binom{n}{n/2}} \;\geq\; 1 - 2^{-2^{\Omega(n)}}$$

which can be seen by viewing F as being randomly chosen by including each possible monomial independently with probability 1/2. Finally, note that C_f can be computed from the truth table of f in time $2^{O(n)}$ as $C_f = A^{-1} \cdot T_f$. ∎

ECCC ISSN 1433-8092
http://eccc.hpi-web.de