Substitution-Permutation Networks, Pseudorandom Functions, and Natural Proofs[*]

Eric Miles and Emanuele Viola

Northeastern University
{enmiles,viola}@ccs.neu.edu

Abstract. This paper takes a new step towards closing the troubling gap between pseudorandom functions (PRF) and their popular, bounded-input-length counterparts. This gap is both quantitative, because these counterparts are more efficient than PRF in various ways, and methodological, because these counterparts usually fit in the substitution-permutation network paradigm (SPN) which has not been used to construct 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 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 practical constructions. $\mathcal{F}_2$ 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 fools all parity tests that look at $\leq 2^{0.9n}$ outputs.

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.

R. Safavi-Naini and R. Canetti (Eds.): CRYPTO 2012, LNCS 7417, pp. 68–85, 2012.

1 Introduction

This paper takes a new step towards closing the troubling gap between pseudorandom functions ([17], cf. [16, §3.6]) and their popular, bounded-input-length counterparts. These counterparts are mostly obtained in two ways. One is to use bounded-input-length hash functions such as the SHA-1 compression function, or block ciphers such as the Advanced Encryption Standard (AES) by Daemen and Rijmen [10]. We note that the latter satisfy the additional constraint of computing permutation functions.

This gap is both quantitative and methodological. It is quantitative because all candidate pseudorandom functions (hereafter, PRF) based on complexity-theoretic assumptions (e.g. [17,21,35,20,43]) 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, bounded-input-length constructions often have seed length which *equals* the input length. This is for example the case with the 128-bit version of AES.

It is methodological because many modern bounded-input-length hash functions and block ciphers are constructed using the *substitution-permutation network* (SPN) paradigm. This is for example the case with two of the finalists for the ongoing SHA-3 cryptographic hash function competition, namely Grøstl [13] and JH [45], and also the AES block cipher. An SPN is computed over a number of rounds, where each round "confuses" the input by dividing it into bundles and applying a substitution function (S-box) to each bundle, and then "diffuses" the bundles by applying a matrix with certain "branching" properties (cf. [42]). 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 [31] 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, though unlike popular constructions we do not require that an SPN computes a permutation function. Each of the many hash functions and block ciphers based on the SPN structure (e.g. those mentioned above) suggests different choices for the parameters, S-boxes, and diffusion matrices. As a first step we choose to follow the design considerations behind the AES block cipher, and particularly its S-box. We do this for two reasons. First, it is a well-documented, widely-used block cipher that has been around for over a decade. Second, the algebraic structure of its S-box lends itself to an asymptotic generalization; we exploit this fact in some of our results. We hope that future work will systematically address other available bounded-input-length constructions.

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 [34] 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 $O(n \log^2 n)$. 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. [26].
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 [11] that uses the AES S-box in place of a random permutation. We prove that this candidate fools all parity tests that look at $\leq 2^{0.9n}$ outputs.
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.

For context, we mention that Hoory, Magen, Myers and Rackoff [24] and Brodsky and Hoory [8], building on work by Gowers [19], study the random composition of a family of permutations. The SPN structure can be seen as falling into this framework, by taking each round as an individual permutation chosen randomly by the key. However, the permutations constructed in these works do not have the form of an SPN round, and furthermore the circuit complexity of the composed permutations is not of interest to them (their constructions have size and depth $\Omega(n^3)$).

Natural Proofs. The landscape of circuit lower bounds remains bleak, despite exciting recent results [44]. 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 [4,39,1].

Of particular interest to us is the Natural Proofs work by Razborov and Rudich [39]. 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. [17,21,35,20,43]) 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 [35] 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 [2] 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 [2] 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. [30, §12.2].

Assuming the (exponential) security of some 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). The notation introduced in this section will be used throughout the paper.

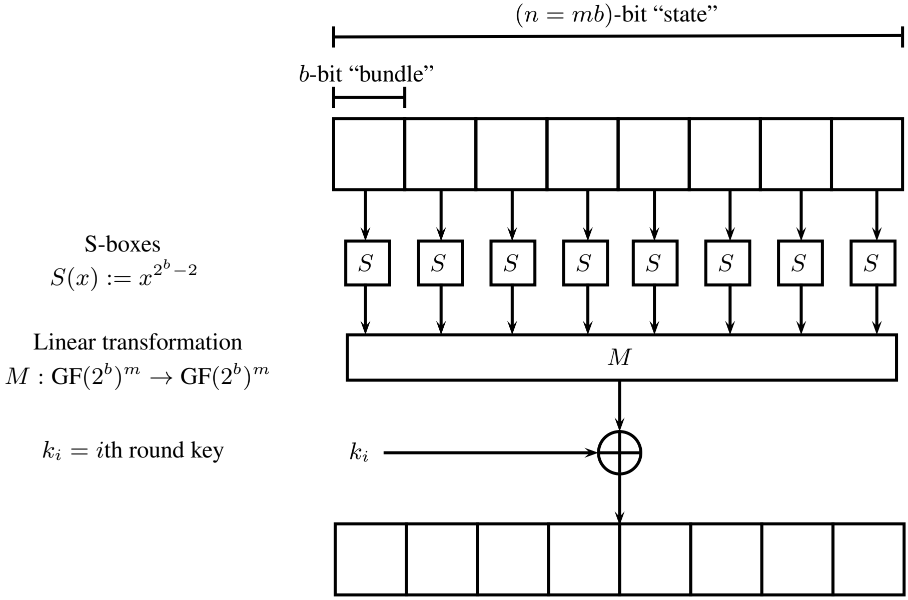

Fig. 1. One round of an SPN

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 input size*

- $m \in \mathbb{N}$, the *number of S-box invocations per round*
- $S : \mathrm{GF}(2^b) \to \mathrm{GF}(2^b)$, the *S-box*
- $M : \left(\mathrm{GF}(2^b)\right)^m \to \left(\mathrm{GF}(2^b)\right)^m$, 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 $\mathrm{GF}(2^b)$.

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.

Security against Linear and Differential Cryptanalysis. We now briefly review how the security of an SPN is evaluated against two general attacks on block ciphers: linear and differential cryptanalysis. (See the full version for a more extensive discussion.) Resistance to these attacks is typically seen as the main security feature of SPNs. Note that we consider here the basic versions of these attacks, and we leave to future work understanding the resistance of our candidates to more sophisticated attacks (such as those considered by Knudsen [28]).

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. *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 [32] 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

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

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

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

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

[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.

Differential cryptanalysis [6] 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. [26], gives a bound on p_{LC} and p_{DC} for 2-round SPNs with maximal branch number.

Theorem 1. ([26], 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 $p_{\mathrm{DC}}(C_k) \leq p^m$.*

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 exponential in $n = mb$. (For completeness we note that one cannot directly apply the above theorem to AES because it is a more complicated SPN.)

We extend this result to $r > 2$ rounds in the following theorem.

Theorem 2. *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 [26], and it is open to get a tighter bound on p_{LC} and p_{DC} for $r > 2$ rounds ([26] only consider $r = 2$). Such an extension has been considered before, for example by Keliher et al. [27] and Cho et al. [9], 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 $r > 2$.

Security against Degree-Exploiting Attacks. 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 [10, p. 81]), considerations are usually also taken to prevent attacks that would exploit algebraic structure in the cipher. In our candidates 2-5, we adopt essentially the same S-box that is used in AES.[2] This S-box 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). Specifically, 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 [37,36,29].

We need the degree of each output bit 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, showing that a PRF that has degree $o(n)$ cannot have hardness 2^n.

Theorem 3. *Let $F = \{f_k : \{0,1\}^n \to \{0,1\}\}_k$ be any set of functions 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 a random $f_k \in F$ from a random function with advantage $\geq 1 - 2^{-2^{\Omega(n)}}$.*

The only non-linear operation in the entire cipher is the S-box, which for Candidates 2-5 has degree $b-1$, and thus the maximum possible degree of each output bit for these candidates is $(b-1)^r$. 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 $r = 1$.) We do not know if $b^r \geq n$ is sufficient to guarantee degree $\Omega(n)$, and it is an interesting research direction to understand what restrictions (if any) on the SPN parameters ensure that the function has high degree.

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 note that there is often an intimate connection between the two types, as many key recovery techniques, including linear and differential cryptanalysis, construct a distinguishing algorithm which is then used to select the correct round keys from a set of potential keys.

[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 that 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. To our knowledge there are no known attacks against the AES variant that uses this "reduced" S-box.

2 Our Candidates

We now describe our candidates. Candidates 1, 2, and 5 output n bits, while Candidates 3 and 4 output 1 bit. We use $\mathcal{F}_i$ to refer to the function computing Candidate i. In each candidate, the $(r+1)$ n-bit round keys are chosen independently and uniformly at random. (Popular constructions typically employ a so-called "key schedule" that generates the round keys from a key of size $\ll n(r+1)$.)

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.) The only restriction we make on $\mathcal{F}_1$'s linear transformation M is that it is invertible and has all entries $\neq 0$; we observe that this holds for any M with maximal branch-number. We show that any adversary A has small advantage in distinguishing $\mathcal{F}_1$ from a random function F.

Theorem 4. *If A makes at most q total queries to its oracle, then*

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

The bound achieved here is similar to that of Luby and Rackoff [31] in the sense that it is exponentially small in the size of the random function, with a polynomial loss in the number of queries. (The fact that security degrades with the number of rounds, contrary to what one might expect, seems to be an artifact of the proof.) The proof of this theorem is very similar to that of [34, 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 4 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 AES S-box on b bits (recall that it 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 an $m \to 2m$ maximum distance separable (MDS) code; specifically, if $G^T = [I \,|\, A]$, 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 [41] 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 [15] 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 [15] to fields of characteristic 2.

By combining Theorem 2 with a theorem of Nyberg [36], we show that this candidate has exponential security against linear and differential cryptanalysis.

Theorem 5. 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. [26,10]). 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. [22]) 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. [23, Corollary 6.5] and cf. [22]), we obtain a TC^0 circuit.

Because Candidate 3 deviates somewhat from the SPN structure, we cannot use Theorem 1, 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. We were unable to determine if this candidate is 4-wise independent.

Definition 2. *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 6. *$\mathcal{F}_3$ is $(3, 2^{-\Omega(n)})$-wise independent.*

Finally, we mention that implicit in an 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 [39, §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.) The question of whether field inversion is in $\mathrm{AC}^0[\oplus]$ was raised by Healy and Viola in [22]. Their work, and later Kopparty's [29], do show that several functions related to field inversion are not in $\mathrm{AC}^0[\oplus]$.

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 [11], 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 [25], in which we exploit the algebraic structure to recover the key with just 4 queries.

We then put forth a related candidate $\mathcal{F}'_4$ where we only output the Goldreich-Levin bit [18]: $\mathcal{F}'_4(x) := \langle (x + k_0)^{2^b - 2}, k_1 \rangle$. We prove that this candidate is a d-wise small-bias generator with error $d/2^n$ (cf. [33,3]), i.e. that it fools all parity tests that look at $\leq 2^{0.9n}$ outputs.

Theorem 7. *For any choice of $d \leq 2^n$, $\mathcal{F}'_4$ is a d-wise small-bias generator with error $d/2^n$. That is, 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 [7] (cf. [5,40]) 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 [12].

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. [10]), 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.

Due to space constraints, the technical details of most of our candidates and full proofs of all theorems are deferred to the full version of this paper. However, in the following subsection we explain Candidate 1's proof of security.

2.1 Security of Candidate 1

Recall that $\mathcal{F}_1$ is an SPN in which the S-box $S : \mathrm{GF}(2^b) \to \mathrm{GF}(2^b)$ is chosen uniformly at random and the linear transformation M is invertible and has only non-zero entries. To tie the latter restriction to practical constructions, we observe that any M with maximal branch number suffices for this construction.

Claim. Let $M \in (\mathrm{GF}(2^b))^{m \times m}$ 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 $\boldsymbol{k} := (k_0, \ldots, k_{r-1}) \in (\{0,1\}^n)^r$, let $\mathcal{F}_1 = \mathcal{F}_1(S, \boldsymbol{k})$ 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).

We make the standard assumption that the adversary A is deterministic, computationally unbounded, and never queries an oracle twice with the same input.

Proof Overview. The proof of Theorem 4 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 $(S, \boldsymbol{k})$ such that, conditioned on $\neg$BAD, $\{\mathcal{F}_1(x_i)\}_{i \leq q}$ is uniformly distributed. Essentially, BAD occurs when any of the x_i induce colliding queries to some pair of S-boxes in the final round. When considering distinct instances of the S-box in the final round, even under the same query to $\mathcal{F}_1$, this event has low probability simply due to the fact that each b-bit block of k_{r-1} is uniform and independent. However when considering the *same* final round S-box (necessarily under two distinct queries to $\mathcal{F}_1$), bounding the collision probability is more involved and relies on the properties of M stated above.

In the second stage of the proof, we consider the distribution over transcripts of A's interaction with its oracle. We show that the probability mass assigned to transcripts for which A outputs 1 differs by at most $\max_{\{x_1,\ldots,x_q\}}(\Pr[\text{BAD}])$ (which is negligible by the first stage). To show this we employ a probability argument that has been used in a number of other works, e.g. [34,38,14].

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.[3] A different method (that does not give a useful bound in our setting)

[3] This can be seen for example by considering 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 is almost pairwise-independent, but trivially distinguishable with two adaptive queries.

for obtaining adaptive security from non-adaptive security is given by Hoory et al. [24, Prop. 3].

Stage 1. Fix distinct $x_1, \ldots, x_q \in \{0,1\}^n$. We view $(S, \boldsymbol{k})$ being chosen as follows:

1. Uniformly choose $k_0, \ldots, k_{r-3}$.
2. Run the computation of rounds $1, \ldots, r-2$ of $\mathcal{F}_1(x_i)$ for all $i \leq q$. 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 $qm(r-2)$ S-inputs whose output is determined after this step.
3. Uniformly choose k_{r-2}.
4. Uniformly choose the output of S for each round-$(r-1)$ S-box whose output is not already determined.
5. Uniformly choose k_{r-1}.
6. Uniformly choose the output of S on all remaining (round r) S-box inputs.

It is clear that, for any $x_1, \ldots, x_q$, this distribution is uniform. Our analysis uses the state of the SPN's computation immediately before the final two rounds, and we denote these states for query x_i as $z_i^{(r-1)}$ and $z_i^{(r)}$.

We now define the event BAD. To reduce notation, we use the following definition.

Definition 3. *Let $x, y \in (GF(2^b))^m$, 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 $T \subseteq GF(2^b)$, we say that x and T* collide *if $\exists \ell \leq m, t \in T : \; x^{(\ell)} = t$. 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 $(S, \boldsymbol{k})$ such that at least one of the following holds:

(a) $\exists h, h' \in H : S(h) = S(h')$.
(b) $\exists i < q : z_i^{(r)}$ and H collide.
(c) $\exists i, i' \leq q : z_i^{(r)}$ and $(z_i^{(r-1)} + k_{r-2})$ collide.
(d) $\exists i \leq q : z_i^{(r)}$ self-collides.
(e) $\exists i \neq i' \leq q : z_i^{(r)}$ and $z_{i'}^{(r)}$ collide.

It is crucial for us that determining whether BAD holds can be checked after step 5 in choosing $(S, \boldsymbol{k})$. The following two lemmas show that BAD occurs with low probability, and that the query answers are uniformly distributed when conditioned on $\neg$BAD.

Lemma 1. $\Pr_{S,\boldsymbol{k}}[\text{BAD}] < O(r^2 m^3 q^3) \cdot 2^{-b}$.

Lemma 2. *For any distinct $x_1, \ldots, x_q$ and any $y_1, \ldots, y_q$:*

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

Stage 2. The proof of Theorem 4 concludes by using the two preceding lemmas in a probability argument similar to [34, Thm. 3.2]. To do this, we consider 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 $i \neq j$.

Proof (of Theorem 4). Let Γ be the set of possible transcripts such that $A(\sigma) = 1 \Leftrightarrow \sigma \in \Gamma$. Then,

$$\begin{aligned}
&\left| \Pr_F\left[A^F = 1\right] - \Pr_{S,\boldsymbol{k}}\left[A^{\mathcal{F}_1} = 1\right] \right| \\
= &\left| \sum_{\sigma \in \Gamma} \left(\Pr_F[T_F = \sigma] - \Pr_{S,\boldsymbol{k}}[T_{\mathcal{F}_1} = \sigma] \right) \right| \\
\leq &\left| \sum_{\sigma \in \Gamma} \Pr_{S,\boldsymbol{k}}[\text{BAD}] \cdot \left(\Pr_F[T_F = \sigma] - \Pr_{S,\boldsymbol{k}}[T_{\mathcal{F}_1} = \sigma \mid \text{BAD}] \right) \right| \qquad (1) \\
&+ \left| \sum_{\sigma \in \Gamma} \Pr_{S,\boldsymbol{k}}[\neg\text{BAD}] \cdot \left(\Pr_F[T_F = \sigma] - \Pr_{S,\boldsymbol{k}}[T_{\mathcal{F}_1} = \sigma \mid \neg\text{BAD}] \right) \right|. \qquad (2)
\end{aligned}$$

Lemma 2 implies that (2) = 0, because $\Pr_F[T_F = \sigma] = 2^{-qmb}$ for any possible transcript σ. We rewrite (1) as

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

Each of the two summations is bounded by $\alpha := \max_{\sigma \in \Gamma} \left(\Pr_{S,\boldsymbol{k}}[\text{BAD}] \right)$, 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 $\alpha < O(r^2 m^3 q^3) \cdot 2^{-b}$ by Lemma 1. □

3 Conclusion and Future Work

Two obvious directions for future work are to extend the analysis of $\mathcal{F}_1$ to handle inverse queries (necessarily choosing the S-box as a random *permutation*), and to extend Theorem 6 to prove almost d-wise independence of $\mathcal{F}_3$ for $d > 3$. A more foundational question left unanswered is to understand how the degree of each output bit of an SPN (as a polynomial in the input bits) is affected by the degree of the S-box and by the "mixing" properties of the linear transformation.

Exploring other choices of the S-box besides inversion may lead to more efficient constructions, and utilizing other properties of the linear transformation besides maximal-branch-number may allow stronger proofs of security. This could potentially give a (plausibly secure) SPN computable by circuits of size $O(n)$. Recall from §1 that a PRF computable with size $O(n)$ and with security 2^n would bring the Natural Proofs barrier to the current frontier of lower bounds against unbounded-depth circuits.

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 $\bmod 3$ 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.

We believe a good candidate PRF should be the simplest candidate that resists known attacks. As noted in [10], 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 or when deploying a system that is to be widely used, one can argue that a better way for the research community to proceed is to put forth the simplest candidate PRF, possibly break it, and iterate until hopefully converging to a secure PRF. We view this paper as a step in this direction.

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 [26].

References

1. Aaronson, S., Wigderson, A.: Algebrization: a new barrier in complexity theory. In: 40th ACM Symp. on the Theory of Computing, STOC, pp. 731–740 (2008)
2. Allender, E., Koucký, M.: Amplifying lower bounds by means of self-reducibility. J. of the ACM 57(3) (2010)
3. Alon, N., Goldreich, O., Håstad, J., Peralta, R.: Simple constructions of almost k-wise independent random variables. Random Structures & Algorithms 3(3), 289–304 (1992)
4. Baker, T., Gill, J., Solovay, R.: Relativizations of the *P=?NP* question. SIAM J. Comput. 4(4), 431–442 (1975)
5. Bazzi, L.M.J.: Polylogarithmic independence can fool DNF formulas. SIAM J. Comput. 38(6), 2220–2272 (2009)
6. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology 4(1), 3–72 (1991)

7. Braverman, M.: Poly-logarithmic independence fools AC^0 circuits. In: 24th IEEE Conf. on Computational Complexity, CCC. IEEE (2009)
8. Brodsky, A., Hoory, S.: Simple permutations mix even better. Random Struct. Algorithms 32(3), 274–289 (2008)
9. Cho, H.-S., Sung, S.H., Kwon, D., Lee, J.-K., Song, J.H., Lim, J.: New Method for Bounding the Maximum Differential Probability for SPNs and ARIA. In: Park, C., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 21–32. Springer, Heidelberg (2005)
10. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)
11. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. J. Cryptology 10(3), 151–162 (1997)
12. Gao, S., von zur Gathen, J., Panario, D., Shoup, V.: Algorithms for exponentiation in finite fields. J. Symb. Comput. 29(6), 879–889 (2000)
13. Gauravaram, P., Knudsen, L.R., Matusiewicz, K., Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: Grøstl: a SHA-3 candidate (2011), http://www.groestl.info
14. Gentry, C., Ramzan, Z.: Eliminating Random Permutation Oracles in the Even-Mansour Cipher. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 32–47. Springer, Heidelberg (2004)
15. Gerasoulis, A.: A fast algorithm for the multiplication of generalized Hilbert matrices with vectors. Mathematics of Computation 50, 179–188 (1988)
16. Goldreich, O.: Foundations of Cryptography: Volume 1, Basic Tools. Cambridge University Press (2001)
17. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. of the ACM 33(4), 792–807 (1986)
18. Goldreich, O., Levin, L.: A hard-core predicate for all one-way functions. In: 21st ACM Symp. on the Theory of Computing, STOC, pp. 25–32 (1989)
19. Gowers, W.: An almost m-wise independent random permutation of the cube. Combinatorics, Probability and Computing 5(2), 119–130 (1996)
20. Haitner, I., Reingold, O., Vadhan, S.P.: Efficiency improvements in constructing pseudorandom generators from one-way functions. In: 42nd ACM Symp. on the Theory of Computing, STOC, pp. 437–446 (2010)
21. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. 28(4), 1364–1396 (1999)
22. Healy, A., Viola, E.: Constant-Depth Circuits for Arithmetic in Finite Fields of Characteristic Two. In: Durand, B., Thomas, W. (eds.) STACS 2006. LNCS, vol. 3884, pp. 672–683. Springer, Heidelberg (2006)
23. Hesse, W., Allender, E., Barrington, D.A.M.: 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)
24. Hoory, S., Magen, A., Myers, S., Rackoff, C.: Simple permutations mix well. Theor. Comput. Sci. 348(2-3), 251–261 (2005)
25. Jakobsen, T., Knudsen, L.: Attacks on block ciphers of low algebraic degree. Journal of Cryptology 14, 197–210 (2001)
26. Kang, J.S., Hong, S., Lee, S., Yi, O., Park, C., Lim, J.: Practical and provable security against differential and linear cryptanalysis for substitution-permutation networks. ETRI Journal 23(4), 158–167 (2001)
27. Keliher, L., Meijer, H., Tavares, S.: New Method for Upper Bounding the Maximum Average Linear Hull Probability for SPNs. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 420–436. Springer, Heidelberg (2001)

28. Knudsen, L.R.: Truncated and Higher Order Differentials. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 196–211. Springer, Heidelberg (1995)
29. Kopparty, S.: On the complexity of powering in finite fields. In: ACM Symp. on the Theory of Computing, STOC (2011)
30. Kushilevitz, E., Nisan, N.: Communication complexity. Cambridge University Press (1997)
31. Luby, M., Rackoff, C.: How to construct pseudorandom permutations from pseudorandom functions. SIAM J. Comput. 17(2), 373–386 (1988)
32. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
33. Naor, J., Naor, M.: Small-bias probability spaces: efficient constructions and applications. SIAM J. Comput. 22(4), 838–856 (1993)
34. Naor, M., Reingold, O.: On the construction of pseudorandom permutations: Luby-Rackoff revisited. J. Cryptology 12(1), 29–66 (1999)
35. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. J. of the ACM 51(2), 231–262 (2004)
36. Nyberg, K.: Differentially Uniform Mappings for Cryptography. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 55–64. Springer, Heidelberg (1994)
37. Pieprzyk, J.: On bent permutations. In: Proceedings of the International Conference on Finite Fields, Coding Theory, and Advances in Communications and Computing, Las Vegas (August 1991)
38. Ramzan, Z., Reyzin, L.: On the Round Security of Symmetric-Key Cryptographic Primitives. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 376–393. Springer, Heidelberg (2000)
39. Razborov, A., Rudich, S.: Natural proofs. J. of Computer and System Sciences 55(1), 24–35 (1997)
40. Razborov, A.A.: A simple proof of Bazzi's theorem. ACM Transactions on Computation Theory (TOCT) 1(1) (2009)
41. Roth, R.M., Seroussi, G.: On generator matrices of MDS codes. IEEE Transactions on Information Theory 31, 826–830 (1985)
42. Shannon, C.: Communication theory of secrecy systems. Bell Systems Technical Journal 28(4), 656–715 (1949)
43. Vadhan, S.P., Zheng, C.J.: Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In: ACM Symp. on the Theory of Computing, STOC (2012)
44. Williams, R.: Non-uniform ACC lower bounds. In: IEEE Conf. on Computational Complexity, CCC (2011)
45. Wu, H.: The hash function JH (2011), `http://www3.ntu.edu.sg/home/wuhj/research/jh/index.html`