Electronic Colloquium on Computational Complexity, Report No. 113 (2017)

Pseudorandom Functions: Three Decades Later*

Andrej Bogdanov† Alon Rosen‡

Abstract

In 1984, Goldreich, Goldwasser and Micali formalized the concept of pseudorandom functions and proposed a construction based on any length-doubling pseudorandom generator. Since then, pseudorandom functions have turned out to be an extremely influential abstraction, with applications ranging from message authentication to barriers in proving computational complexity lower bounds.

In this tutorial we survey various incarnations of pseudorandom functions, giving self-contained proofs of key results from the literature. Our main focus is on feasibility results and constructions, as well as on limitations of (and induced by) pseudorandom functions. Along the way we point out some open questions that we believe to be within reach of current techniques.

I have set up on a Manchester computer a small programme using only 1000 units of storage, whereby the machine supplied with one sixteen figure number replies with another within two seconds. I would defy anyone to learn from these replies sufficient about the programme to be able to predict any replies to untried values.

A. TURING (from [GGM84])

*This survey appeared in the book *Tutorials on the Foundations of Cryptography*, published in honor of Oded Goldreichs 60th birthday.

†Dept. of Computer Science and Engineering and Institute of Theoretical Computer Science and Communications, Chinese University of Hong Kong. andrejb@cse.cuhk.edu.hk

‡Efi Arazi School of Computer Science, IDC Herzliya. alon.rosen@idc.ac.il

ISSN 1433-8092

1 Introduction

A family of functions $F_s\colon \{0,1\}^k \to \{0,1\}^\ell$, indexed by a key $s \in \{0,1\}^n$, is said to be *pseudorandom* if it satisfies the following two properties:

Easy to evaluate: The value $F_s(x)$ is efficiently computable given s and x.

Pseudorandom: The function F_s cannot be efficiently distinguished from a uniformly random function $R\colon \{0,1\}^k \to \{0,1\}^\ell$, given access to pairs $(x_i, F_s(x_i))$, where the x_i's can be adaptively chosen by the distinguisher.

One should think of the key s as being kept secret, and of the running time of evaluation as being substantially smaller than that of the distinguisher. This faithfully models a prototypical attack on a cryptographic scheme: the adversary's running time is bounded but can still exceed that of the system, and he may adaptively adjust his probing of the system's input/output behavior.

The definition of pseudorandom functions (PRFs), along with the demonstration of its feasibility, is one of the keystone achievements of modern cryptography [66]. This owes much to the fact that the definition hits a "sweet spot" in terms of level of abstraction: it is simple enough to be studied and realized, and at the same time is powerful enough to open the door to countless applications.

Notably, PRFs lend themselves to simple proofs of security. Being indistinguishable from a random function means that analysis cleanly reduces to an idealized system in which a truly random function is used instead of the pseudorandom one.

1.1 Applications

Perhaps the most natural application of pseudorandom functions is that of *message authentication*. The goal is to allow Bob to verify that a message m was sent to him by Alice and nobody else. To this end, Alice and Bob share a randomly sampled secret key s, known only to them. When Alice wishes to authenticate m, she appends a tag σ that is efficiently computable from m and s. Verifiability of (m,σ) follows from the fact that Bob also knows m and s and so can compute σ efficiently.

An authentication scheme is said to be *unforgeable* if no computationally bounded adversary (not possessing s) can generate a pair (m,σ) that passes verification, where m can be any message that was not previously sent (and hence authenticated) by Alice. To authenticate m using a PRF family F_s, Alice simply sends to Bob the (message, tag) pair

$$(m, F_s(m)). \tag{1}$$

Upon receiving (m,σ), Bob uses s to evaluate $F_s(m)$ and verifies that it equals σ. Unforgeability follows from the fact that the probability with which any computationally bounded adversary correctly guesses $\sigma = F_s(m)$ does not noticeably change if $F_s(m)$ is replaced with $R(m)$, where R is a random function. The probability of correctly guessing $R(m)$ is $2^{-\ell}$. This remains true even if the adversary gets to see pairs of the form (m_i,σ_i), where $m \neq m_i$ for all i and the m_i's are adaptively chosen.

Symmetric-key encryption. In the setting of *symmetric-key encryption*, Alice and Bob share a randomly sampled secret key s, which is used along with some other string r to generate an encryption $\mathrm{Enc}_s(m;r)$ of a plaintext m. Alice sends $\mathrm{Enc}_s(m;r)$ to Bob, who can use s in order to compute the decryption $m = \mathrm{Dec}_s(\mathrm{Enc}_s(m;r))$.

An encryption scheme is said to be *secure* if for any two plaintexts m_0, m_1 the distributions $\mathrm{Enc}_s(m_0;r)$ and $\mathrm{Enc}_s(m_1;r)$ cannot be efficiently distinguished. Given a PRF family F_s, one can implement a secure encryption scheme as follows:

$$\mathrm{Enc}_s(m;r) = (r, F_s(r) \oplus m), \qquad \mathrm{Dec}_s(r,c) = F_s(r) \oplus c. \tag{2}$$

Similarly to the case of message authentication, security is established by observing that the advantage of any efficient distinguisher between $\mathrm{Enc}_s(m_0;r)$ and $\mathrm{Enc}_s(m_1;r)$ will not noticeably change if we replace $F_s(r)$ with $R(r)$, where R is a random function. In the latter case, the adversary's task is to distinguish between $R(r) \oplus m_0$ and $R(r) \oplus m_1$, which is information-theoretically impossible.

This argument is valid even if the distinguisher gets to see $\mathrm{Enc}_s(m_i;r_i)$ for adaptively chosen m_i's (m_0, m_1 are also allowed), provided that $r_i \neq r$ for all i. In practice, this can be enforced by either deterministically choosing the r's using a counter, or by sampling them independently at random each time. The counter solution does not require including r as part of the encryption, but requires maintaining state between encryptions. The randomized solution does not require state, but has longer ciphertexts and moreover requires r to be long enough so that collisions of the form $r_i = r$ are unlikely.

Interestingly, neither of the above solutions necessitates the full strength of PRFs, in the sense that they do not require security against *adaptive* access to the function. In the counter solution, the PRF adversary only observes the function values on a predetermined set of inputs, whereas in the randomized mode, it observes values on randomly chosen inputs. This motivates two interesting relaxations of PRFs, called *nonadaptive* PRFs and *weak* PRFs, respectively, and opens the door to more efficient constructions.

Key derivation. The following is a convenient method for generating a long sequence of cryptographic keys "on-the-fly". Let F_s be a PRF, and define a key k_i by

$$k_i = F_s(i). \tag{3}$$

This method has advantages both in terms of memory usage and in terms of key management, at least as long as one is able to protect the (relatively short) "master-key" s from leaking to an attacker (by definition, F_s remains pseudorandom even if some of the k_i's are leaked). In terms of security, any efficient system that uses the k_i's as secret keys is guaranteed to be no less secure than the same system would have been if it were to use truly random and independent keys.

Storing, protecting, and managing a single short key s is indeed convenient. However, it has the disadvantage that compromise of s results in loss of security for the entire system. One way to mitigate this concern would be to store $\mathrm{FHE}(s)$, where FHE is a *fully homomorphic* encryption scheme [128]. The idea would be to store $c = \mathrm{FHE}(s)$ and erase s, keeping only the FHE decryption key. One can then homomorphically compute $\mathrm{FHE}(F_s(i))$ for any desired i (think of the circuit $C_i(s) = F_s(i)$), and then decrypt the result at the location in which the FHE decryption key is stored (say on a client machine). An attacker who compromises the system learns nothing about the master key s, whereas an attacker who compromises the client alone learns only the FHE key.

While conceptually simple, this solution is still not practical. Currently known FHE schemes can practically support only simple computations, certainly not ones nearly as complex as evaluating a PRF. Can PRFs be made simple enough to allow their fast homomorphic evaluation? Alternatively, could one devise FHE schemes so that efficient homomorphic evaluation of compatible PRFs is enabled?

Hardness of learning. The fundamental task in machine learning is to make future predictions of an unknown concept based on past training data. In the model of *probably approximately correct* (PAC) learning [135, 84] the concept is described by an efficient function F, the data comes in the form of input–output samples $(x, F(x))$, and the objective is for the learner to make almost always correct predictions (with respect to a given distribution on inputs). Statistical considerations show that $O(\log|\mathbf{C}|)$ random samples provide sufficient information to learn any function coming from a given class $\mathbf{C}$ with precision at least 99%.

In particular, a PRF F_s can in principle be learned from $O(n)$ random samples, where n is the size of its key. The learning, however, cannot be carried out efficiently: any learner L that is able to predict the value of F_s at a new input x^* based on past data $(x_1, F_s(x_1)), \ldots, (x_q, F_s(x_q))$ can be applied to distinguish the sequences

$$\begin{aligned}&(x_1, F_s(x_1)), \ldots, (x_q, F_s(x_q)), (x^*, F_s(x^*)) \quad \text{and}\\ &(x_1, R(x_1)), \ldots, (x_q, R(x_q)), (x^*, R(x^*)),\end{aligned}$$

thereby violating the pseudorandomness of F_s. To distinguish, one can use the first q elements as training data for the learner and test the value $F(x^*)$ against the learner's prediction $L(x^*)$. If the learner is probably approximately correct, $L(x^*)$ is likely to agree with $F(x^*)$ when $F = F_s$. On the other hand, when $F = R$, the value $F(x^*)$ is statistically independent of the training data and uncorrelated with $L(x^*)$.

A learning algorithm can thus be viewed as a potential cryptanalytic attack against any PRF. Vice versa, any algorithm that is believed to learn a given concept class should be tested on conjectured PRF constructions that fall within this class.

1.2 Feasibility

The construction of PRFs necessitates postulating computational hardness. This is not surprising given that the existence of PRFs requires at the very least ruling out the possibility that P equals NP: distinguishing F_s from a random function R reduces to the NP-search problem of finding a key s consistent with the samples $(F(1), F(2), \ldots, F(m))$. Such a key always exists when $F = F_s$, but only with small probability when $F = R$ (assuming $m\ell > n$).

The first hardness assumption under which PRFs were constructed is the existence of *pseudorandom generators* [66]. A pseudorandom generator (PRG) is an efficiently computable deterministic function $G\colon \{0,1\}^n \to \{0,1\}^m$ with $m > n$ whose output $G(s) \in \{0,1\}^m$, where s is sampled uniformly from $\{0,1\}^n$, cannot be efficiently distinguished from a uniformly random string $r \in \{0,1\}^m$.

While PRGs and PRFs both map a short random string into a longer pseudorandom string, the definitions differ in the quantity of output bits and in the adversary's access to them. Any PRF family $F_s\colon \{0,1\}^k \to \{0,1\}$ gives rise to a PRG:

$$G(s) = (F_s(1), F_s(2), \ldots, F_s(m)),$$

as long as $n < 2^k$ (assuming, for simplicity, that the distribution on keys s is uniform). In other words, the truth-table of a PRF is an efficiently computable sequence of pseudorandom bits of essentially unbounded length. In contrast, the output length of a PRG is a priori bounded by its running time.

From this perspective, a PRF can be thought of as a PRG whose output length is much larger than the running time of the distinguisher. As this output is too large to be stored in the distinguisher's memory, a definitional choice must be made regarding how these bits are accessed by the adversary. In this respect, the definition of a PRF provides the adversary with imposing power: his access to the pseudorandom bits is *adversarial* and *adaptive*.

Goldreich, Goldwasser, and Micali showed how to use any length-doubling PRG, $G\colon \{0,1\}^n \to \{0,1\}^{2n}$, to construct a PRF family $F_s\colon \{0,1\}^k \to \{0,1\}^\ell$, that is keyed by $s \in \{0,1\}^n$, for arbitrary k and ℓ. Subsequently, it was shown that PRGs are polynomial-time equivalent to one-way functions [75, 71, 134]. A function $f\colon \{0,1\}^n \to \{0,1\}^\ell$ is *one-way* if f is efficiently computable, but given $y = f(x)$ for a random x, it is infeasible to find any x' such that $f(x') = y$.

Theorem 1 ([66, 75])**.** *Pseudorandom functions exist iff one-way functions exist.*

One-way functions are the most rudimentary primitive in modern cryptography. Their existence is necessary for virtually all applications of interest, save a select few in which information-theoretic security is achievable. The definition of a one-way function merely postulates the ability to hide a "secret" that is computationally hard to reconstruct. This encompasses, in particular, the secret key s of any candidate PRF construction. In contrast, the security requirements of a PRF are significantly more stringent: the adversary is given access to multiple input–output samples of its choice and is only asked to detect any form of nonrandom behavior, a seemingly much easier task than reconstructing the secret key s.

Owing to the relatively mild security requirement of one-way functions, candidate constructions abound. Any stochastic computational process that is typically difficult to invert can be modeled as a one-way function. In contrast, pseudorandom functions must exhibit significant internal structure in order to resist the vast variety of distinguishers that they can be tested against (see Section 7 for some representative examples). It is thus remarkable that the two notions are equivalent.

1.3 Efficiency, Security, and Functionality

The work of Goldreich, Goldwasser, and Micali (GGM) provides an elegant conceptual solution to the problem of constructing PRFs. This has opened the door towards the finer study of their theory and practice, and has resulted in a large body of work. In this survey we will focus on the following aspects:

Efficiency. Every evaluation of the GGM PRF on a k-bit input necessitates k sequential invocations to the underlying PRG, while its provable security deteriorates as k becomes larger. Are there more efficient constructions?

Naor and Reingold gave a construction that has lower parallel evaluation complexity than the GGM construction, but assumes the availability of a pseudorandom synthesizer, an object (seemingly) stronger than a PRG. In Section 3 we present the two constructions of PRFs, and in Section 4 we give concrete instantiations of PRFs obtained using this paradigm.

On the negative side, the existence of efficient learning algorithms for certain types of circuits implies inherent lower bounds on the complexity of pseudorandom functions. Razborov and Rudich explain how such learning algorithms arise naturally from proofs of circuit lower bounds. We discuss these connections in Section 6.

Security. Pseudorandom functions are required to be secure against all efficient distinguishers. It is sometimes useful to consider security against restricted classes of distinguishers that model specific types of attacks such as differential cryptanalysis. A sufficiently restrictive class of adversaries may allow for a proof of security that is unconditional. Proofs of security against restricted distinguishers can also provide confidence in the soundness of heuristic constructions.

In Section 7 we discuss some restricted classes of distinguishers arising from the study of pseudorandomness (bounded query distinguishers, linear distinguishers, space-bounded algorithms), complexity theory (polynomials, rational functions), and learning theory (statistical queries).

Functionality. In Section 5 we illustrate the robustness of the definition of PRFs with respect to domain size and discuss how PRFs provide a basis for implementing "huge random objects", the most notable example of which are pseudorandom permutations.

For certain cryptographic applications it is useful to have pseudorandom functions with additional functionality. In Section 8 we present two such extensions: key-homomorphic PRFs and puncturable PRFs.

Open questions. In spite of the enormous body of work on pseudorandom functions in the last three decades, many questions of interest remain unanswered. We mark some of our favorite ones with the symbol ⑦ as they come up in the text. For convenience, all the open questions are indexed at the end of the chapter.

1.4 The Wide Scope of Pseudorandom Functions

Pseudorandom functions permeate cryptography and are of fundamental importance in computational complexity and learning theory. In this survey we do not attempt to provide comprehensive coverage of their applications, but focus instead on a handful of representative settings which highlight their conceptual importance. The following (partial) list gives an indication of the wide scope of PRFs.

Basic cryptographic applications. Pseudorandom functions fit naturally into message authentication and in particular underlie the security of the widely deployed authentication function HMAC [22, 21]. They are also used in constructions of deterministic stateless digital signatures [61], and randomized stateless symmetric-key encryption (see Section 1.1).

Pseudorandom permutations (PRPs, see Section 5.2), which are closely related to PRFs, model block ciphers such as DES and AES, where the PRP security notion was a criterion in the design [117].

Advanced cryptographic applications. PRFs have been applied to achieve resettable security in protocols [50, 18], to hide memory access patterns in oblivious RAM [62, 68], and to bootstrap fully homomorphic and functional encryption schemes [9, 8]. The construction of authentication schemes from PRFs extends naturally to provide digital signatures from *verifiable PRFs* [23, 100].

Key-homomorphic PRFs are useful for constructing distributed PRFs, proxy re-encryption, and other applications with high relevance to "cloud" security (see Section 8.2). The recently introduced notion of puncturable PRFs, in conjunction with indistinguishability obfuscation, has found applications for the construction of strong cryptographic primitives, and demonstrates how to bridge between private-key and public-key encryption (see Section 8.2).

Puncturable PRFs have also been recently combined with indistinguishability obfuscation to exhibit hard on the average instances for the complexity classses PPAD [33, 60] and CLS [76].

Other applications. In the realm of data structures, permutation-based hashing, which is inspired by the Feistel construction of PRPs, has been applied to improve the performance of dynamic dictionaries [12]. PRPs were also recently used in the construction of adaptively secure Bloom filters [113]. More generally, PRFs are a basic building block in implementations of huge random objects (see Section 5.3).

Lower bounds and barriers. As pointed out in Section 1.1, PRF constructions present a fundamental barrier for efficient learning algorithms (see Section 6.1) and for our ability to prove circuit lower bounds (see Section 6.2).

Finally, pseudorandom functions provide natural examples for "pseudo-entropic" functions that cannot be virtually black-box obfuscated in a strong sense [69, 32].

1.5 Intellectual Merit

The evolution in the design and use of PRFs exemplifies how theory affects practice in indirect ways, and how basic conceptualizations free our minds to develop far-reaching and unexpected ideas. The wide array of applications of PRFs can in large part be attributed to their simplicity and flexibility. These traits facilitate the robust design of cryptographic primitives, while relying on clearly stated and well-defined assumptions (compare this with vague terms such as "diffusion" and "confusion").

For instance, while a candidate PRP was already proposed in the mid 1970s (DES), there was no methodology available at the time for capturing the desired security properties. Indeed, rigorous analysis of various modes of operation for block ciphers [24, 87] and Feistel-like constructions [93, 109] only emerged after the 1984 work of Goldreich, Goldwasser, and Micali [66].

From a pedagogical point of view, the study of pseudorandom functions clarifies concepts and sharpens distinctions between notions that arise in the study of cryptographic constructions. Some examples that come to mind are:

Computational indistinguishability. PRFs are a prime example of a distribution that is extremely non-random from a statistical perspective, yet indistinguishable from random by computationally bounded observers (see discussion on "A delicate balance" in Section 2.1). Moreover, computational indistinguishability in PRF constructions and applications exemplifies the use of the hybrid proof technique, which is prevalent in cryptographic reasoning.

Key recovery, prediction, and distingushing. For an adversary to break a cryptographic system it does not necessarily have to fully recover the key, which may be underspecified by the system's behavior. A more reasonable notion of security is that of unpredictability of future responses based on past interaction. In the case of PRFs, this type of attack is exemplified by PAC learning algorithms, which reconstruct an approximation of the function based on past input–output data. As explained in Section 6.1, unpredictability is implied by indistinguishability from a random function. The converse does not hold in general. The ability to distinguish a system's behavior from random already opens the door to severe security breaches, even if the function cannot be predicted or the key fully recovered.

Modeling access to a system. The definition of PRFs cleanly captures what type of access an adversary can have to a system, be it adaptive, nonadaptive, sequential, or statistically random (see Section 2.1). It also clarifies the distinction between the adversary's running time and the number of times it queries the function/system.

How not to model a random function. For the definition of PRFs to make sense, the function's description given by the random key s must be kept secret from the distinguisher. This should be contrasted to the random oracle model [56, 25], whose instantiations (wrongly) assume that the oracle retains its random properties even if its description is fully available to the distinguisher.

2 Definitions

If you have something interesting to say,
it doesn't matter how you say it.
LEONID LEVIN (1985)

We give a formal definition of pseudorandom functions, discuss our definitional choices, and try to provide some intuition behind the requirements. We then gain some practice through two "warm-ups": first, we show how to generically increase the range size of a PRF. Second, we give a security analysis of the symmetric-key encryption scheme from Section 1.1. Although these types of proofs are standard in cryptography [63, 82], they will serve as compelling motivations to introduce some additional ideas and should help the reader get accustomed to the notation.

For formal definitions of circuits and oracles the reader is referred to Goldreich's textbook on computational complexity [64].

2.1 Pseudorandom Functions

Since it does not make sense to require pseudorandomness from a single fixed function $F : \{0,1\}^k \to \{0,1\}^\ell$, the definition of pseudorandom functions refers to distributions of functions sampled from a family. Each member of the family is a function $F : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^\ell$, where the first argument is called the *key* and denoted s, and the second argument is called the *input* to the function. The key is sampled according to some distribution S and then fixed. We are interested in the pseudorandomness of the function $F_s(x) = F(s,x)$ over the distribution S of the keys.

The PRF's adversary is modeled by a Boolean circuit D that is given oracle access to some function F. Namely, throughout its computation, D has access to outputs of the function F on inputs $x_1, x_2, \ldots$ of his choice. The type of access can vary depending on the definition of security. By default D is given *adaptive* access, meaning that the input x_j may depend on values $F(x_i)$ for $i < j$. For an oracle F we denote by D^F the output of D when given access to F.

Definition 1 (Pseudorandom function [66])**.** Let S be a distribution over $\{0,1\}^n$ and $\{F_s \colon \{0,1\}^k \to \{0,1\}^\ell\}$ be a family of functions indexed by strings s in the support of S. We say $\{F_s\}$ is a (t,ε)-*pseudorandom function family* if for every Boolean-valued oracle circuit D of size at most t,

$$\left|\Pr_s[D^{F_s} \text{ accepts}] - \Pr_R[D^R \text{ accepts}]\right| \leq \varepsilon,$$

where s is distributed according to S, and R is a function sampled uniformly at random from the set of all functions from $\{0,1\}^k$ to $\{0,1\}^\ell$.

The string s is called the *key*, the circuit D is called the *distinguisher*, and the difference in probabilities is its *distinguishing advantage* with respect to F_s. Since D is nonuniform, we may assume that it is deterministic, as it can always hardwire the coin tosses that maximize its distinguishing advantage.

Weak, nonadaptive, and sequential PRFs. In certain settings it is natural to restrict the oracle access mode of the distinguisher. In a *nonadaptive* PRF the distinguisher must make all its queries to the oracle at once. In a *weak* PRF at every invocation the oracle returns the pair $(x, F(x))$ for a uniformly random x in $\{0,1\}^k$. In a *sequential* PRF the i-th oracle invocation is answered by the value $F(i)$.

Efficiency. We say $\{F_s\}$ has *size* c if s can be sampled by a circuit of size at most c and for every s there exists a circuit of size at most c that computes F_s. We are interested in the parameter regime where the size of $\{F_s\}$ is much smaller than the distinguisher size t and the inverse of its distinguishing advantage $1/\varepsilon$. In the theory of cryptography it is customary to view n as a symbolic *security parameter* and study the asymptotic behavior of other parameters for a function ensemble indexed by an infinite sequence of values for n. The PRF is viewed as efficient if its input size k grows polynomially in n, but its size is bounded by some polynomial of n.

Security. Regarding security, it is less clear what a typical choice of values for t and ε should be. At one end, cryptographic dogma postulates that the adversary be given at least as much computational power as honest parties. In the asymptotic setting, this leads to the minimalist requirement of *superpolynomial security*: for every t that grows polynomially in n, ε should be negligible in n (it should eventually be smaller than $1/p(n)$ for every polynomial p). At the other end, it follows from statistical considerations that $(2^n, 1/2)$ and $(\omega(n), 2^{-n})$-PRFs cannot exist. In an attempt to approach these limitations as closely as possible, the maximalist notion of *exponential* security sets t and ε to $2^{\alpha n}$ and $2^{-\beta n}$, respectively, for some constants $\alpha, \beta \in (0,1)$. Most cryptographic reductions, including all the ones presented here, guarantee deterioration in security parameters that is at most polynomial. Both super-polynomial and exponential security are preserved under such reductions.

A delicate balance. PRFs strike a delicate balance between efficiency and security. A random function $R\colon \{0,1\}^k \to \{0,1\}^\ell$ is ruled out by the efficiency requirement: its description size, let alone implementation size, is as large as $\ell \cdot 2^k$. For the description size to be polynomial in k, the PRF must be sampled from a set of size at most $2^n = 2^{\mathrm{poly}(k)}$, which is a tiny fraction of the total number of functions $2^{\ell 2^k}$. Let $\mathbf{F}$ be such a set and assume for simplicity that F_s is sampled uniformly from $\mathbf{F}$. Which sets $\mathbf{F}$ would give rise to a PRF? One natural possibility is to choose $\mathbf{F}$ at random, uniformly among all sets of size 2^n. Then with overwhelming probability over the choice of $\mathbf{F}$ the function F_s is indistinguishable from random, but the probability that it can be computed efficiently (in terms of circuit size) is negligible.

Uniformity. We model computational efficiency using circuit size. A more common alternative is the running time of some uniform computational model such as Turing machines. Most of the theory covered here carries over to the uniform setting: all reductions between implementations preserve uniformity, but some of the reductions between adversaries may rely on nonuniform choices. We think that the circuit model of computation is a more natural one in the context of PRFs. Besides, proofs of security in the circuit model are notationally simpler.

2.2 Computational Indistinguishability

It will be convenient to define a general notion of *computational indistinguishability*, which considers distinguishers D that adaptively interact as part of probabilistic experiments called *games*. When a game is not interactive we call it a *distribution*.

Definition 2 (Computational indistinguishability)**.** We say that games H_0, H_1 are (t,ε)-*indistinguishable* if for every oracle circuit D of size at most t,

$$\left|\Pr[D^{H_0} \text{ accepts}] - \Pr[D^{H_1} \text{ accepts}]\right| < \varepsilon,$$

where the probabilities are taken over the coin tosses of H_0, H_1.

The definition of pseudorandom functions can be restated by requiring that the following two games are (t,ε)-computationally indistinguishable:

F_s: Sample $s \in \{0,1\}^n$ from S and give D adaptive oracle access to F_s

R: Sample $R\colon \{0,1\}^k \to \{0,1\}^\ell$ and give D adaptive oracle access to R

Definition 2 is more general in that it accomodates the specification of games other than those occurring in the definition of a PRF.

Proposition 1. *Suppose that games H_0, H_1 are (t_1,ε_1)-indistinguishable and that games H_1, H_2 are (t_2,ε_2)-indistinguishable. Then, H_0, H_2 are $(\min\{t_1,t_2\}, \varepsilon_1+\varepsilon_2)$-indistinguishable.*

In other words, computational indistinguishability is a transitive relation (up to appropriate loss in parameters). Proposition 1 is proved via a direct application of the triangle inequality to Definition 2.

Proposition 2. *Let H^q denote q independently sampled copies of a distribution H. If H_0 and H_1 are (t,ε)-indistinguishable then H_0^q and H_1^q are $(t, q\varepsilon)$-indistinguishable.*

Proof. For $i \in \{0,\ldots,q\}$, consider the "hybrid" distribution $D_i = (H_0^i, H_1^{q-i})$. We claim that D_i and D_{i+1} are (t,ε)-indistinguishable. Otherwise, there exists a circuit B of size t that distinguishes between D_i, D_{i+1} with advantage ε. We use B to build a B' of size t that distinguishes between H_0, H_1 with the same advantage.

The circuit B' is given an h that is sampled from either H_0 or H_1. It then samples h_0^{i-1} from H_0^{i-1} and h_1^{q-i} from H_1^{q-i} (the samples can be hardwired inducing no overhead in size), feeds the vector $d = (h_0^{i-1}, h, h_1^{q-i})$ to B, and outputs whatever B outputs. If h is sampled from H_0, then d is distributed according to D_i, whereas if it is sampled from H_1, then d is distributed according to D_{i-1}. We get that B' distinguishes between H_0, H_1 with advantage ε, in contradiction to their (t,ε)-indistinguishability. Thus, D_i and D_{i+1} are (t,ε)-indistinguishable. The claim now follows by observing that $D_0 = H_1^q$ and $D_q = H_0^q$ and by invoking Proposition 1 for q times. □

Two games are (∞,ε)-indistinguishable if they are indistinguishable by any oracle circuit, regardless of its size. In the special case of distributions, (∞,ε)-indistinguishability is equivalent to having statistical (i.e., total variation) distance at most ε.

2.3 Warm-Up I: Range Extension

Sometimes it is desirable to increase the size of the range of a PRF. This may for instance be beneficial in applications such as message authentication where one requires a function whose output on a yet unobserved input to be unpredictable. The larger the range, the harder it is to predict the output.

We show that a pseudorandom function with many bits of output can be obtained from a pseudorandom function with one bit of output. Let $F'_s\colon \{0,1\}^k \to \{0,1\}$ be any pseudorandom function family and define $F_s\colon \{0,1\}^{k-\lceil \log \ell \rceil} \to \{0,1\}^\ell$ by $F_s(x) = (F'_s(x,1), F'_s(x,2), \ldots, F'_s(x,\ell))$, where the integers $1,\ldots,\ell$ are identified with their $\lceil \log \ell \rceil$-bit binary expansion.

Proposition 3. *If $\{F'_s\}$ is a (t,ε)-pseudorandom function family then $\{F_s\}$ is a $(t/\ell,\varepsilon)$-pseudorandom function family.*

Proof. For every oracle circuit D whose oracle type is a function from $\{0,1\}^{k-\lceil \log \ell \rceil}$ to $\{0,1\}^{\ell}$, let D' be the circuit that emulates D as follows: when D queries its oracle at x, D' answers it by querying its own oracle at $(x,1),\ldots,(x,\ell)$ and concatenating the answers. The distributions D^{F_s} and $D'^{F'_s}$ are then identical, and so are D^R and $D'^{R'}$ for random functions R and R'.

It follows that D and D' have the same distinguishing advantage. By construction, D' is at most ℓ times larger than D. Therefore, if D is a circuit of size t/ℓ with distinguishing advantage ε, D' has size t with distinguishing advantage ε. By assumption such a D' does not exist so neither does such a D. □

Proposition 3 provides a generic secure transformation from a PRF with one bit of output to a PRF with ℓ bits of output for any given value of ℓ. This generality, however, comes at the price of worse complexity and security: implementation size grows by a factor of ℓ, while security drops by the same factor. Such losses are often unavoidable for a construction obtained by means of a generic transformation, and it is indeed desirable to directly devise efficient constructions.

2.4 Warm-Up II: Symmetric-Key Encryption

We now state and prove the security of the encryption scheme $(\mathrm{Enc},\mathrm{Dec})$ described in (2). Recall that $\mathrm{Enc}_s(m;r) = (r, F_s(r) \oplus m)$.

Proposition 4. *If $\{F_s : \{0,1\}^k \to \{0,1\}^\ell\}$ is a weak $(t+\ell t, \varepsilon)$-pseudorandom function family then for every two messages $m_0, m_1 \in \{0,1\}^\ell$, the following games are $(t, 2\varepsilon + t/2^k)$-indistinguishable:*

E_0: Sample random $s \in \{0,1\}^n$ and $r \in \{0,1\}^k$ and output $\mathrm{Enc}_s(m_0;r)$

E_1: Sample random $s \in \{0,1\}^n$ and $r \in \{0,1\}^k$ and output $\mathrm{Enc}_s(m_1;r)$

In both games the distinguisher is also given access to an oracle that in the i-th invocation samples a uniform and independent r_i and outputs $\mathrm{Enc}_s(x_i;r_i)$ on input x_i.

Proof. Consider the following two games:

R_0: Sample random $R\colon \{0,1\}^k \to \{0,1\}^\ell$ and $r \in \{0,1\}^k$ and output $(r, R(r) \oplus m_0)$

R_1: Sample random $R\colon \{0,1\}^k \to \{0,1\}^\ell$ and $r \in \{0,1\}^k$ and output $(r, R(r) \oplus m_1)$

In both games the distinguisher is also given access to an oracle that on input x_i samples a uniform and independent r_i and outputs $(r_i, R(r_i) \oplus x_i)$.

Claim 1. *For $b \in \{0,1\}$ games E_b and R_b are (t,ε)-indistinguishable.*

Proof. Suppose for contradiction that there exists a distinguisher A of size t that distinguishes between E_b and R_b with advantage ε. We use A to build a distinguisher D between F_s and R with the same advantage.

The distinguisher D is given access to an oracle F that is either F_s or R. It emulates A, answering his queries, which are either according to E_b or to R_b, as follows:

- Sample r, query F to obtain $F(r)$, and output $\mathrm{Enc}_s(m_b;r) = (r, F(r) \oplus m_b)$
- On input x_i, sample r_i, query F to obtain $F(r_i)$, and output $(r_i, F(r_i) \oplus x_i)$

Accounting for the ℓ extra $\oplus$ gates incurred by each oracle query (out of at most t queries) of A, the circuit D is of size $t + \ell t$. Note that D^{F_s} and D^R are identically distributed to A^{E_b} and A^{R_b}, respectively, so D distinguishes F_s from R with advantage ε, contradicting the $(t+\ell t, \varepsilon)$-pseudorandomness of F_s. □

Claim 2. *Games R_0 and R_1 are $(t, t/2^k)$-indistinguishable.*

Proof. Let A be a potential distinguisher between R_0 and R_1. Note that A's view of the games R_0 and R_1 is identical conditioned on the event that A never makes a query x that is answered by $(r, R(r) \oplus x)$. Since an A of size t makes at most t queries and r_i is chosen uniformly and independently for every query the probability of this event is at most $t/2^k$. □

Combining the two claims with Proposition 1, we can conclude that E_0 and E_1 are $(t, 2\varepsilon + t/2^k)$-indistinguishable. □

The analysis incurs security loss that grows linearly with the number of encryption queries made by D. In this case the number of queries was bounded by t, which is the size of D. However, as we will see later, it is sometimes useful to separately quantify the number of queries made by the distinguisher.

Definition 3 (Bounded-query PRF)**.** A (t, q, ε)-*pseudorandom function* is a (t, ε)-pseudorandom function in which the distinguisher makes at most q queries.

We also give an analogous definition for computational indistinguishability.

Definition 4 (Bounded-query indistinguishability)**.** Games H_0 and H_1 are (t, q, ε)-*indistinguishable* if they are (t, ε)-indisinguishable by distinguishers that make at most q queries.

Decoupling the number of queries from the adversary's running time (as well as from the function's input size) will turn out to be beneficial in the proofs of security of the GGM and NR constructions (Section 3), in the construction of pseudorandom permutations (Section 5.2), and in the discussion of natural proofs (Section 6.2).

3 Generic Constructions

Beware of proofs by induction,
especially in crypto.
ODED GOLDREICH (1990s)

We now present two generic methods for constructing a pseudorandom function. The first method, due to Goldreich, Goldwasser, and Micali (GGM), relies on any length-doubling *pseudorandom generator*. The second method is due to Naor and Reingold (NR). It builds on a stronger primitive called a *pseudorandom synthesizer*.

Both the GGM and NR methods inductively extend the domain size of a PRF. Whereas the GGM method doubles the domain size with each inductive step, the NR method squares it. Instantiations of the NR method typically give PRFs of lower depth. The GGM construction, on the other hand, has shorter keys and relies on a simpler building block.

3.1 The Goldreich–Goldwasser–Micali Construction

We start by defining the notion of a pseudorandom generator (PRG). Pseudorandom generation is a relatively well-understood cryptographic task. In particular, it admits many candidate instantiations along with highly efficient implementations.

Definition 5 (Pseudorandom generator [38, 137])**.** Let $G\colon \{0,1\}^n \to \{0,1\}^m$ be a deterministic function, where $m > n$. We say that G is a (t,ε)-*pseudorandom generator* if the following two distributions are (t,ε)-indistinguishable:

- Sample a random "seed" $s \in \{0,1\}^n$ and output $G(s)$.
- Sample a random string $r \in \{0,1\}^m$ and output r.

A PRG $G\colon \{0,1\}^n \to \{0,1\}^{2n}$ can be viewed as a pseudorandom function $F'_s\colon \{0,1\} \to \{0,1\}^n$ over one input bit. For this special case the pair of values $(F'_s(0), F'_s(1))$ should be indistinguishable from a truly random pair. This is satisfied if we set $F'_s(0) = G_0(s)$ and $F'_s(1) = G_1(s)$, where $G_0(s)$ and $G_1(s)$ are the first n bits and the last n bits of the output of G, respectively.

This extends naturally for larger domains. Suppose for example that we wish to construct a two-bit input PRF $F_s\colon \{0,1\}^2 \to \{0,1\}^n$, which is specified by its four values $(F_s(00), F_s(01), F_s(10), F_s(11))$. To this end one can define the values of F_s by an inductive application of the pseudorandom generator

$$G_0(F'_s(0)) \quad G_1(F'_s(0)) \quad G_0(F'_s(1)) \quad G_1(F'_s(1)). \tag{4}$$

Since F'_s is pseudorandom we can replace it with a random $R'\colon \{0,1\} \to \{0,1\}^n$, and infer that distribution (4) is computationally indistinguishable from the distribution

$$G_0(R'(0)) \quad G_1(R'(0)) \quad G_0(R'(1)) \quad G_1(R'(1)). \tag{5}$$

On the other hand the distribution (5) can be described as the pair of values obtained by applying G on the independent random seeds $R'(0)$ and $R'(1)$. The pair $(G(R'(0)), G(R'(1))$ is computationally indistinguishable from a pair of uniformly random values. Therefore, distribution (4) is computationally indistinguishable from the truth-table of a random function $R\colon \{0,1\}^2 \to \{0,1\}^n$. The GGM construction generalizes this idea naturally to larger input lengths. It is described in Figure 1.

Building block: A length-doubling pseudorandom generator, $G : \{0,1\}^n \to \{0,1\}^{2n}$

Function key: A seed $s \in \{0,1\}^n$ for G

Function evaluation: On input $x \in \{0,1\}^k$ define $F_s\colon \{0,1\}^k \to \{0,1\}^n$ as

$$F_s(x_1 \cdots x_k) = G_{x_k}(G_{x_{k-1}}(\cdots G_{x_1}(s)\cdots)),$$

where $G(s) = (G_0(s), G_1(s)) \in \{0,1\}^n \times \{0,1\}^n$.

Size: $k \cdot \mathrm{size}(G)$

Depth: $k \cdot \mathrm{depth}(G)$

Figure 1: The Goldreich–Goldwasser–Micali construction

The construction can be thought of as a labeling of the leaves of a binary tree of depth k, where the leaf indexed by $x \in \{0,1\}^k$ is labeled by the value $F_s(x)$. The value at each leaf is evaluated in the time it takes to reach the leaf but is never stored. Figure 2 illustrates the case $k = 3$.

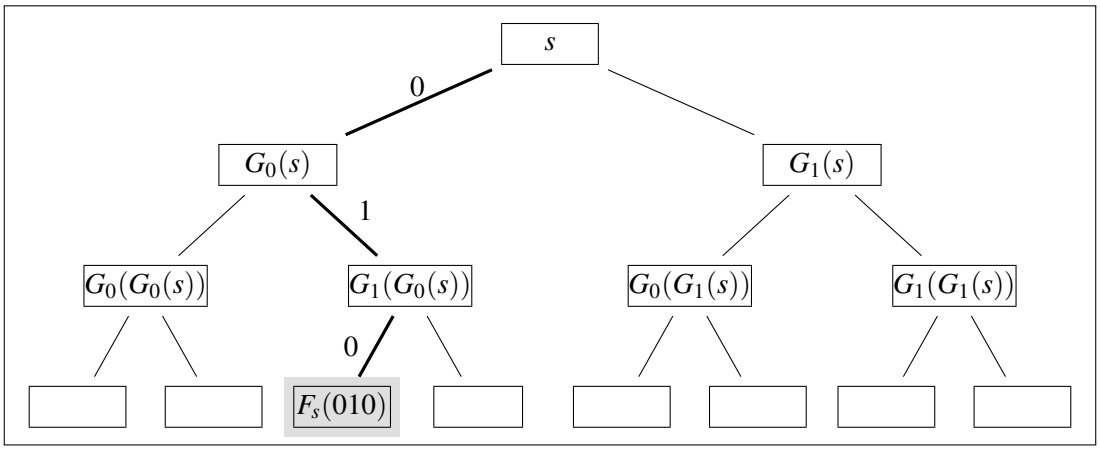

Figure 2: Evaluating $F_s(010) = G_0(G_1(G_0(s)))$

Theorem 2 ([66])**.** *If $G\colon \{0,1\}^n \to \{0,1\}^{2n}$ is a (t,ε)-pseudorandom generator then $\{F_s\}$ is a $(t', kt'\varepsilon)$-pseudorandom function family, as long as $t' = o(\sqrt{t/k})$ and the size of G is at most t'.*

Proof. We prove the theorem by induction on k. Given $F'_s\colon \{0,1\}^{k-1} \to \{0,1\}^n$ define $F_s\colon \{0,1\}^k \to \{0,1\}^n$ as $F_s(x,y) = G_x(F'_s(y))$. For the inductive step we will apply the following claim. Let c denote the circuit size of G.

Claim 3. *If F'_s is a $(t - O(kq^2) + cq, q, \varepsilon')$-PRF then F_s is a $(t - O(kq^2), q, \varepsilon' + q\varepsilon)$-PRF.*

Recall that a (t,q,ε)-PRF is a (t,ε)-PRF in which the distinguisher makes at most q queries to the function (Definition 3).

Proof. Let $(x,y) \in \{0,1\} \times \{0,1\}^{k-1}$ be an oracle query made by a purported distinguisher D between F_s and a random R. Consider the following three games:

F_s: Sample random $s \in \{0,1\}^n$. Answer with $F_s(x,y) = G_x(F'_s(y))$.

H: Sample random $R'\colon \{0,1\}^{k-1} \to \{0,1\}^n$. Answer with $H(x,y) = G_x(R'(y))$.

R: Sample random $R\colon \{0,1\}^k \to \{0,1\}^n$. Answer with $R(x,y)$.

Claim 4. *Games F_s and H are $(t - O(kq^2), \varepsilon')$-indistinguishable.*

Proof. If D distinguishes F_s from H with advantage ε' then consider the circuit $A^{F'}$ that simulates D by answering D's queries (x,y) with $G_x(F'(y))$. Then $A^{F'_s}$ and $A^{R'}$ are identically distributed to D^{F_s} and D^H, respectively, so A is a circuit of size at most $t - O(kq^2) + cq$ and query complexity q that distinguishes F'_s from R' with advantage ε'. This contradicts the $(t - O(kq^2) + cq, q, \varepsilon')$-pseudorandomness of F'_s. □

Claim 5. *Games H and R are $(t - O(kq^2), q\varepsilon)$-indistinguishable.*

Proof. Suppose that D distinguishes H from R with advantage $q\varepsilon$. We use D to build a circuit A of size t that distinguishes between q pseudorandom strings $(G_0(s_i), G_1(s_i))$ and q random strings $r_i \in \{0,1\}^{2n}$ with advantage $q\varepsilon$. By Proposition 2 this is in contradiction with the assumed (t,ε)-pseudorandomness of G.

The distinguisher A obtains q strings $z_i = (z_{0i}, z_{1i}) \in \{0,1\}^{2n}$ as input. It answers D's query $(x,y) \in \{0,1\} \times \{0,1\}^{k-1}$ with z_{xi}, where the index $i = i(y)$ is chosen to ensure consistency among D's queries. For example $i(y)$ can be set to the smallest previously unused index when y is queried for the first time. This tracking of queries can be implemented with $O(kq^2)$ additional gates. Then the random variables $A(G(s_1), \ldots, G(s_q))$ and $A(r_1, \ldots, r_q)$ are identically distributed as D^H and D^R, respectively, so A has the same distinguishing advantage as D. □

Combining the two claims with Proposition 1, we get that F_s is a $(t - O(kq^2), q, \varepsilon' + q\varepsilon)$-PRF. □

We now prove that for every $q \leq t$, F_s is a $(t - O(kq^2) - c(k-1)q, q, ((k-1)q+1)\varepsilon)$-PRF by induction on k. In the base case $k = 1$, $F_s(x) = G_x(s)$ and F_s is (t,q,ε)-secure by the assumed security of G. The inductive step is immediate from the claim we just proved.

If we set $q = \alpha\sqrt{t/k}$ for a sufficiently small absolute constant $\alpha > 0$, assume that $c \leq q$, and simplify the expression, we get that F_s is a $(t/2, q, kq\varepsilon)$-PRF. Because $q \leq t/2$, F_s is a $(q, kq\varepsilon)$-PRF. □

The quadratic loss in security can be traced to the distinguisher transformation that enforces the distinctness of its queries. A simple "data structure" for this purpose has size that is quadratic in the number of queries. In other computational models, such as random access machines, more efficient data structures can be used, resulting in better security parameters.

Extensions and properties. The GGM construction readily extends to produce a PRF from $\{1, \ldots, d\}^n$ to $\{1, \ldots, d\}$ from a PRG $G\colon \{0,1\}^n \to \{0,1\}^{dn}$. The output of such a PRG can sometimes be naturally divided up into d blocks of n bits. This variant is particularly attractive when random access to the blocks is available (see Section 4.2 for an example). The security of the GGM PRF extends to quantum adversaries, where the distinguisher may query the function in superposition [139].

The GGM construction is not *correlation intractable* [65]: it is possible to efficiently find an $x \in \{0,1\}^k$ that maps to say 0^ℓ given the key s for a suitable instantiation of the PRG G. At the same time, the GGM construction is *weakly one-way* for certain parameter settings [53]: for a nonnegligible fraction of the inputs x it is infeasible to recover x given s and $F_s(x)$.

3.2 The Naor–Reingold Construction

Using pseudorandom synthesizers as building blocks, Naor and Reingold give a generic construction of a PRF [111]. Synthesizers are not as well understood as PRGs, and in particular do not have as many candidate instantiations. Most known instantiations rely on assumptions of a "public-key" flavor. Towards the end of this section we show how weak PRFs give rise to pseudorandom synthesizers, opening the door for basing synthesizers on "private-key" flavored assumptions.

Definition 6 (Pseudorandom synthesizer [111])**.** Let $S\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a deterministic function. We say that S is a (t,q,ε)-*pseudorandom synthesizer* if the following two distributions are (t,ε)-indistinguishable:

- Sample $a_1, \ldots, a_q, b_1, \ldots, b_q \leftarrow \{0,1\}^n$. Output the q^2 values $S(a_i, b_j)$.
- Output q^2 independent uniform random strings in $\{0,1\}^n$.

A synthesizer can be seen as an almost length-squaring pseudorandom generator with good locality properties, in that it maps $2q$ random "seed" elements to q^2 pseudorandom elements, and any component of its output depends on only two components of the input seed.

Using a recursive tree-like construction, it is possible to obtain PRFs on k-bit inputs, which can be computed using a total of about k synthesizer evaluations, arranged in $\log k$ levels. Given a synthesizer S and two independent PRF instances F_0 and F_1 on t input bits each, one gets a PRF on $2t$ input bits, defined as

$$F(x_1 \cdots x_{2t}) = S\big(F_0(x_1 \cdots x_t), F_1(x_{t+1} \cdots x_{2t})\big). \tag{6}$$

The base case of a 1-bit PRF can trivially be implemented by returning one of two random strings in the function's secret key.

Building block: A synthesizer $S : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$

Function key: A collection of $2k$ strings in $\{0,1\}^n$, where k is a power of two

Function evaluation: On input $x \in \{0,1\}^k$ recursively define $F_s \colon \{0,1\}^k \to \{0,1\}^n$ as

$$F_s(x) = \begin{cases} S(F_{s_0}(x_0), F_{s_1}(x_1)), & \text{if } k > 1, \\ s_x, & \text{if } k = 1, \end{cases}$$

where z_0 and z_1 denote the left and right halves of the string z.

Size: $k \cdot \text{size}(S)$

Depth: $\log k \cdot \text{depth}(S)$

Figure 3: The Naor–Reingold construction

The evaluation of F_s can be thought of as a recursive labeling process of a binary tree with k leaves and depth $\log k$. The i-th leaf has two possible labels: $s_{i,0}$ and$s_{i,1}$. The i-th input bit x_i selects one of these labels s_{i,x_i}. The label of each internal node at depth d is the value of S on the labels of its children, and the value of F_s is simply the label of the root.

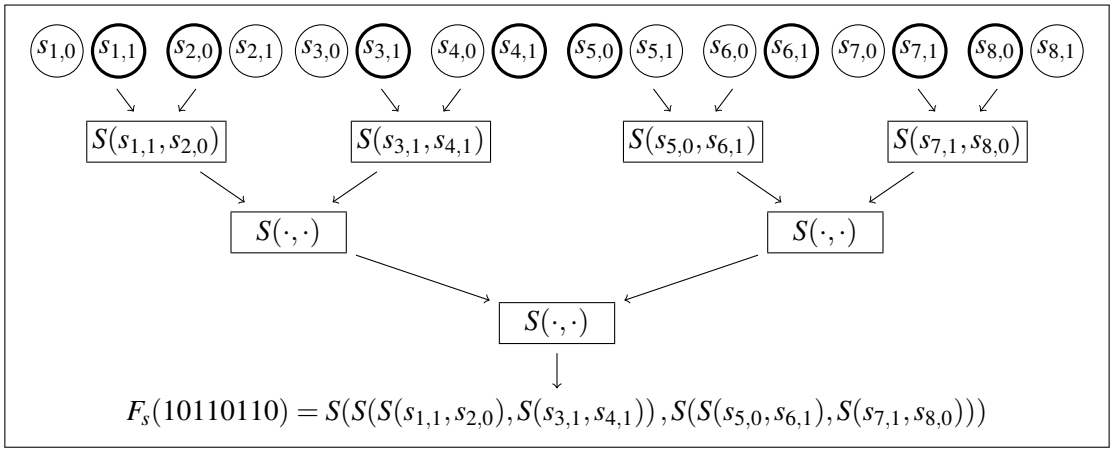

Figure 4: Evaluating $F_s(x)$ at $x = 10110110$

This labeling process is very different than the one associated with the GGM construction. First, the binary tree is of depth $\log k$ instead of depth k as in GGM. Second, the labeling process starts from the leaves instead of from the root. Moreover, here each input defines a different labeling of the tree, whereas in GGM the labeling of the tree is fully determined by the key.

Theorem 3 ([111]). *If the function $S\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a (t,q,ε)-pseudorandom synthesizer then $\{F_s\}$ is a $(q,(k-1)\varepsilon)$-pseudorandom function family, as long as the size of S is at most q and that $q = o(\sqrt{t/\max\{k,n\}\log k})$.*

Proof. We prove the theorem by induction on k. Given $F'_s\colon \{0,1\}^k \to \{0,1\}^n$ define $F_s\colon \{0,1\}^{2k} \to \{0,1\}^n$ by $F_{s_0,s_1}(x_0,x_1) = S(F'_{s_0}(x_0), F'_{s_1}(x_1))$. Let c be the size of S and $k^\star = \max\{k,n\}$.

Claim 6. *If F'_s is a $(t+O(k^\star q^2 + ckq), q, \varepsilon')$-PRF then F_s is a $(t,q,2\varepsilon'+\varepsilon)$-PRF.*

Proof. Let $(x_0,x_1) \in \{0,1\}^k \times \{0,1\}^k$ be an oracle query made by a purported distinguisher D between F_s and a random R. Consider the following four games:

F_s: Sample random $s_0, s_1 \leftarrow \{0,1\}^{2k\times n}$. Answer with $S(F'_{s_0}(x_0), F'_{s_1}(x_1))$.

H: Sample random $R_0, R_1\colon \{0,1\}^k \to \{0,1\}^n$. Answer with $S(F'_{s_0}(x_0), R_1(x_1))$.

H': Sample random $R_0, R_1\colon \{0,1\}^k \to \{0,1\}^n$. Answer with $S(R_0(x_0), R_1(x_1))$.

R: Sample a random $R\colon \{0,1\}^{2k} \to \{0,1\}^n$. Answer with $R(x_0,x_1)$.

Claim 7. *Games F_s and H are (t,q,ε')-indistinguishable.*

Proof. If this were not the case, namely there is a distinguisher D with the corresponding parameters, then F'_{s_1} and R_1 would be distinguishable by a circuit A^F that simulates D and answers each query x by $S(F'_{s_0}(x_0), F(x))$. The key s_0 can be hardwired to maximize the distinguishing advantage between F'_{s_1} and R. The additional complexity of A is $(c+O(1))k$ gates per query, as the evaluation of F'_{s_0} requires $k-1$ evaluations of S. This contradicts the assumed security of F'_{s_1}.

□

Claim 8. *Games H and H' are (t,q,ε')-indistinguishable.*

Proof. The proof is analogous to the previous claim, except that now A^F answers each query x with $S(F(x_0), R_1(x_1))$. The distinguisher emulates the function $R_1(x_1)$ by answering every new query with a fresh random string (eventually hardwired to maximize the distinguishing advantage). This requires tracking previous queries, which can be accomplished with $O(kq^2)$ additional gates. Another cq gates are sufficient for evaluating the synthesizer. The resulting circuit A has size $t + O(kq^2 + cq)$ and distinguishes F'_{s_0} from R_0 with q queries and advantage ε', violating the assumed security of F'_{s_0}. □

Claim 9. *Games H' and R are (t,q,ε)-indistinguishable.*

Proof. Suppose that D distinguishes between H' and R in size t and q queries with advantage ε. We will argue that D can be used to break the security of the synthesizer. The challenge of the synthesizer consists of a collection of q^2 strings $u_{ij}, 1 \le i \le j \le q$ coming from one of the two distributions in Definition 6.

We describe the circuit A that distinguishes these two distributions. The circuit A simulates D, answering query (x_0,x_1) with u_{ij} where the indices $i = i(x_0)$ and $j = j(x_1)$ are chosen in some manner consistent with

past queries. For example, $i(x_0)$ can set to the smallest previously unused index when x_0 is queried for the first time, and similarly for $j(x_1)$. This tracking of queries can be implemented with $O(k^\star q^2)$ additional gates. Then A perfectly simulates the games H' and R under the two distributions u_{ij} from Definition 6, respectively, thereby distinguishing them with advantage ε in size $t + O(k^\star q^2)$. □

This completes the proof of Theorem 3. □

We now prove by induction on k that F_s is a $(t - O(k^* q^2 + ckq)\log k, q, (k-1)\varepsilon)$-PRF. In the base case $k = 1$, F'_s is perfectly secure, so it is a $(t, q, 0)$-PRF for all t and q. The inductive step from length k to length $2k$ follows from the above claim.

Setting $q = \alpha\sqrt{t/k^\star \log k}$ for a sufficiently small absolute constant $\alpha > 0$ and assuming that $c \leq q$, after simplifying we obtain that F_s is a $(t/2, q, (k-1)\varepsilon)$-PRF. Since $q \leq t/2$, F_s is in particular a $(q, (k-1)\varepsilon)$-PRF. □

The NR function can be shown to admit natural time/space tradeoffs, as well as techniques for compressing the key size. These ideas are described in detail in [111]. As in the case of GGM, the NR construction is also secure against quantum distinguishers [139]. We next show that any weak PRF gives rise to a synthesizer.

Proposition 5. *If $W_s\colon \{0,1\}^n \to \{0,1\}^n$ is a (t,ε)-weak PRF for uniformly distributed keys $s \in \{0,1\}^n$ then the function $S(s,x) = W_s(x)$ is a $(t - cq^2, q, \varepsilon + \binom{q}{2} \cdot 2^{-n})$-pseudorandom synthesizer for every q, where c is the circuit size of W_s.*

Proposition 5 assumes that the weak PRF is length preserving. This assumption is essentially without loss of generality (see Section 5.1), though for efficiency it may be desirable to guarantee this property directly by construction.

Proof. The q queries provided to the synthesizer's adversary can be represented as a $q \times q$ matrix. Assume for contradiction that there is a circuit D' of size $t - cq^2$ that distinguishes between the following two distributions with advantage $q\varepsilon$:

S: Sample random $s_i, x_j \in \{0,1\}^n$. Output the $q \times q$ matrix $S(s_i, x_j) = W_{s_i}(x_j)$.

R: Sample q^2 random entries $r_{ij} \in \{0,1\}^n$. Output the $q \times q$ matrix r_{ij}.

Let S' be the distribution S with the additional condition that the strings x_j are pairwise distinct. Distributions S and S' are $(\infty, q, \binom{q}{2}2^{-n})$-indistinguishable.

Claim 10. *Distributions S' and R are $(t - cq^2, q, \varepsilon)$-indistinguishable.*

Proof. Let H_i be the hybrid in which the first $q - i$ rows are sampled from distribution S' and the rest are sampled from distribution R. Then D' distinguishes $H_{i^\star - 1}$ and $H_{i^\star}$ with advantage ε for some $i^\star$. This holds even after a suitable fixing of the values s_i for all $i < i^\star$ and r_{ij} for all $i > i^\star$ and j. Consider now the following distinguisher D: First, obtain samples $(x_1, y_1), \ldots, (x_q, y_q)$ from the oracle. Then generate the matrix M whose (i,j)-th entry is $W_{s_i}(x_j)$ for $i < i^*$, y_j for $i = i^*$ and r_{ij} for $i > i^*$ and simulate D' on input M. Then D has size at most t and distinguishes W_s from a random function with advantage at least ε. □

The proposition now follows from the triangle inequality (Proposition 1). □

Despite their close relation, in Section 7.6 we present evidence that pseudorandom synthesizers are objects of higher complexity than weak PRFs.

4 Instantiations

Why don't they do all the riots
at the same time?
CHARLIE RACKOFF (1980s)

The building blocks underlying the GGM and NR constructions can be instantiated with specific number-theoretic and lattice-based computational hardness assumptions, resulting in efficient constructions of pseudorandom functions. The first class of instantiations is based on the *decisional Diffie–Hellman* (DDH) problem. The second class is based on the *learning with errors* (LWE) problem, via a deterministic variant of LWE called *learning with rounding* (LWR).

Utilizing the structure of the constructions, it is possible to optimize efficiency and obtain PRFs that are computable by constant-depth polynomial-size circuits with unbounded fan-in threshold gates (TC^0 circuits). Beyond giving rise to efficient PRFs that are based on clean and relatively well-established assumptions, these constructions have direct bearing on our ability to develop efficient learning algorithms and prove explicit lower bounds for the corresponding circuit classes (see Section 6).

The algebraic structure underlying the PRF instantiations also opens the door to more advanced applications such as verifiability, key homomorphism, and fast homomorphic evaluation. Some of these are described in Section 8.

4.1 Number-Theoretic Constructions

We consider the availability of public parameters $(\mathbb{G}, g)$, where g is a randomly chosen generator of a group $\mathbb{G}$ of prime order q with $|q| = n$. For concreteness think of $\mathbb{G}$ as being a subgroup of $\mathbb{Z}_p^*$ where p is a prime such that q divides $p-1$.

The DDH problem. *We say that the DDH problem is* (t,ε)-hard *in* $(\mathbb{G}, g)$ *if the following two games are* (t,ε)*-indistinguishable:*

- *Sample random and independent* $a, b \in \mathbb{Z}_q$ *and output* $(g^a, g^b, g^{ab}) \in \mathbb{G}^3$.
- *Sample random and independent* $a, b, c \in \mathbb{Z}_q$ *and output* $(g^a, g^b, g^c) \in \mathbb{G}^3$.

For (t,ε)-DDH hardness to hold it is necessary that the discrete logarithm problem is (t,ε)-hard in the group $\mathbb{G}$; namely no circuit of size less than t can find x given g^x with probability larger than ε. It is not known whether (t,ε)-hardness of the discrete logarithm problem is sufficient for $(\mathrm{poly}(t), \mathrm{poly}(\varepsilon))$-DDH hardness.

The fastest known method for breaking DDH is to find the discrete logarithm of g^a or of g^b.[1] The best classical algorithms for finding discrete logarithms run in time $2^{\tilde{O}(n^{1/3})}$. Thus, given the current state of knowledge, it does not seem unreasonable to assume that there exist $\alpha, \beta > 0$ such that DDH is $(2^{n^\alpha}, 2^{-n^\beta})$-hard.

[1] In particular, since discrete logarithms can be found in time $\mathrm{poly}(n)$ by a quantum algorithm [131], then the DDH problem is not $(\mathrm{poly}(n), 1-\varepsilon)$-hard for quantum algorithms.

Instantiating GGM using a DDH-based PRG. Consider the following family of (effectively) length-doubling functions $G_{g^a} : \mathbb{Z}_q \to \mathbb{G} \times \mathbb{G}$, defined as:

$$G_{g^a}(b) = (g^b, g^{ab}). \tag{7}$$

When $g^a \in \mathbb{G}$ and $b \in \mathbb{Z}_q$ are sampled independently at random, the distribution of $G_{g^a}(b)$ is (t,ε)-indistinguishable from a random pair in $\mathbb{G}^2$ if and only if DDH is (t,ε)-hard. This suggests using the function G_{g^a} as a PRG, and instantiating the GGM construction with it. However, the efficiency of the resulting construction is not very appealing, as it requires k sequential exponentiations modulo p.

To improve efficiency, Naor and Reingold [110] proposed the following construction of a DDH-based length-doubling PRG $G_{g^a} : \mathbb{G} \to \mathbb{G} \times \mathbb{G}$:

$$G_{g^a}(g^b) = (G^0_{g^a}(g^b), G^1_{g^a}(g^b)) = (g^b, g^{ab}). \tag{8}$$

At a first glance this alternative construction does not appear useful, as it is not clear how to compute $G_{g^a}(g^b)$ efficiently. Nevertheless, a closer look at the proof of security of GGM reveals that efficient public evaluation of the underlying PRG is not actually necessary. What would suffice for the construction and proof of GGM to work is that the underlying PRG can be efficiently computed using the random bits a that were used to sample its index g^a.

The key observation is that, if a is known, then $G_{g^a}(g^b)$ can be efficiently evaluated, and thus satisfies the required property. Invoking the GGM construction with the PRG from (8), where at level $i \in [k]$ one uses a generator indexed by g^{a_i} for independently and randomly chosen $a_i \in \mathbb{Z}_q$, one obtains the PRF

$$F_{a_0,\ldots,a_k}(x) = G^{x_k}_{g^{a_k}}(G^{x_{k-1}}_{g^{a_{k-1}}}(\cdots G^{x_1}_{g^{a_1}}(g^{a_0})\cdots)). \tag{9}$$

The final observation leading to an efficient construction of a PRF is that the k sequential exponentiations required for evaluating $F_{\bar{a}}(x)$ can be collapsed into a single subset product $a_1^{x_1} \cdots a_k^{x_k}$, which is then used as the exponent of g^{a_0}.

Public parameters: A group G and a random generator g of $\mathbb{G}$ of prime order q with $|q| = n$

Function key: A vector $\bar{a}$ of $k+1$ random elements $a_0, \ldots, a_k \in \mathbb{Z}_q^*$

Function evaluation: On input $x \in \{0,1\}^k$ define $F_{\bar{a}} : \{0,1\}^k \to \mathbb{G}$ as

$$F_{\bar{a}}(x_1 \cdots x_k) = g^{a_0 \prod_{i=1}^k a_i^{x_i}}.$$

Size: $k \cdot \mathrm{poly}(\log|\mathbb{G}|)$

Depth: $O(1)$ (with threshold gates)

Figure 5: The Naor–Reingold DDH-based construction

Theorem 4 ([110])**.** *If the DDH problem is $(t,1/2)$-hard in $(\mathbb{G},g)$ then $\{F_{\bar{a}}\}$ is a (t',ε')-pseudorandom function family, as long as $t' = o(\varepsilon'^{2/3} t^{1/3}/k)$ and group operations in $\mathbb{G}$ require circuit size at most t'.*

Proof. As seen in (9), the function $F_{\bar{a}}$ is based on the GGM construction. Thus, Theorem 4 follows from Theorem 2. The PRG from (8), which underlies the construction, is (t,ε)-pseudorandom if and only if the DDH problem is (t,ε)-hard, meaning that security depends on the hardness parameters of the DDH problem.

The DDH problem is *random self-reducible*: If it can be solved on a random instance, then it can be solved on any instance. This can be leveraged to reduce the distinguishing advantage at the cost of increasing the complexity of the distinguisher. Let OP be the circuit size of a group operation in $\mathbb{G}$.

Claim 11. *If DDH is $(t,1/2)$-hard then it is $(o(\varepsilon^2 t - 10\cdot \text{OP}),\varepsilon)$-hard for all $\varepsilon > 0$.*

Proof. Consider the randomized mapping $T : \mathbb{G}^3 \to \mathbb{G}^3$ defined as

$$T(g^a, g^b, g^c) = \left((g^a)^r \cdot g^{s_1}, g^b \cdot g^{s_2}, (g^c)^r \cdot (g^a)^{r\cdot s_2} \cdot (g^b)^{s_1} \cdot g^{s_1 \cdot s_2}\right),$$

where s_1, s_2, and r are uniformly and independently sampled in $\mathbb{Z}_q$. It can be verified that T is computable using 10 group operations, given g^a, g^b, g^c, s_1, s_2, and r. Letting $(g^{a'}, g^{b'}, g^{c'}) = T(g^a, g^b, g^c)$ and writing $c = ab + e \mod q$, we have that:

$$a' = ra + s_1 \bmod q, \quad b' = b + s_2 \bmod q, \quad c' = a'b' + er \bmod q.$$

Using the fact that $c = ab \bmod q$ if and only if $e = 0 \bmod q$ and that if $e \neq 0 \bmod q$ then $er \bmod q$ is uniformly distributed in $\mathbb{Z}_q$ (since q is prime), it follows that

- If $c = ab$ then $c' = a'b'$ and a', b' are uniform and independent in $\mathbb{Z}_q$.
- If $c \neq ab$ then a', b', c' are uniform and independent in $\mathbb{Z}_q$.

To obtain the desired parameters, invoke the distinguisher $O(1/\varepsilon^2)$ times independently on the output of the reduction. Accept if the number of times the distinguisher accepts exceeds a threshold that depends on the distinguisher's acceptance probability (this can be determined in advance and hardwired). □

The theorem now follows by plugging the parameters into those of Theorem 2. □

Efficiency. The evaluation of the NR function can be performed by invoking the following two steps in sequence:

1. Compute the "subset product" $a_0 \cdot a_1^{x_1} \cdots a_k^{x_k} \mod q$.
2. Compute the PRF output $F_{\bar{a}}(x) = g^{a_0 \cdot a_1^{x_1} \cdots a_k^{x_k}}$.

As shown in [126], both steps can be computed by constant-depth polynomial-size circuits with threshold gates.[2] Thus, if the DDH problem is indeed hard, PRFs can be computed within the class TC^0, which corresponds to such circuits.

An additional efficiency optimization with interesting practical implications comes from the observation that, for sequential evaluation, efficiency can be substantially improved by ordering the inputs according to a *Gray code*, where adjacent inputs differ in only one position. The technique works by saving the entire state of the subset product $\prod a_i^{x_1}$ from the previous call, updating to the next subset product by multiplying with either a_j or a_j^{-1} depending on whether the j-th bit of the next input is turned on or off (according to the Gray code ordering). This requires only a single multiplication per call (for the subset product part), rather than up to $k-1$ when computing the subset product from scratch.

[2] The second step reduces to subset product by hardwiring $g_i = g^{2^i} \bmod p$ for $i = 1, \ldots, \lceil \log q \rceil$ and observing that $g^x = \prod g_i^{x_i} \bmod p$ for $x = \sum 2^i x_i$.

Applications and extensions. The algebraic structure of the NR pseudorandom functions has found many applications, including PRFs with "oblivious evaluation" [57], verifiable PRFs [94], and zero-knowledge proofs for statements of the form "$y = F_s(x)$" and "$y \neq F_s(x)$" [111]. Naor and Reingold [111], and Naor, Reingold and Rosen [112] give variants of the DDH-based PRF based on the hardness of RSA/factoring. The factoring-based construction directly yields a large number of output bits with constant computational overhead per output bit. Such efficiency cannot be attained generically (i.e., by applying a PRG to the output of a PRF).

4.2 Lattice-Based Constructions

Underlying the efficient construction of lattice-based PRFs are the (decision) learning with errors (LWE) problem, introduced by Regev, and the learning with rounding (LWR) problem, introduced by Banerjee, Peikert, and Rosen (BPR).

The LWE problem ([125]). *We say that the LWE problem is* (t, m, ε)-hard *if the following two distributions are* (t, ε)*-indistinguishable:*

- *Sample random* $\mathbf{s} \in \mathbb{Z}_q^n$ *and output m pairs* $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$*, where* $\mathbf{a}_i$*'s are uniformly random and independent and* $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q$ *for small random* e_i*.*
- *Output m uniformly random and independent pairs* $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$*.*

One should think of the "small" error terms $e_i \in \mathbb{Z}$ as being of magnitude $\approx \alpha q$, and keep in mind that without random independent errors, LWE would be easy. While the dimension n is the main hardness parameter, the error rate α also plays an important role: as long as αq exceeds $\sqrt{n}$ or so, LWE is as hard as approximating conjectured hard problems on lattices to within $\tilde{O}(n/\alpha)$ factors in the worst case [125, 118, 97]. Moreover, known attacks using lattice basis reduction [89, 130] or combinatorial/algebraic methods [37, 13] require time $2^{\tilde{\Omega}(n/\log(1/\alpha))}$. Unlike DDH, no nontrivial quantum algorithms are known for LWE.

The learning with rounding problem is a "derandomized" variant of LWE, where instead of adding a small random error term to $\langle \mathbf{a}_i, \mathbf{s} \rangle \in \mathbb{Z}_q$, one deterministically rounds $\langle \mathbf{a}_i, \mathbf{s} \rangle$ to the nearest element of a public subset of p well-separated values in $\mathbb{Z}_q$, where p is much smaller than q. Since there are only p possible rounded values in $\mathbb{Z}_q$, we view them as elements of $\mathbb{Z}_p$ and denote the rounded value by $\lfloor \langle \mathbf{a}_i, \mathbf{s} \rangle \rceil_p \in \mathbb{Z}_p$, where $\lfloor x \rceil_p$ equals $\lfloor (p/q) \cdot x \bmod q \rceil \bmod p$.

The LWR problem ([17]). *We say that the LWR problem is* (t, m, ε)-hard *if the following two distributions are* (t, ε)*-indistinguishable:*

- *Sample random* $\mathbf{s} \in \mathbb{Z}_q^n$ *and output m pairs* $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$*, where the* $\mathbf{a}_i$*'s are uniformly random and independent and* $b_i = \lfloor \langle \mathbf{a}_i, \mathbf{s} \rangle \rceil_p$*.*
- *Output m uniformly random and independent pairs* $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$*.*

The LWR problem can be hard only if $q > p$, for otherwise no error is introduced. The "absolute" error is roughly q/p, and the "error rate" relative to q (the analogue of the parameter α in the LWE problem) is on the order of $1/p$.

An LWE-error distribution is *B-bounded* if for all errors e in the support of the distribution it holds that $e \in [-B, B]$.[3] Let RD be the cost of rounding a single element in $\mathbb{Z}_q$ into an element in $\mathbb{Z}_p$.

Proposition 6 ([17])**.** *If the LWE problem is (t, m, ε)-hard for some B-bounded error distribution then the LWR problem is $(t - m \cdot \mathrm{RD}, m, mp(2B+1)/q) + \varepsilon)$-hard.*

The proof relies on the fact that when e is small relative to q/p we have $\lfloor \langle \mathbf{a}, \mathbf{s} \rangle + e \rceil_p = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_p$ with high probability (see Figure 6), while $\lfloor x \rceil_p$ for a random $x \in \mathbb{Z}_q$ is random in $\mathbb{Z}_p$ (assuming p divides q). Therefore, given samples $(\mathbf{a}_i, b_i)$ of an unknown type (either LWE or uniform), we can round the b_i terms to generate samples of a corresponding type (LWR or uniform, respectively).

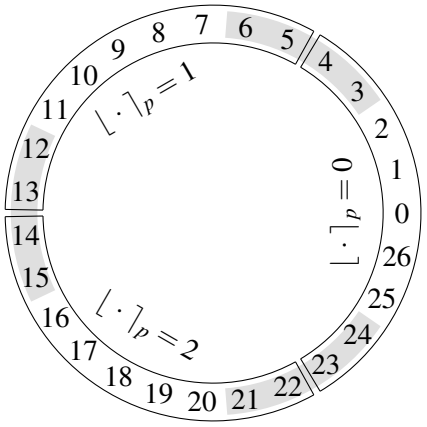

Figure 6: Rounding an LWE sample $\langle a, x \rangle + e$ with $q = 27$, $p = 3$, and $B = 2$. The shaded areas denote the possibility of a rounding error. For instance, when $\langle a, x \rangle = 3$, $\lfloor \langle a, x \rangle \rceil_p = 0$ but it is possible that $\lfloor \langle a, x \rangle + e \rceil_p = 1$, but when $\langle a, x \rangle = 17$, $\lfloor \langle a, x \rangle \rceil_p$ and $\lfloor \langle a, x \rangle + e \rceil_p$ are equal with probability one.

Proof. Consider the following three distributions:

H_0: Output m pairs $(\mathbf{a}_i, \lfloor \langle \mathbf{a}_i, \mathbf{s} \rangle \rceil_p) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$.

H_1: Output m pairs $(\mathbf{a}_i, \lfloor \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \rceil_p) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$.

H_2: Output m pairs $(\mathbf{a}_i, \lfloor b_i \rceil_p) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$, where the b_i's are random.

In all the distributions above, the $\mathbf{a}_i$'s are uniformly random and independent in $\mathbb{Z}_q^n$ and the e_i's are chosen independently from the LWE error distribution. For simplicity, we assume that $\mathbf{s}$ is random in the set of nonzero divisors $\mathbb{Z}_q^{n*} = \{x \in \mathbb{Z}_q^n : \gcd(x_1, ..., x_n, q) = 1\}$.

Claim 12. *Distributions H_0 and H_1 are $(\infty, mp(2B+1)/q)$-indistinguishable.*

Proof. Since the e_i's are selected according to a B-bounded error distribution, it holds that $e_i \in [-B, B]$ for all $i \in [m]$. We thus know that, as long as $\langle \mathbf{a}_i, \mathbf{s} \rangle$ does not fall within $\pm B$ of a multiple of q/p, it is guaranteed that $\lfloor \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \rceil_p = \lfloor \langle \mathbf{a}_i, \mathbf{s} \rangle \rceil_p$.

[3] Under typical LWE error distributions the event $e_i \notin [-B, B]$ does have some positive probability. This probability, however, is usually negligible (think of a Gaussian distribution with $\alpha \approx B/q$), and so one can conduct the analysis conditioning on the event not occurring without substantial loss in parameters.

For any fixed $\mathbf{s} \in \mathbb{Z}_q^{n*}$ the probability over random $\mathbf{a}_i \in \mathbb{Z}_q^n$ that $\langle \mathbf{a}_i, \mathbf{s} \rangle \in \mathbb{Z}_q$ falls within $\pm B$ of a multiple of q/p is $p(2B+1)/q$. By the union bound:

$$\Pr[\exists i \in [m] \colon \lfloor \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \rceil_p \neq \lfloor \langle \mathbf{a}_i, \mathbf{s} \rangle \rceil_p] \leq mp(2B+1)/q,$$

where the probability is taken over random and independent $\mathbf{a}_i \in \mathbb{Z}_q^n$ and $e_i \in [-B, B]$. Since this holds for every fixed $\mathbf{s} \in \mathbb{Z}_q^{n*}$ then it also holds for random $\mathbf{s}$. □

Claim 13. *Distributions H_1 and H_2 are $(t - m \cdot \mathrm{RD}, \varepsilon)$-indistinguishable.*

Proof. Suppose that there exists an oracle circuit D of size $t - m \cdot \mathrm{RD}$ that distinguishes between H_1 and H_2, and consider the circuit D' of size t that on input m pairs $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q \times \mathbb{Z}_q$ simulates D on input $(\mathbf{a}_i, \lfloor b_i \rceil_p) \in \mathbb{Z}_q \times \mathbb{Z}_p$. If $(\mathbf{a}_i, b_i)$ are LWE samples then the input fed to D is distributed as in H_1, whereas if $(\mathbf{a}_i, b_i)$ are random then the input fed to D is distributed as H_2. Thus, D' has the same distinguishing advantage as D, in contradiction to the (t, ε)-hardness of LWE. □

The theorem follows by combining the two claims with Proposition 1 and by observing that H_2 is distributed uniformly at random in $\mathbb{Z}_q^n \times \mathbb{Z}_p$. □

Proposition 6 gives a meaningful security guarantee only if $q \gg mp(2B+1)$. Nevertheless, the state of the art in attack algorithms [37, 13, 89, 130] indicates that, as long as q/p is an integer (so that $\lfloor x \rceil_p$ for a random $x \in \mathbb{Z}_q$ is random in $\mathbb{Z}_p$) and is at least $\Omega(\sqrt{n})$, LWR may be exponentially hard for any $p = \mathrm{poly}(n)$, and superpolynomially hard when $p = 2^{n^\varepsilon}$ for any $\varepsilon < 1$. It is open whether one could obtain worst-case hardness guarantees for LWR in such parameter regimes. ⍰

In some applications, such as the PRG described below, the parameter m can be fixed in advance, allowing smaller q. Several works have studied LWR in this setting [7, 40, 6]. In other applications, however, m cannot be a priori bounded. It is an open problem whether the dependency of q on m can be removed. ⍰

Instantiating GGM using an LWR-based PRG. The LWR problem yields a simple and practical pseudorandom generator $G_{\mathbf{A}} : \mathbb{Z}_q^n \to \mathbb{Z}_p^m$, where the moduli $q > p$ and the (uniformly random) matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ are publicly known [17]. Given a seed $\mathbf{s} \in \mathbb{Z}_q^n$, the generator is defined as

$$G_{\mathbf{A}}(\mathbf{s}) = \left\lfloor \mathbf{A}^{\mathsf{T}} \cdot \mathbf{s} \right\rceil_p, \tag{10}$$

where rounding is performed coordinate-wise. The generator's seed length (in bits) is $n \log_2 q$ and its output length is $m \log_2 p$, which gives an expansion rate of $(m \log_2 p)/(n \log_2 q) = (m/n) \log_q p$. For example, to obtain a length-doubling PRG, we may set $q = p^2 = 2^{2k} > n$ and $m = 4n$. In this case rounding corresponds to outputting the k most significant bits.

When evaluating the GGM construction instantiated with $G_{\mathbf{A}}$, one can get the required portion of $G_{\mathbf{A}}(\mathbf{s})$ by computing only the inner products of $\mathbf{s}$ with the corresponding columns of $\mathbf{A}$, not the entire product $\mathbf{A}^{\mathsf{T}} \cdot \mathbf{s}$. This becomes particularly attractive if one considers GGM trees with fan-in $d > 2$.

Instantiating NR using an LWR-based weak PRF. Consider the following weak pseudorandom function $W_{\mathbf{s}} \colon \mathbb{Z}_q^n \to \mathbb{Z}_p$, indexed by $\mathbf{s} \in \mathbb{Z}_q^n$:

$$W_{\mathbf{s}}(\mathbf{a}) = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_p. \tag{11}$$

Weak (t, m, ε)-pseudorandomness of $W_{\mathbf{s}}$ follows from (t, m, ε)-hardness of the LWR problem, using the fact that the $\mathbf{a}_i$ vectors are public [17]. To instantiate the NR construction of PRFs from synthesizers, one can

invoke Proposition 5, giving a synthesizer from a weak PRF. This requires the weak PRF's output length to match its input length. To this end, one can apply an efficient bijection, $K\colon \mathbb{Z}_p^{\ell\times\ell} \to \mathbb{Z}_q^{n\times\ell}$, for $\ell \geq n$ such that $p^\ell = q^n$, and modify the weak PRF from Equation (11) as follows:

$$W_{\mathbf{S}}(\mathbf{A}) := K\left(\lfloor \mathbf{A}^\mathsf{T} \cdot \mathbf{S} \rceil_p\right) \in \mathbb{Z}_q^{n\times\ell},$$

where $\mathbf{S}, \mathbf{A} \in \mathbb{Z}_q^{n\times\ell}$. The resulting synthesizer can be plugged into Equation (6) to give LWR-based PRFs $F_{\{\mathbf{S}_{i,b}\}}\colon \{0,1\}^k \to \mathbb{Z}_q^{n\times\ell}$. Security assuming $(t, m\ell, \varepsilon)$-hardness of LWE follows from combining Propositions 6 and 5 with Theorem 3. This results in a (t', ε')-pseudorandom function family, where $t' = t - \mathrm{poly}(n, m, \ell)$ and $\varepsilon' = O(\ell(k-1)(mp(2B+1)/q + \varepsilon))$. As a concrete example, the evaluation of this PRF when $k = 8$ (so $x = x_1 \cdots x_8$) unfolds as follows:

$$\left\lfloor \left\lfloor \lfloor \mathbf{S}_{1,x_1} \cdot \mathbf{S}_{2,x_2} \rceil_q \cdot \lfloor \mathbf{S}_{3,x_3} \cdot \mathbf{S}_{4,x_4} \rceil_q \right\rceil_q \cdot \left\lfloor \lfloor \mathbf{S}_{5,x_5} \cdot \mathbf{S}_{6,x_6} \rceil_q \cdot \lfloor \mathbf{S}_{7,x_7} \cdot \mathbf{S}_{8,x_8} \rceil_q \right\rceil_q \right\rceil_q,$$

where for clarity we let $\lfloor \mathbf{S}_{i,x_i} \cdot \mathbf{S}_{j,x_j} \rceil_q$ stand for $K\left(\lfloor \mathbf{S}_{i,x_i} \cdot \mathbf{S}_{j,x_j} \rceil_p\right)$.

A direct construction of PRFs. One drawback of the synthesizer-based PRF is that it involves $\log k$ levels of rounding operations, which appears to lower-bound the depth of any circuit computing the function by $\Omega(\log k)$. Aiming to get around this issue, BPR suggested to imitate the DDH-based construction, where sequential exponentiations are collapsed into one subset product. Since such a collapse is not possible in the case of LWR, they omitted all but the last rounding operation, resulting in a "subset-product with rounding" structure.

Public parameters: Moduli $q \gg p$

Function key: A random $\mathbf{a} \in \mathbb{Z}_q^n$ and k random short (B-bounded) $\mathbf{S}_i \in \mathbb{Z}_q^{n\times n}$

Function evaluation: On input $x \in \{0,1\}^k$ define $F = F_{\mathbf{a},\{\mathbf{S}_i\}}\colon \{0,1\}^k \to \mathbb{Z}_p^n$ as

$$F_{\mathbf{a},\{\mathbf{S}_i\}}(x_1 \cdots x_k) = \left\lfloor \mathbf{a}^\mathsf{T} \cdot \prod_{i=1}^{k} \mathbf{S}_i^{x_i} \right\rceil_p.$$

Size: $\mathrm{poly}(k, n)$

Depth: $O(1)$ (with threshold gates)

Figure 7: The Banerjee–Peikert–Rosen LWR-based construction.

The BPR function can be proved to be pseudorandom assuming that the LWE problem is hard. Two issues that affect the choice of parameters are the distribution of the secret key components $\mathbf{S}_i$, and the choice of q and p. For the former, the proof requires the $\mathbf{S}_i$ to be short. (LWE is no easier to solve for such short secrets [10].) This appears to be an artifact of the proof, which can be viewed as a variant of the LWE-to-LWR reduction from Proposition 6, enhanced to handle adversarial queries.

Theorem 5 ([17])**.** *If the LWE problem is (t, mn, ε)-hard for some B-bounded error distribution then $\{F_{\mathbf{a},\{\mathbf{S}_i\}}\}$ is a (t', m, ε')-pseudorandom function family, where*

$$t' = t - m\max\{n, 2k\}\mathrm{OP} - O(nm^2) - n \cdot \mathrm{RD}, \qquad \varepsilon' = mnp(2n^k B^{k+1} + 1)/q + k\varepsilon n,$$

and OP *is the cost of a group operation in $\mathbb{Z}_q$.*

Proof. Define the function $P\colon \{0,1\}^k \to \mathbb{Z}_q^n$ as

$$P(x) = P_{\mathbf{a},\{\mathbf{S}_i\}}(x) := \mathbf{a}^\mathsf{T} \cdot \prod_{i=1}^{k} \mathbf{S}_i^{x_i} \tag{12}$$

to be the subset product inside the rounding operation. The fact that $F = \lfloor P \rceil_p$ lets us imagine adding independent error terms to each output of P. Consider then a related randomized function $\tilde{P}$ that computes the subset product by multiplying by each $\mathbf{S}_i^{x_i}$ in turn, but also adds a fresh error term immediately following each multiplication.

By LWE-hardness and using induction on k, the randomized function $\tilde{P}$ can be shown to be itself pseudorandom (over $\mathbb{Z}_q$), hence so is $\lfloor \tilde{P} \rceil_p$ (over $\mathbb{Z}_p$). Moreover, for every queried input, with high probability $\lfloor \tilde{P} \rceil_p$ coincides with $\lfloor P \rceil_p = F$, because P and $\tilde{P}$ differ only by a cumulative error term that is small relative to q (this is where we need to assume that $\mathbf{S}_i$'s entries are small). Finally, because $\lfloor \tilde{P} \rceil_p$ is a (randomized) pseudorandom function over $\mathbb{Z}_p$ that coincides with the deterministic function F on all queries, it follows that F is pseudorandom as well.
Specifically, consider the following three games:

R: Give adaptive oracle access to a random function $R\colon \{0,1\}^k \to \mathbb{Z}_q^n$.

P: Give adaptive oracle access to $P\colon \{0,1\}^k \to \mathbb{Z}_q^n$ defined in (12).

$\tilde{P}$: Give adaptive oracle access to $\tilde{P}\colon \{0,1\}^k \to \mathbb{Z}_q^n$ inductively defined as:

- For $i = 0$, define $\tilde{P}_0(\lambda) = \mathbf{a}^\mathsf{T}$, where λ is the empty string.
- For $i \geq 1$, and on input $(x,y) \in \{0,1\}^{i-1} \times \{0,1\}$, define $\tilde{P}_i\colon \{0,1\}^i \to \mathbb{Z}_q^n$ as

$$\tilde{P}_i(x,y) = \tilde{P}_{i-1}(x) \cdot \mathbf{S}_i^y + y \cdot \mathbf{e}_x, \tag{13}$$

  where $\mathbf{a}, \mathbf{S}_1, \ldots, \mathbf{S}_k$ are sampled at random and the $\mathbf{e}_x \in \mathbb{Z}_q^n$ are all sampled independently according to the B-bounded LWE error distribution.

The function $\tilde{P} = \tilde{P}_k$ is specified by $\mathbf{a}$, $\{\mathbf{S}_i\}$, and exponentially many vectors $\mathbf{e}_x$. The error vectors can be sampled "lazily", since the value of $\tilde{P}(x)$ depends only on $\mathbf{a}$, $\{\mathbf{S}_i\}$, and $\mathbf{e}_x$.

Lemma 1. *Games $\lfloor P \rceil_p$ and $\lfloor \tilde{P} \rceil_p$ are $(\infty, m, mnp(2n^k B^{k+1}+1)/q)$-indistinguishable.*

Proof. Observe that for $x \in \{0,1\}^k$

$$\begin{aligned}\tilde{P}(x) &= (\cdots((\mathbf{a}^\mathsf{T} \cdot \mathbf{S}_1^{x_1} + x_1 \cdot \mathbf{e}_\lambda) \cdot \mathbf{S}_2^{x_2} + x_2 \cdot \mathbf{e}_{x_1}) \cdots) \cdot \mathbf{S}_k^{x_k} + x_k \cdot \mathbf{e}_{x_1 \cdots x_{k-1}} \bmod q \\ &= \underbrace{\mathbf{a}^\mathsf{T} \cdot \prod_{i=1}^{k} \mathbf{S}_i^{x_i}}_{P(x)} + \underbrace{x_1 \cdot \mathbf{e}_\varepsilon \cdot \prod_{i=2}^{k} \mathbf{S}_i^{x_i} + x_2 \cdot \mathbf{e}_{x_1} \cdot \prod_{i=3}^{k} \mathbf{S}_i^{x_i} + \cdots + x_k \cdot \mathbf{e}_{x_1 \cdots x_{k-1}}}_{\mathbf{e}_x} \bmod q.\end{aligned}$$

Now since both $\mathbf{S}_i$ and $\mathbf{e}_i$ are sampled from a B-bounded distribution, then each entry of an "error term" vector $\mathbf{e}_x$ is bounded by $n^k B^{k+1}$ (the magnitude being dominated by the entries of $\mathbf{e}_\varepsilon \cdot \prod_{i=2}^{k} \mathbf{S}_i^{x_i}$). By an analogous argument to the one in the proof of Proposition 6, it follows that, for every fixed choice of $\mathbf{S}_1, \ldots, \mathbf{S}_k$,

$$\Pr_{\mathbf{a}}\left[\exists x\colon \ \lfloor P(x) + \mathbf{e}_x \rceil \neq \lfloor P(x) \rceil_p\right] \leq mnp(2n^k B^{k+1}+1)/q.$$

Since this holds for every choice of $\mathbf{S}_i$'s, it also holds for a random choice. □

Lemma 2. *Games $\lfloor \tilde{P} \rceil_p$ and $\lfloor R \rceil_p$ are $(t - 2mk\text{OP} - n\text{RD}, m, k\varepsilon n)$-indistinguishable.*

Proof. For $i \in [k]$, consider the following games:

R_i: Give adaptive oracle access to a random function $R_i \colon \{0,1\}^i \to \mathbb{Z}_q^n$.

$\tilde{P}_i$: Give adaptive oracle access to the function $\tilde{P}_i \colon \{0,1\}^i \to \mathbb{Z}_q^n$ as defined in (13).

H_i: Give adaptive oracle access to the function $H_i \colon \{0,1\}^i \to \mathbb{Z}_q^n$, defined as

$$H_i(x,y) = \mathbf{a}_x \cdot \mathbf{S}_i^y + y \cdot \mathbf{e}_x, \tag{14}$$

where $(x,y) \in \{0,1\}^{i-1} \times \{0,1\}$, and $\mathbf{a}_x$, $\mathbf{S}_i$, and $\mathbf{e}_x$ are all sampled at random.

We prove inductively that $\tilde{P}_k$ and R_k are $(t - 2mk\text{OP}, m, k\varepsilon n)$-indistinguishable. For the induction basis we have that $\tilde{P}_0$ and R_0 are $(\infty, m, 0)$-indistinguishable by definition, and so are in particular $(t, m, 0)$-indistinguishable. For the inductive step, suppose that $\tilde{P}_{i-1}$ and R_{i-1} are $(t - 2m(i-1)\text{OP}, m, (i-1)\varepsilon n)$-indistinguishable.

Claim 14. *Games $\tilde{P}_i$ and H_i are $(t - 2mi\text{OP}, m, (i-1)\varepsilon n)$-indistinguishable.*

Proof. Suppose that there exists a circuit D of size $t - 2mi\text{OP}$ that distinguishes between $\tilde{P}_i$ and H_i with advantage $(i-1)\varepsilon n$ using m queries. We use D to build an A that $(t - 2m(i-1)\text{OP}, m, (i-1)\varepsilon n)$-distinguishes between $\tilde{P}_{i-1}$ and R_{i-1}.

The distinguisher A starts by sampling a random $\mathbf{S} \in \mathbb{Z}_q^{n \times n}$. Then, given a query of the form $(x,y) \in \{0,1\}^{i-1} \times \{0,1\}$ from D, it queries its oracle with input x to obtain $\mathbf{a}_x$, and replies to D with $\mathbf{a}_x \cdot \mathbf{S}^y + y \cdot \mathbf{e}_x$, using a random LWE error $\mathbf{e}_x$.

If A's oracle is distributed according to $\tilde{P}_{i-1}$, then A's replies to D are distributed as $\tilde{P}_i$. On the other hand, if A's oracle is distributed according to R_{i-1}, then $\mathbf{a}_x = R_{i-1}(x)$ is random, and so A's replies are distributed as H_i. Thus, A has the same advantage as D. Accounting for the two additional operations required by A for each of the m queries made by D we get that A is of size $t - 2mi\text{OP} + 2m\text{OP}$. □

Claim 15. *Games H_i and R_i are $(t - mn\text{OP} - O(nm^2), m, \varepsilon n)$-indistinguishable.*

Proof. Using a hybrid argument (akin to the proof of Proposition 2) it can be shown that the (t, mn, ε)-hardness of LWE implies that the following two distributions are $(t - mn\text{OP}, m, \varepsilon n)$-indistinguishable:

- Sample a random $\mathbf{S} \in \mathbb{Z}_q^{n \times n}$ and output $((\mathbf{a}_1, \mathbf{b}_1) \ldots, (\mathbf{a}_m, \mathbf{b}_m)) \in (\mathbb{Z}_q^n \times \mathbb{Z}_q^n)^m$, where the $\mathbf{a}_j$'s are uniformly random and $\vec{b}_j = \mathbf{a}_j^\mathsf{T} \cdot \mathbf{S} + \mathbf{e}_j \bmod q$ for random $\mathbf{e}_j$.
- Output m uniformly random pairs $((\mathbf{a}_1, \mathbf{b}_1) \ldots, (\mathbf{a}_m, \mathbf{b}_m)) \in (\mathbb{Z}_q^n \times \mathbb{Z}_q^n)^m$.

Suppose that there exists a distinguisher D of size $t - mn\text{OP} - O(nm^2)$ that distinguishes between H_i and R_i with m queries and advantage εn. We use D to build a distinguisher A that $(t - mn\text{OP}, m, \varepsilon n)$-distinguishes the two distributions from above.

Given m pairs $(\mathbf{a}_j, \mathbf{b}_j) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n$, the distinguisher A emulates an oracle for D as follows: for $j \in [m]$, answer the query $(x_j, y_j) \in \{0,1\}^{i-1} \times \{0,1\}$ given by D with $\mathbf{a}_j$ if $y_j = 0$ and with $\mathbf{b}_j$ if $y_j = 1$. Similarly to Theorem 2, $O(nm^2)$ additional gates are required for A to memorize previous answers so that he can answer consistently.

If $\vec{b}_j = \mathbf{a}_j^\mathsf{T} \cdot \mathbf{S} + \mathbf{e}_j \bmod q$ then the replies given by the above oracle are distributed exactly as in (14) (with $\mathbf{S}_i = \mathbf{S}$), and hence according to H_i. On the other hand, if $\vec{b}_j$ is random, then the oracle's replies are random and independent and hence distributed according to R_i. Thus A has the same advantage as D. □

Combining the two claims using Propositions 1 and 2, we get that games $\tilde{P}_i$ and R_i are $(\min\{t - mn\text{OP} - O(nm^2), t - 2mi\text{OP}\}, m, (i-1)\varepsilon n + \varepsilon n)$-indistinguishable. Thus, games $\tilde{P}_k$ and R_k are $(t - m\max\{n, 2k\}\text{OP} - O(nm^2), m, k\varepsilon n)$-indistinguishable.

Since $\tilde{P}_k = \tilde{P}$ and $R_k = R$, we finally conclude that the games $\lfloor \tilde{P} \rceil$ and $\lfloor R \rceil$ are $(t - m\max\{n, 2k\}\text{OP} - O(nm^2) - n \cdot \text{RD}, m, k\varepsilon n)$-indistinguishable. □

The theorem now follows by combining Lemma 1 with Lemma 2 via the triangle inequality (Proposition 1), and by observing that $F = \lfloor P \rceil$ and that $\lfloor R \rceil$ is a random function from $\{0,1\}^k$ to $\mathbb{Z}_p^n$ (assuming p divides q). □

Parameters. In the proof, the gap between P and $\tilde{P}$ grows exponentially in k, because noise is added after each multiplication by an $\mathbf{S}_i$. So in order to ensure that $\lfloor \tilde{P} \rceil_p = \lfloor P \rceil_p$ on all queries, we need both q and $1/\alpha$ to exceed $(nB)^{k+1} = n^{\Omega(k)}$. However, as in Proposition 6, it is unclear whether such large parameters are necessary, or whether $\mathbf{S}_i$ really need to be short. It would be desirable to have a security reduction for smaller q and $1/\alpha$, ideally both $\text{poly}(n)$ even for large k. ⍰

It would be even better if the construction were secure if the $\mathbf{S}_i$ were uniformly random in $\mathbb{Z}_q^{n \times n}$, because one could then recursively compose the function in a k-ary tree to rapidly extend its input length. ⍰

One reason for optimism (beyond the fact that no attacks are known) is that the PRF does not actually expose any low-error-rate LWE samples to the attacker; they are used only in the proof as part of a thought experiment. This appears to be related to the so-called *hidden number* problem (see, e.g., [2]). It would be interesting to investigate whether there exist connections between the problems. ⍰

A closely related PRF, due to Banerjee and Peikert [16], is described in Section 8.1. This PRF achieves the tightest known tradeoffs between LWE-based security and parallellism, though if one were to instantiate it with concrete parameters, then it would still be slower than instantiations of the subset product with rounding PRF with comparable security (see description of the SPRING family below).

Efficiency. The fastest instantiations of the PRF use ring elements instead of matrices. In the ring variant, $\mathbf{a}$ is replaced with a uniform $a \in R_q$ for some polynomial ring R_q (e.g., $\mathbb{Z}_q[X]/(X^n+1)$ for n a power of 2), and each $\mathbf{S}_i$ by some $s_i \in R_q^*$, the set of invertible ring elements modulo q.[4] This function is particularly efficient to evaluate using the discrete Fourier transform, as is standard with ring-based primitives (see, e.g., [96, 97]). Similarly to [111], one can optimize the subset-product operation via preprocessing, and evaluate the function in TC^0.

The functions are amenable to the same key-compression and amortization tehniques as the one used to optimize the performance of the NR DDH-based PRFs.

The SPRING family of PRFs ([15]). The SPRING (Subset Product with Rounding over a Ring) family of functions is a concrete instantiation of the PRF described in Figure 7 (aiming for 128-bit security), with parameters

$$n = 128, \quad q = 257, \quad p = 2, \quad k = 128.$$

Using "Gray code" amortization, an implementation of SPRING was shown to perform as fast as 4x slower than that of AES-128 (in software) with further potential optimization [15, 48]. We describe the main ideas behind the implementation.

[4] Here, hardness is based on the *ring*-LWE problem [97], in which we are given noisy/rounded ring products $b_i \approx a_i \cdot s$, where s and the a_i are random elements of R_q, and the error terms are "small" in a certain basis of the ring; the goal again is to distinguish these from uniformly random pairs.

The key, consisting of $a, s_1, \ldots, s_k \in R_q^*$, is stored as vectors in $\mathbb{Z}_q^n$ using the DFT or "Chinese remainder" representation mod q (that is, by evaluating a and the s_i as polynomials at the n roots of $X^n + 1 \bmod q$), so that multiplication of two ring elements corresponds to coordinate-wise product. Then to evaluate the function, one computes a subset product of the appropriate vectors, interpolates the result using an n-dimensional FFT over $\mathbb{Z}_q$, and rounds coordinate-wise. For $k = \omega(\log n)$, the runtime is dominated by the kn scalar multiplications in $\mathbb{Z}_q$ to compute the subset product; in parallel, the arithmetic depth (over $\mathbb{Z}_q$) is $O(\log(nk))$.

The subset-product part of the function might be computed even faster by storing the discrete logs of the Fourier coefficients of a and s_i, with respect to some generator g of $\mathbb{Z}_q^*$. The subset product then becomes a subset sum, followed by exponentiation modulo q, which can be implemented by table lookup if q is relatively small. Assuming that additions modulo $q-1$ are significantly less expensive than multiplications modulo q, the sequential running time is dominated by the $O(n \log n)$ scalar operations in the FFT, and the parallel arithmetic depth is again $O(\log n)$.

5 Transformations

Do you have the notion of a refill?
SHAFI GOLDWASSER (1985)

The significance of pseudorandom functions can be partly explained by their remarkable robustness and flexibility. In Section 2.3 we saw that the size of the function range is essentially irrelevant in the PRF definition. PRFs are similarly robust with respect to the choice of domain size: the domain can be easily enlarged and restricted for strong and weak PRFs alike. Domain extension can be accomplished with the help of *pairwise independence*, a restricted notion of pseudorandomness for which simple, unconditional constructions are available.

With regard to flexibility, PRFs serve as the main building block used in the construction of more complex pseudorandom functionalities. The best-known example is that of *pseudorandom permutations*, but the paradigm can be applied more generally to obtain succinct implementations of various *huge random objects* such as pseudorandom codes and pseudorandom graphs.

5.1 Domain Extension (and Restriction)

The domain extension problem is to efficiently construct a PRF on long inputs from a PRF on relatively short inputs. The GGM and NR constructions can be viewed as domain extension procedures in which the input length grows by one bit and doubles in every stage, respectively. Consequently, the complexity of the domain-extended PRF grows with the resulting input size (at different rates in the two constructions).

We present a domain extension procedure of Levin [90] in which the efficiency of the original PRF is essentially preserved. However, unlike in the GGM and NR constructions, the original domain size affects the security of the domain-extended PRF. Levin's construction makes use of pairwise independent functions.

Definition 7 (Pairwise independence)**.** A family $\{H_s \colon \{0,1\}^k \to \{0,1\}^{k'}\}$ of functions is *pairwise independent* if it is perfectly indistinguishable from a random function by any distinguisher that makes at most two queries.

In other words, $\{H_s\}$ is $(\infty, 2, 0)$-indistinguishable from random. This is a special case of the notion of bounded independence (see Definition 12 in Section 7). Pairwise independent hash families can have size as small as linear in $k + k'$ [80].

Theorem 6. *If $H_s\colon \{0,1\}^k \to \{0,1\}^{k'}$ is a pairwise independent family of functions and $F'_{s'}\colon \{0,1\}^{k'} \to \{0,1\}^\ell$ is a (t,q,ε)-PRF then the function $F_{s,s'}(x) = F'_{s'}(H_s(x))$ is a $(t-c,q,\varepsilon+\binom{q}{2}\cdot 2^{-k})$-PRF, where c is the circuit size of H_s.*

Proof. We analyze the advantage of the distinguisher with respect to the following sequence of games:

F': Sample s and s'. Answer query x by $F'_{s'}(H_s(x))$.

R': Sample s and $R'\colon \{0,1\}^{k'} \to \{0,1\}^\ell$. Answer query x by $R'(H_s(x))$.

R: Sample $R\colon \{0,1\}^k \to \{0,1\}^n$. Answer query x by $R(x)$.

Games F' and R' can be shown to be $(t-c,q,\varepsilon)$-indistinguishable using a standard simulation argument.

Claim 16. *Games R' and R are $(\infty,q,\binom{q}{2}\cdot 2^{-k/2})$-indistinguishable.*

Proof. We will assume, without loss of generality, that the distinguisher's queries are pairwise distinct. We relate R' and R to the following pair of games:

C: Sample s. Answer query x by `collision` if $H_s(x) = H_s(x')$ for some previously queried x', and by $\perp$ if not.

$\perp$: Answer every query by $\perp$.

If games C and $\perp$ are (∞,q,ε)-indistinguishable so must be R' and R: Unless a `collision` occurs, the answers of R' and R are identically distributed (to a sequence of independent random strings).

Any distinguisher between C and $\perp$ is essentially nonadaptive: The query sequence $x_1,\ldots,x_q$ can be extracted by assuming that the distinguisher interacts with the $\perp$ oracle. Its advantage equals the probability of a `collision`, which can be bounded by

$$\Pr[\texttt{collision}] \le \sum_{1\le i<j\le q} \Pr[H_s(x_i) = H_s(x_j)] = \binom{q}{2}\cdot 2^{-k}.$$

The inequality is obtained by taking a union bound over all pairs of queries, while the equality follows from pairwise independence. □

The theorem follows by applying the triangle inequality. □

The security guarantee in Theorem 6 becomes meaningless when the number of queries exceeds $2^{k/2}$. This is unavoidable by the birthday paradox: A distinguisher that looks for collisions among $2^{k/2}$ random queries has constant advantage. Berman, Haitner, Komargodski, and Naor [29] give a different domain extension procedure with improved security: Their construction uses two independent instances of the PRF F'_s and has security that is independent of the input length k, as long as the number of queries is at most 2^{k-2}.

Domain restriction and range extension for synthesizers. Naor and Reingold [110] consider domain *restriction*—namely, the problem of reducing the input length—for synthesizers. They prove the following statement, which follows by an application of Proposition 2:

Proposition 7. *If S' is a (t,q,ε)-synthesizer and $G\colon \{0,1\}^k \to \{0,1\}^{k'}$ is a $(t+c,\varepsilon')$-PRG then the function $S(x) = S'(G(x))$ is a $(t,q,\varepsilon+2q\varepsilon')$-synthesizer, where c is the circuit size of S'.*

We do not know if an analogous transformation exists for weak PRFs. ⑦ Proposition 7 can be combined with the following range extension construction to convert a synthesizer with sufficiently long input and one bit of output into a length-preserving synthesizer:

Proposition 8. *If $S'\colon \{0,1\}^k \to \{0,1\}$ is a $(t + cq^2\ell, q, \varepsilon)$-synthesizer then $S((x_1,\ldots,x_\ell),(y_1,\ldots,y_\ell)) = (S'(x_1,y_1),\ldots,S'(x_\ell,y_\ell))$ is a $(t,q,\ell\varepsilon)$-synthesizer, where c is the circuit size of S'.*

5.2 Pseudorandom Permutations

Nu, permutaziot!

ABRAHAM LEMPEL (1981)

A pseudorandom permutation (PRP) is a permutation that is easy to evaluate, but hard to distinguish from a random permutation. PRPs are a model of block ciphers, which are used to implement various modes of operation in symmetric-key encryption and authentication schemes. PRPs will also serve as an illustrative example of the "huge random objects" discussed in Section 5.3.

Definition 8 (Pseudorandom permutation [93])**.** A family of permutations $F_s\colon \{0,1\}^k \to \{0,1\}^k$ is (t,q,ε)-*pseudorandom* if the following two games are (t,q,ε)-computationally indistinguishable:

F_s: Sample a random $s \leftarrow \{0,1\}^n$ and answer query $x \in \{0,1\}^k$ by $F_s(x)$,

P: Sample a random permutation $P\colon \{0,1\}^k \to \{0,1\}^k$ and answer x by $P(x)$.

The security requirement of Definition 8 is met by any pseudorandom *function* family F_s:

Proposition 9. *If F_s is a (t,q,ε)-PRF then games F_s and P are $(t,q,\varepsilon + \binom{q}{2}2^{-k})$-indistinguishable.*

Proof. Consider the game

R: Sample a random function $R\colon \{0,1\}^k \to \{0,1\}^k$ and answer query x by $R(x)$.

Claim 17. *Games R and P are $(\infty, q, \binom{q}{2}2^{-k})$-indistinguishable.*

Proof. We may assume, without loss of generality, that the distinguisher's queries are pairwise distinct. We analyze the hybrid game H_i in which the first i queries $x_1,\ldots,x_i$ are answered using P and the rest are answered using R. Games H_{i-1} and H_i are identically distributed conditioned on $R(x_i)$ taking a different value from $P(x_1),\ldots,P(x_{i-1})$. The probability this fails is $(i-1)2^{-k}$. The claim follows by the triangle inequality. □

As F_s and R are (t,q,ε)-indistinguishable, Proposition 9 follows by applying the triangle inequality again. □

In general, the functionality requirement of being a permutation may not be satisfied by certain PRFs such as those obtained via the GGM and NR constructions. Luby and Rackoff [93] describe a generic transformation for constructing a PRP from any length-preserving PRF.

Both of these constructions are based on the Feistel shuffle

$$\mathbf{Fei}[F](x,y) = (y, x + F(y)), \qquad x,y \in \{0,1\}^{k/2},$$

where F is a function from $\{0,1\}^{k/2}$ to $\{0,1\}^{k/2}$ and $+$ is bit-wise XOR. The function $\mathbf{Fei}[F]$ is a permutation on $\{0,1\}^n$; its inverse is $(x,y) \mapsto (y+F(x),x)$.

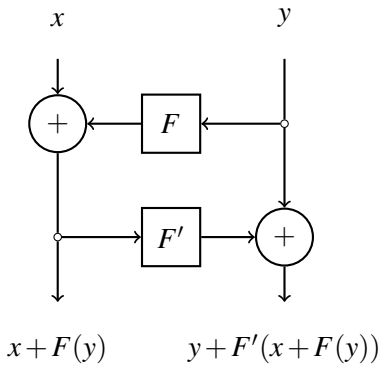

Figure 8: The two-round Feistel network. The intermediate outputs are flipped for clarity

The Feistel permutation is clearly not pseudorandom (regardless of the choice of F) as its output reveals half of its input. The starting point of the PRP construction is the composition $\mathbf{Fei}^2[F,F'] = \mathbf{Fei}[F'] \circ \mathbf{Fei}[F]$ for "independent" functions F and F' (see Figure 8). This is the permutation

$$\mathbf{Fei}^2(x,y) = \big(x+F(y), y+F'(x+F(y))\big).$$

The permutation $\mathbf{Fei}^2$ also fails to be pseudorandom (regardless of the choice of F and F') as its outputs satisfy the relation

$$\mathbf{Fei}^2(x,y) + \mathbf{Fei}^2(x',y) = (x+x', \text{something}) \tag{15}$$

for any pair of queries of the form (x,y) and (x',y). It turns out that this is the only nonrandom feature of the permutation, in the following sense:

Proposition 10. *For every q, the following two games are $(\infty, q, q(q-1)\cdot 2^{-k/2})$-indistinguishable:*

F: Sample $F,F'\colon \{0,1\}^{k/2} \to \{0,1\}^{k/2}$. Answer query (x,y) by $Fei^2[F,F'](x,y)$.

P: Sample a random permutation $P\colon \{0,1\}^k \to \{0,1\}^k$ and answer with $P(x,y)$,

assuming all queries made by the distinguisher are ynique.

Definition 9 (ynique sequence)**.** We call a sequence $(x_1,y_1),\ldots,(x_q,y_q)$ *y*nique if all y-components are distinct (i.e., $y_i \neq y_j$ when $i \neq j$).

Proof. We analyze the advantage of the distinguisher with respect to the sequence F,R,P, where R is the game

R: Sample $R\colon \{0,1\}^k \to \{0,1\}^k$. Answer query (x,y) by $R(x,y)$.

By Proposition 9, games R and P are $(\infty, q, \binom{q}{2}\cdot 2^{-k/2})$-indistinguishable. It remains to analyze the distinguishing advantage between F and R.

Claim 18. *Games F and R are $(\infty, q, \binom{q}{2} \cdot 2^{-k/2})$-indistinguishable, assuming the sequence of queries made by the distinguisher is ynique.*

Proof. Consider the hybrid H_i in which the first i queries $(x_1, y_1), \ldots, (x_i, y_i)$ are answered as in F and the rest are answered as in R. We will show that H_i and H_{i-1} are $(\infty, \infty, (i-1) \cdot 2^{-k/2})$-indistinguishable. The claim then follows by the triangle inequality.

The first i outputs of H_i are

$$\big(x_j + F(y_j), y_j + F'(x_j + F(y_j))\big)_{j=1,\ldots,i-1},\ \big(x_i + F(y_i), y_i + F'(x_i + F(y_i))\big)$$

By $y$niqueness, $F(y_i)$ is random and independent of $F(y_1), \ldots, F(y_{i-1})$, F', as well as the other $q-i$ outputs of H_i, which can be fixed to maximize the distinguishing advantage. We can therefore represent this distribution as

$$\big(x_j + F(y_j), y_j + F'(x_j + F(y_j))\big)_{j=1,\ldots,i-1},\ \big(x_i + r, y_i + F'(x_i + r)\big).$$

for a random $r \leftarrow \{0,1\}^k$. The probability that $x_i + r = x_j + F(y_j)$ for some $j < i$ is at most $(i-1) \cdot 2^{-k/2}$. Conditioned on this event not happening, $F'(x_i + r)$ is independent of all the other $j-1$ evaluations of F and F' and of r. Changing notation again, we can represent the distribution as

$$\big(x_j + F(y_j), y_j + F'(x_j + F(y_j))\big)_{j=1,\ldots,i-1},\ \big(x_i + r, y_i + r'\big).$$

for a random $r' \leftarrow \{0,1\}^k$. The pair $(x_i + r, y_i + r')$ is uniformly random. By $y$niqueness it can be replaced with $R(x_i, y_i)$ as in the distribution H_{i-1}. □

The proposition now follows from the triangle inequality. □

The Luby–Rackoff and Naor–Reingold constructions. The requirement that all queries have distinct y-coordinates can be enforced by preprocessing the queries. Luby and Rackoff apply another Feistel round for this purpose. Here we describe a variant of Naor and Reingold [109], who use a pairwise independent permutation instead. A family of permutations $H_s \colon \{0,1\}^k \to \{0,1\}^k$ is *pairwise independent* if it is perfectly indistinguishable from a random permutation by any distinguisher that makes at most two queries. One simple example is the family $H_{a,b}(x) = a \cdot x + b$ where $a \leftarrow \mathbb{F}_{2^k}^{\times}$, $b, x \leftarrow \mathbb{F}_{2^k}$, and the operations are performed over the field $\mathbb{F}_{2^k}$.

Theorem 7. *If $\{F_s\}$ is a (t, q, ε)-PRF and H_s is a pairwise independent family of permutations then the function $\mathbf{Fei}^2[F_{s'}, F_{s''}] \circ H_s$ is a $(t - O(kq^2 + ckq), q, \frac{3}{2}q(q-1) \cdot 2^{-k/2} + 2\varepsilon)$-PRP, assuming F_s and H_s have circuit size at most c.*

Proof. We may assume that the distinguisher never makes the same query twice by modifying it to memorize its previous answers. This incurs a loss of at most $O(kq^2)$ in size. Consider the sequence of games

F_s: Sample s, s', s'' independently. Answer by $\mathbf{Fei}^2[F_{s''}, F_{s'}](H_s(x, y))$.

I: Sample s and $F, F' \colon \{0,1\}^{k/2} \to \{0,1\}^{k/2}$. Answer by $\mathbf{Fei}^2[F', F](H_s(x, y))$.

P: Sample H_s and a random permutation P on $\{0,1\}^k$. Answer by $P(H_s(x, y))$.

Games F_s and I are $(t - O(kq^2 + ckq), q, 2\varepsilon)$-indistinguishable by an analysis as in the proof of Theorem 3. Game P is perfectly indistinguishable from a random permutation. To show indistinguishability of I and P we need the following claim. Let $(x_1, y_1), \ldots, (x_q, y_q)$ denote the query sequence.

Claim 19. *The probability that the sequence $(H_s(x_i, y_i))_{i=1,\ldots,q}$ is not ynique in game P is at most $\binom{q}{2} \cdot 2^{-k/2}$.*

Proof. By the same argument used in the proof of Theorem 6, it can be assumed without loss that the distinguisher makes its queries nonadaptively. Writing (x'_i, y'_i) for $H(x_i, y_i)$,

$$\Pr[(x'_i, y'_i)_{i=1,\ldots,q} \text{ is not ynique}] \leq \sum_{1 \leq i < j \leq q} \Pr[y'_i = y'_j] = \binom{q}{2} \cdot \frac{2^{-k/2} - 2^{-k}}{1 - 2^{-k}}.$$

The inequality follows from the union bound, and the equality follows from pairwise independence of H_s. After simplifying we obtain the desired bound. □

Claim 20. *Games I and P are $(\infty, q, \frac{3}{2}q(q-1) \cdot 2^{-k/2})$-indistinguishable.*

Proof. Consider the following pair of games:

$I^\star$: Same as F, but `fail` if $H_s(x, y)$ is not ynique.

$P^\star$: Same as P', but `fail` if $H_s(x, y)$ is not ynique.

By the above claim, $P^\star$ and P are $(\infty, q, \binom{q}{2} \cdot 2^{-k/2})$-indistinguishable. By Proposition 10, $I^\star$ and $P^\star$ are $(\infty, q, q(q-1) \cdot 2^{-k/2})$-indistinguishable. Applying the triangle inequality, $I^\star$ and P are $(\infty, q, \frac{3}{2}q(q-1) \cdot 2^{-k/2})$-indistinguishable. The distinguishing advantage cannot increase when $I^\star$ is replaced by I, proving the claim. □

The theorem follows by applying the triangle inequality. □

The pairwise independence of H_s is only used in the proof to ensure that the y-components of the sequence $H_s(x_i, y_i)$ are pairwise pseudorandom. Luby and Rackoff accomplish the same effect with an initial Feistel round.

Strongly pseudorandom permutations. A family of permutations is *strongly pseudorandom* if security holds even against adversaries that are allowed to query both the permutation P and its inverse P^{-1}. This property is required in certain cryptographic applications. The construction from Theorem 7 may not be strongly pseudorandom as the inverse permutation satisfies relations analogous to (15). Strong pseudorandomness can be achieved by adding another hashing step at the output, or via an additional Feistel round (see Figure 9).

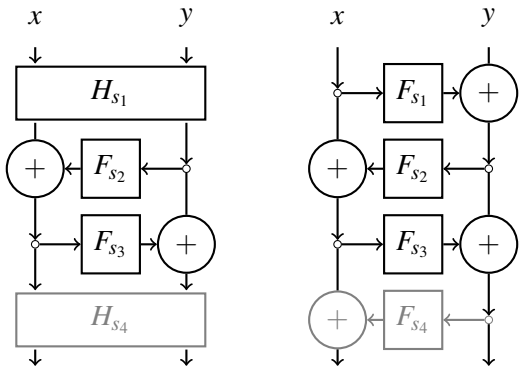

Figure 9: The pseudorandom permutations of Naor–Reingold and Luby–Rackoff. The last layer is needed for strong pseudorandomness

Security versus domain size. In Theorem 7, the parameter k governs both the input size and the security guarantee of the pseudorandom permutation. The security guarantee is poor for PRPs on small domains, which are useful in practice [34]. This is unavoidable for the Luby–Rackoff construction; Aiello and Venkatesan [1] proved that the four-round Feistel network is not $(\text{poly}(k) \cdot 2^{k/4}, 2^{k/4}, \frac{1}{2})$-pseudorandom. Maurer, Pietrzak, and Renner [99] show that increasing the number of Feistel rounds can improve the dependence on k in the security. However, their security guarantee is still inadequate for small values of k.

While it is not known in general if this dependence between security and input size is necessary for Feistel-type PRP instantiations, it is an inherent limitation of "information-theoretic" security proofs such as the proof of Theorem 7. There, the security of the PRP is deduced from the security of an idealized game I in which the underlying PRF instances are replaced by truly random functions. It is then shown that the game I is statistically secure: Any q-query adversary, regardless of its size, distinguishes I from a random permutation P with probability at most $O(q^2 \cdot 2^{-k/2})$.

A counting argument shows that the analogues of games I and P for the r-round Feistel network $\mathbf{Fei}^r[F_1, \ldots, F_r]$ are *not* $(\infty, r2^{k/2}, \frac{1}{2})$-indistinguishable [103]. To see this, consider the sequence of permutation values at the lexicographically first q inputs. The permutation $\mathbf{Fei}^r$ is fully specified by the truth-tables of the r underlying PRFs $F_1, \ldots, F_r \colon \{0,1\}^{k/2} \to \{0,1\}^{k/2}$, so the sequence $(\mathbf{Fei}^r(1), \ldots, \mathbf{Fei}^r(q))$ can by described by at most $rk/2 \cdot 2^{k/2}$ bits. On the other hand, for a random permutation P, the sequence $(P(1), \ldots, P(q))$ has min-entropy $\log(2^k)_q \geq q \log(2^k - q)$. The two can be distinguished with constant advantage when $q \geq r2^{k/2}$.

In summary, the security analysis of the Feistel construction is limited by the relatively small input size of the underlying (pseudo)random functions, which is an artifact of its balanced nature—namely, the requirement that x and y should be equally long. It is therefore sensible to investigate the security of unbalanced variants. In the extreme setting $|x| = 1$, $|y| = k - 1$ the analog of the Feistel shuffle is the Thorp shuffle

$$\mathbf{Th}[F](x, y) = (y, x + F(y)), \qquad x \in \{0,1\}, y \in \{0,1\}^{k-1}$$

with underlying (pseudo)random function $F \colon \{0,1\}^{k-1} \to \{0,1\}$. Morris, Rogaway, and Stegers [103] show that the r-round Thorp network $\mathbf{Th}^r$ (instantiated with random functions) is $(\infty, q, (q/r^\star) \cdot (4kq/2^k)^{r^\star})$-indistinguishable from a random permutation, where $r^\star = r/(2k+1)$. In the case $q = 2^k$, Morris [102] proves that $r = O(k^3 \log 1/\varepsilon)$ rounds yield $(\infty, 2^k, \varepsilon)$ security for any k and ε.

The Feistel shuffle $\mathbf{Fei}[F]$ and the Thorp shuffle $\mathbf{Th}[F]$ are examples of *oblivious* card-shuffling procedures: The permutation can be viewed as a rule for shuffling a deck of 2^k cards with the randomness described by the underlying (pseudo)random function F.[5] For the resulting PRP to be efficiently computable, the shuffle should be oblivious in the sense that the new position of every card in the deck can be computed efficiently as a function of only its previous position and the randomness, and not the positions of the other cards in the deck. Oblivious shuffles that enjoy rapid "local" mixing give rise to PRP constructions with a good tradeoff between efficiency and security.

5.3 Implementing Huge Random Objects

A pseudorandom permutation is an example of a huge object that is guaranteed to satisfy the global property of being a permutation, while being "locally" indistinguishable from a random permutation. An implementation of a pseudorandom permutation by a (pseudo)random function would preserve the local indistinguishability (by Proposition 9), but is likely to violate the global property. This may be relevant in applications where the user of the implementation may rely, for instance, on the existence of inverses.

[5] This perspective is attributed to Naor [109], who was the first to propose the Thorp shuffle-based PRP construction.

The distinction is even more prominent in the case of a strong pseudorandom permutation. In a "truthful" implementation, such as the four-round Feistel network, the user is guaranteed that evaluating $P_s^{-1}(P_s(x))$ always outputs x. In contrast, if P is instantiated with an arbitrary PRF, a consistent inverse may not even exist, much less be efficiently computable.

Goldreich, Goldwasser, and Nussboim [67] initiated a general study of efficient implementations of huge random objects of various types. The Luby–Rackoff construction suggests a generic two-step template for this purpose:

1. Starting from a random function R, construct an object O^R that is statistically indistinguishable from a random object of the desired type (for a suitable bound on the number of queries).

2. Replace the random function R by a PRF to obtain an efficient implementation O_s of the object.

Let us call the implementation O_s *truthful* if the object O_s is of the desired type.[6] Goldreich et al. observe that, even if one is willing to tolerate untruthful implementations with some small probability in the second step, the first step must guarantee a truthful random object with probability one. This phenomenon is illustrated in the following example:

Example 1 (Random injective function)**.** A random function $R\colon \{0,1\}^n \to \{0,1\}^{3n-1}$ is injective with probability at least $1-2^{-n}$. However, a PRF F_s with the same domain and range need not be injective for any key s. In fact, any PRF can be converted into a noninjective PRF by planting a random collision: If F_s is a (t,q,ε)-PRF then the family

$$F_{s,a}(x) = \begin{cases} F_s(x), & \text{if } x \neq a \\ F_s(0), & \text{if } x = a \end{cases}$$

is a $(t,q,\varepsilon + q\cdot 2^{-n})$-PRF that is not injective for any key (s,a), $a \neq 0$.

Example 1 can be explained by the fact that injectivity is a global property of functions, while step 2 only guarantees that local indistinguishability is preserved. In view of this it is interesting to ask if an almost always truthful implementation of a random injective function can be obtained. In this case the answer is positive.

Proposition 11. *If $F_s\colon \{0,1\}^k \to \{0,1\}^\ell$ is a (t,ε)-PRF and $H_s\colon \{0,1\}^k \to \{0,1\}^\ell$ is a pairwise-independent hash family then $I_{s,s'}(x) = F_s(x) \oplus H_{s'}(x)$ satisfies the following two properties:*

1. *$I_{s,s'}$ is $(t-c, \varepsilon + \binom{2^k}{2}\cdot 2^{-\ell})$-indistinguishable from a random injective function, where c is the size of $H_{s'}$.*

2. *$\Pr_{s,s'}[I_{s,s'}$ is not injective$] \leq \binom{2^k}{2}\cdot 2^{-\ell}$.*

Proof. The first property follows by a hybrid argument (details omitted). We prove that the second property holds for every fixing of s:

$$\begin{aligned} \Pr_{s'}[I_{s,s'} \text{ is not injective}] &\leq \sum_{x\neq x'} \Pr[I_{s,s'}(x) = I_{s,s'}(x')] \\ &= \sum_{x\neq x'} \Pr[H_{s'}(x) \oplus H_{s'}(x') = F_s(x) \oplus F_s(x')] \\ &= \binom{2^k}{2}\cdot 2^{-\ell}. \end{aligned}$$

The inequality is the union bound, and the last equality follows from the pairwise independence of $H_{s'}$. □

[6] Our use of the term deviates slightly from the definition in [67].

Can Proposition 11 be strengthened so as to also provide the distinguisher access to F^{-1}? More generally, $I_{s,s'}$ almost always truthfully implements a random code of linear distance. In contrast, a result of Lovett and Viola on the complexity of sampling [92] implies that implementing any linear code by the above two-step template is impossible. Goldreich et al. ask if there is an alternative implementation in which the distinguisher can be also furnished with a decoding oracle. ⑦

The work [67] contains many additional results and open questions regarding huge random objects arising from random graph theory and the theory of random Boolean functions.

Efficient implementations of huge random objects can enable an experimentalist to carry out simulations on random objects that are prohibitively large (e.g., random graphs, random codes) with results that are guaranteed to be sound, assuming the availability of a sufficiently strong PRF. In this setting, the stateless nature of PRFs is a desirable feature (as it provides a short description of the huge object) but not entirely necessary.

Bogdanov and Wee [41] introduce the notion of a *stateful implementation*, which may keep state in between queries. Their work gives a perfect stateful implementation of a specification suggested by Goldreich et al.: A Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ in which the distinguisher may query the XOR of all the values of f over a subcube of its choice. Ben-Sasson et al. [28] extend these results and apply them towards the construction of perfect zero-knowledge interactive proofs. By the work of Lovett and Viola [92], efficient perfect stateless implementations of such objects are impossible even with oracle access to a truly random function. We do not know if stateless implementations that are secure against distinguishers of bounded query complexity can be obtained in this model. More generally, it would be interesting to better undestand the differences between stateful and stateless implementations of huge random objects. ⑦

6 Complexity of Pseudorandom Functions

Chazak Razborov!

BENNY CHOR (1980S)

The GGM and NR constructions from Section 3 are generic methods for obtaining a PRF from a simpler pseudorandom object. The resulting PRF is in general more complex than the underlying primitive. In the specific instantiations discussed in Section 4, this increase in complexity was mitigated by careful implementation.

There is however a limit to the amount of efficiency that can be squeezed by further optimizations. PRF implementations inherently require a certain amount of complexity. We discuss two related reasons for this: the availablity of efficient learning algorithms and the existence of "natural" lower-bound proofs for sufficiently simple circuit classes. We then describe some heuristic PRF candidates that are just complex enough to arguably match these limitations.

6.1 Learning Algorithms

A learner for a class of functions $\mathbf{F}$ is a two-stage algorithm L that works as follows: In the first stage the algorithm is given oracle access to some function $F \in \mathbf{F}$. In the second stage the algorithm receives an input x and outputs a prediction for the value $F(x)$. The learner has *approximation error* δ if $\Pr[L^F(x) \neq F(x)] \leq \delta$ for x chosen from the uniform distribution on inputs.[7]

[7] Other distributions on inputs are also studied in learning theory.

Learning algorithms differ depending on the learner's mode of oracle access. A membership query learner may query the oracle adaptively on inputs of its choice. A learner from random examples has only access to random input–output pairs $(x, F(x))$ for independent and uniform inputs x.

The existence of a low-complexity learner for a class of functions implies bounds on the security of any implementation of a pseudorandom function family in this class [119]. We state the result for Boolean-valued functions $F_s \colon \{0,1\}^k \to \{0,1\}$.

Proposition 12. *If the function family $\{F_s\}$ can be learned from membership queries (resp., random examples) with approximation error $\frac{1}{2} - \varepsilon$ by an algorithm of size t then F_s is not a $(t + O(k), \varepsilon - t/2^k)$-PRF (resp., weak PRF).*

Learning with approximation error $\frac{1}{2}$ amounts to random guessing. Proposition 12 states that any non-negligible improvement to the approximation error by an efficient learner rules out pseudorandomness of F_s.

Proof. Consider the distinguisher D^F that simulates L^F and makes one additional random query x to obtain answer $F(x)$ (or, in the case of random examples, obtains one additional example $(x, F(x))$). If $L^F(x) = F(x)$ the distinguisher accepts, and otherwise it rejects.

By assumption, $L^{F_s}(x)$ equals $F_s(x)$ with probability at least $\frac{1}{2} + \varepsilon$. On the other hand, if R is a random function, $L^R(x)$ is statistically independent of $R(x)$ as long as the oracle has not been queried at x, which fails to happen with probability at most $t/2^k$. The advantage of the distinguisher is therefore at least $(\frac{1}{2} + \varepsilon) - (\frac{1}{2} + t/2^k) = \varepsilon - t/2^k$. □

We will say a pseudorandom function F_s can be implemented in class $\mathbf{F}$ if F_s belongs to $\mathbf{F}$ for all s. Applying Proposition 12 to different algorithms from the computational learning theory literature yields the following limitations on the implementation complexity of PRFs:

1. Weak PRFs cannot be implemented by linear threshold functions, as such functions can be learned efficiently using the algorithm of Blum et al. [35].

2. PRFs cannot be implemented by polynomial-size formulas in disjunctive normal form (DNF), as these can be learned efficiently (under the uniform distribution) by Jackson's harmonic sieve algorithm [81].

3. Any function family implemented by AND/OR circuits of size s, depth d, and input length k is not $(\binom{k}{\leq O(\log s)^{d-1}}, \frac{1}{2})$-weakly pseudorandom, as such circuits can be learned under the uniform distribution by the algorithm of Linial, Mansour, and Nisan [91].

The learning algorithms of Blum et al. and Linial, Mansour, and Nisan can be implemented in the statistical query model that is discussed in Section 7.7.

Proposition 12 can also be applied in the contrapositive form to argue computational limitations on learning. In particular:

1. Polynomial-size circuit families cannot be learned efficiently from membership queries, assuming polynomial-size one-way function families exist. This follows from the equivalence of one-way functions and PRFs discussed in Section 1.2.

2. Polynomial-size constant-depth circuit families with linear threshold gates (the class TC^0) cannot be learned efficiently from membership queries, assuming the hardness of DDH or LWE. This follows from the complexity of the constructions in Section 4.

3. Polynomial-size constant-depth AND/OR circuit families require quasipolynomial time to learn from membership queries, assuming the exponential hardness of factoring Blum integers [85].

6.2 Natural Proofs

I asked my wife what is the definition of natural, and she said "anything that does not contain petroleum products."

LEONID LEVIN (2007)

In an attempt to understand the difficulties inherent in proving lower bounds on the size of Boolean circuits computing explicit functions, Razborov and Rudich introduced a formal framework that captures many of the currently available techniques. A circuit lower-bound proof can be viewed as a property P of functions that distinguishes between the functions computable by circuits in the given class and the "hard" function.

Razborov and Rudich showed that, in many known proofs, the distinguishing property has the following two features. For convenience let us assume that the domain and range are $\{0,1\}^k$ and $\{0,1\}$, respectively.

Smallness: Property P fails not only for the hard function, but for, say, at least half the functions from $\{0,1\}^k$ to $\{0,1\}$.[8]

Constructivity: There exists an oracle circuit of size $2^{O(k)}$ that, given oracle access to F, decides if F has property P.

A small and constructive property that holds for all functions in a given class $\mathbf{F}$ is called *natural* for $\mathbf{F}$.

Proposition 13. *If there exists a property that is natural for the function family $\{F_s\}$ then F_s is not a $(2^{O(k)}, \frac{1}{2})$-PRF.*

It is useful to keep in mind that the (strong) PRF distinguisher can control the input length k of the candidate PRF by fixing some of the input bits.

Proof. Let D be the circuit of size $2^{O(k)}$ that decides if F has property P. By assumption, D^{F_s} always accepts. By smallness, $\Pr[D^R \text{ accepts}] \leq \frac{1}{2}$. Therefore D has distinguishing advantage at least $1/2$. □

Applying Proposition 13 to various properties implicit in circuit lower-bound proofs, Razborov and Rudich derive the following consequences among others:

1. Any function family implemented by AND/OR circuits of size $\exp o(k)^{1/(d-1)}$ and depth d is not pseudorandom. This follows from the parity circuit lower bound of Håstad [58, 138, 73]. Boppana [44] shows that the following simple distinguisher works: Choose two random inputs x, y that differ on a single coordinate and check if $F(x) = F(y)$. Distinguishers of this type are discussed in Section 7.1.

2. Any function family implemented by AND/OR/PARITY circuits of size s and depth d (the class $\mathrm{AC}^0[\oplus]$) is not $(\exp(\log s)^{O(d)}, \frac{1}{2})$-pseudorandom. This follows from a "naturalization" of the lower-bound proof of Razborov and Smolensky [123, 132]. The conclusion also holds if PARITY is replaced by the MOD_q function for any constant prime power q. This distinguisher is described as the Razborov–Rudich test in Section 7.4.

3. Any function family implemented by AND/OR formulas of size $k^{3-\varepsilon}$ for any $\varepsilon > 0$ and sufficiently large k is not $(2^{O(k)}, \frac{1}{2})$-pseudorandom. This follows from Håstad's proof of hardness for the Andreev function [74] (see also [39]).

[8] Razborov and Rudich work with the complementary property NOT P and call the corresponding condition largeness.

In summary, PRFs cannot be constructed in any class that is (a) learnable or (b) has a natural property. Learnability and natural properties are closely related. In one direction, learnability is a natural property, as the learning algorithm can be used to efficiently distinguish functions in the class from a random function. In the other direction, Naor and Reingold [108] and Nisan and Wigderson [115] give constructions of PRFs from functions that are hard to predict on a chosen and random input, respectively. Carmosino et al. [51] apply the latter transformation to obtain a quasipolynomial time learning algorithm for $\mathrm{AC}^0[\oplus]$ circuit families.

While these results essentially rule out the existence of PRFs of very low complexity, it remains an open question whether similar limitations hold for some of their immediate applications such as symmetric-key encryption schemes or authentication protocols. ⍰

6.3 Heuristic Constructions

Propositions 12 and 13 indicate that the efficiency of PRFs is of fundamental relevance in computational learning theory and computational complexity. These connections provide extrinsic motivation for a fine-grained study of the complexity of PRF constructions in various computational models.

In practice, the most efficient PRFs for a given level of security are not obtained by means of generic methods such as the ones from Section 3. The modular nature of such constructions appears to entail a loss in security which can be potentially avoided with a carefully crafted design. However, claims of security for "direct" PRF constructions can no longer be based on standard assumptions and must rely instead on the collective wisdom of cryptanalysts (motivated in part by social incentives for attacking candidate implementations).

The practical construction of PRFs is an intricate art form that we do not attempt to cover here. We recommend Chapter 5 in the textbook of Katz and Lindell [82] as an introduction to this subject. Our emphasis here is on elementary principles of direct PRF and PRP constructions. We mention some concrete proposals and discuss their relevance to the feasibility of learning and the existence of natural proofs.

Two paradigms that have been applied towards practical implementations of PRPs (called *block ciphers* in the applied cryptography literature) are Feistel networks and substitution-permutation networks. Both of these methods in fact yield (pseudorandom) permutations.

Feistel networks. The most well-known Feistel network-based PRP is DES (the Data Encryption Standard). Despite the long history and prominence of DES (it was designed and standardized in the 1970s and has enjoyed widespread use ever since), it has shown remarkable resilience to cryptanalysis. DES is a permutation family on 64-bit strings with a 56-bit key. Its basic building block is the 16-round Feistel network $\mathbf{Fei}^{16}[F_{s_1}, \ldots, F_{s_{16}}]$ instantiated with some special function F_s that "mixes" the input and its key. The *round keys* $s_1, \ldots, s_{16}$ are not independent; they are derived by applying iterative transformations to the 56-bit *master key* of DES.

This type of *key scheduling* process that injects partial information about the key at different rounds is commonly used in block cipher design. It possibly serves as a mechanism to hinder cryptanalysis by humans, as it makes the inner workings of the function challenging to understand. Generic constructions like the ones from Section 3, on the other hand, are designed so that human analysis is a desirable feature (for the objective is to come up with proofs of security).

Differential and linear cryptanalysis. One class of attacks that is natural to consider in the context of iterated constructions like the Feistel network is differential cryptanalysis [30]. The attacker tries to obtain

correlations among pairs of outputs by flipping some bit positions of a (random) input. If the correlations are sufficiently "typical" they may tend to propagate throughout the Feistel network and be used to learn the candidate PRF. Biham and Shamir [31] designed such an attack to learn DES using 2^{47} queries and a similar amount of time, and other iterated constructions even more efficiently.

Miles and Viola [101] suggest the following formalization of differential cryptanalysis. Here $+$ denotes bit-wise XOR.

Definition 10 (Differential cryptanalysis)**.** A family $\{F_s\colon \{0,1\}^k \to \{0,1\}^\ell\}$ of functions is said to be ε-*secure against differential cryptanalysis* if for all $\Delta x \neq 0$ and Δy, $\Pr_{x,s}[F_s(x) + F_s(x+\Delta x) = \Delta y] \leq \varepsilon$.

The two-round Feistel construction is an example that is insecure against differential cryptanalysis (recall (15) in Section 5.2).

Linear cryptanalysis [98] is a different type of attack that attempts to find linear relationships between the bits of random input–output pairs. Miles and Viola [101] formalize it as follows. Here $\langle a,x\rangle$ denotes the inner product modulo 2 function $a_1x_1 + \cdots + a_kx_k$.

Definition 11 (Linear cryptanalysis)**.** A family $\{F_s\colon \{0,1\}^k \to \{0,1\}^\ell\}$ of functions is said to be ε-*secure against linear cryptanalysis* if for all $a \in \{0,1\}^k$ and $b \in \{0,1\}^\ell$, $\mathrm{E}_s[\mathrm{E}_x[(-1)^{\langle a,x\rangle + \langle b,F_s(x)\rangle}]^2] \leq \varepsilon$.

Matsui [98] devised an attack of this type to learn DES using 2^{43} queries.

In spite of these attacks, DES is believed to be a remarkably secure PRP. Its 56-bit key, however, is considered inadequately short for modern applications.

Substitution–permutation networks. Substitution–permutation networks (SPNs) are another blueprint for constructing PRPs. Here the PRP $P_s\colon \{0,1\}^k \to \{0,1\}^k$ is obtained by sequentially composing "simple" permutations of the following types:

- **S-boxes** are highly nonlinear fixed (independent of the key) permutations $S\colon \{0,1\}^c \to \{0,1\}^c$ where c is a small factor of k. The input is partitioned into k/c blocks of size c, and the S-box is applied to each block in parallel.
- **P-boxes** are linear permutations of the whole input that contain a large number of input–output dependencies.
- **Round key** operations are linear shifts of the input by the round key. As in Feistel network-based constructions, round keys are obtained by applying iterative transformations to the master key.

The nonlinear nature of the S-boxes is meant to guarantee security against "local" attacks such as linear and differential cryptanalysis. The P-boxes ensure that the effect propagates throughout the input positions. Miles and Viola prove general bounds on the security of SPNs against linear and differential cryptanalysis.

AES (the Advanced Encryption Standard) is a highly efficient SPN-based family of permutations on 128 bits. There are three variants of the construction, allowing keys of size 128, 192, and 256, respectively. In spite of the scrutiny this design has received, no significant weakness is known.

Miles and Viola propose several SPN-based constructions of PRP and PRF families on infinitely many input lengths and provide some theoretical evidence for their asymptotic security. If their security conjectures hold, quasilinear-size circuit families, TC^0-type circuit families of size $n^{1+\varepsilon}$, and quadratic-time single-tape Turing machines with a quadratic number of states are hard to learn and have no natural property.

Dodis et al. [54] study a model of SPNs in which the S-boxes are implemented by a random permutation oracle that can be queried both by the construction and by the distinguisher. They observe that 2-round SPNs are insecure in this model and construct a 3-round SPN that is provably $O(n^2q^2/2^c)$-secure against a distinguisher that makes q queries. They also obtain similar security for a 1-round SPN variant with a nonlinear P-box.

Weak pseudorandom functions. Owing to the severely restricted nature of the distinguisher, weak PRFs ought to be easier to construct than their strong counterparts. Differential cryptanalysis, in particular, does not apply to weak PRFs, although linear cryptanalysis does. We describe two conjectured separations between PRFs and weak PRFs as evidence that weak PRFs are indeed a less complex object.

Blum, Furst, Kearns, and Lipton [36] (Section 2.3) consider the following family of functions $F_{S,T}\colon \{0,1\}^n \to \{0,1\}$:

$$F_{S,T}(x) = \text{MAJORITY}(x|_S) + \text{PARITY}(x|_T),$$

where the key consists of two random $(\log n)$-bit subsets S and T of $\{1,\ldots,n\}$, $x|_S$ is the projection of x on its S-coordinates, and $+$ denotes XOR. They conjecture that this family is a (n^c, n^{-c})-weak PRF for any constant c and sufficiently large k. The best known algorithms for learning such functions from random examples have complexity $n^{\Omega(\log n)}$ [105]. For every S and T, $F_{S,T}$ is a function of $2\log n$ inputs and can in particular be computed by a DNF of size n^2. In contrast, as discussed in Section 6.1, DNFs cannot compute strong PRFs whose security is superpolynomial in their size.

In the regime of exponential security, the binary modulus ($q = 2$) variant of LWE (see Section 4.2) is a conjectured example of a *randomized* weak PRF. The noisy parity randomized function family $F_s\colon \{0,1\}^n \to \{0,1\}$ ($s \in \{0,1\}^n$) is given by

$$F_s(x) = \langle s, x\rangle + e(x) = s_1x_1 + \cdots + s_nx_n + e(x),$$

where the bits $e(x)\colon x \in \{0,1\}^n$ are independent and δ-biased for some $\delta < 1/2$. Blum, Kalai, and Wasserman [37] give an algorithm for this problem with running time $2^{O(n/(\log n - \log\log 1/(1-2\delta)))}$. Lyubashevsky [95] significantly reduces the query complexity of this algorithm but at the cost of increasing its running time.

Noisy parities are attractive owing to their extreme simplicity, but their randomized nature is undesirable. For instance, using F_s as the basis of an encryption scheme like the one in Section 2.4 would introduce decryption errors with some probability.

Akavia et al. [3] conjecture that the function family $G_{A,b}\colon \{0,1\}^n \to \{0,1\}$ given by

$$G_{A,b}(x) = g(Ax + b) \tag{16}$$

is a weak PRF. Here, A is a random $n \times n$ matrix, b is a random $\{0,1\}^n$ vector, and $g\colon \{0,1\}^n \to \{0,1\}$ is a suitably chosen function of constant depth and size polynomial in n. By the discussion in Section 6.2, the security of any strong PRF in this class is at most quasipolynomial.

Akavia et al. propose setting g to equal the tribes function of Ben-Or and Linial [27] XORed with an additional input bit. In Section 7.5 we show that this instantiation is insecure. Proposing an explicit choice of g for which the family (16) is arguably weakly pseudorandom, or showing that no such choice exists, remains an open problem. ⍰

7 Distinguishers

There is no intuition.
You just do the calculation.
JOHAN HÅSTAD (1990s)

The security of PRFs is required to hold against arbitrary efficient adversaries. It is however sometimes useful to study distinguishers of restricted computational power. Restricted distinguishers can model specific classes of cryptanalytic attacks. Proofs of security against such distinguishers can provide evidence for the soundness of heuristic constructions, and potentially achieve better parameters even when a generic proof of security is available. Moreover, if the class of distinguishers is sufficiently restrictive, unconditional proofs of security may be possible.

In the first two parts we focus on distinguishers of bounded query complexity and linear distinguishers. These lead to natural requirements on the distribution of outputs of the PRF: bounded independence for the first type and small bias for the second. We then turn to space-bounded distinguishers and several types of "randomness tests" that have found application in computational complexity and learning theory: polynomial correlation tests, rational function representations, cylinder product tests, and statistical query algorithms.

7.1 Distinguishers of Bounded Query Complexity

The following is a generalization of pairwise independence (Definition 7 in Section 5.1).

Definition 12 (Bounded independence)**.** A function family $F_s\colon \{0,1\}^k \to \{0,1\}^\ell$ is *q-wise independent* if it is $(\infty, q, 0)$-pseudorandom.

Pairwise (2-wise) indepedence guarantees security against linear and differential cryptanalysis in the sense of Definitions 10 and 11.

Achieving q-wise independence requires a key of length at least $q\ell$. When $\ell = n$, a key of size $q\ell$ is in fact sufficient: the function $F_s(x) = s_0 + s_1 x + \cdots + s_{t-1}x^{t-1}$, where s is the vector $(s_0, \ldots, s_{t-1}) \in \mathbb{F}_{2^k}^t$ and all algebra is over the field $\mathbb{F}_{2^k}$.[9]

Even though the function F_s is perfectly indistinguishable from a random function by a q-query distinguisher, it is not even weakly pseudorandom: since F_s is a polynomial of degree q, after observing $F_s(x_1), \ldots, F_s(x_q)$ at any q points, the value $F_s(x)$ can be computed efficiently for all x using the Lagrange interpolation formula. Therefore, F_s can be distinguished from a random function using any $q+1$ queries.

Bounded independence tends to be effective against adversaries that do not employ attacks based on linear algebra (over various outputs of the PRF). One model for such adversaries is the class of bounded-depth circuits. Circuits in this class cannot compute linear functions unless they are very large [58, 138, 73]. Bounded independence ensures security against bounded-depth distinguishers: a $(\log s)^{O(d)} \cdot \log(1/\varepsilon)$-wise independent function family is (s, ε)-pseudorandom with respect to distinguishers of size s and depth d [47] (see also [133, 72]).

The notions of bounded independence and cryptographic pseudorandomness are incomparable: not only is bounded independence insufficient for cryptographic pseudorandomness, it is also unnecessary. Even for a single bit of output, bounded independence requires perfect indistinguishability from true randomness.

[9] For one bit of output, a key size of $(q \log k)/2 + O_q(1)$ is both necessary and sufficient (see Section 13.2 of [4]).

For cryptographic purposes statistical indistinguishability is adequate. This leads to an approximate notion of bounded independence.

Definition 13 (Approximate bounded independence)**.** $\{F_s\}$ is (q,ε)-wise independent if it is (∞,q,ε)-pseudorandom with respect to nonadaptive distinguishers.

Approximate bounded independence is closely related to the small bias property that we discuss next.

7.2 Linear Distinguishers

A linear distinguisher computes some linear function of the values of the PRF. A distribution that is pseudorandom against such distinguishers is called small-biased [106]. We focus on the case of linear tests over the group $\mathbb{Z}_2$, where it is natural to assume that the PRF is Boolean-valued. The definition can be extended to other Abelian groups.

Definition 14 (Small bias [106])**.** A function family $\{F_s\}$ is (q,ε)-*biased* if every distinguisher that computes a linear function modulo 2 of at most q bits of the function's output has advantage at most $\varepsilon/2$.

It is convenient to view the range of the function as a multiplicative group, which allows for easier Fourier analysis [116]. Under this convention, the values of F_s are represented by the square roots of unity 1 and -1. The small bias property then requires that for all distinct inputs $x_1,\ldots,x_r$, $1 \le r \le q$,

$$\left|\mathrm{E}[F_s(x_1)\cdots F_s(x_r)]\right| \le \varepsilon.$$

If a function family is $(q,\varepsilon/2)$-wise independent then it is clearly (q,ε)-biased. The two definitions are in fact equivalent up to an exponential loss in q:

Lemma 3. *If $\{F_s\}$ is (q,ε)-biased then it is $(q,\sqrt{2^q-1}\cdot\varepsilon/2)$-wise independent.*

It follows from Lemma 3 that $(q,2^{-q/2})$-wise independence cannot be achieved when q exceeds the key length. For smaller values of q, the small bias property provides information-theoretic security against distinguishers that make a bounded number of queries. In particular, these include differential attacks.

Proof. Let $D\colon \{0,1\}^X \to \{-1,1\}$ be any statistical distinguisher that queries F on the set $X = \{x_1,\ldots,x_q\}$. In the Fourier basis we can write

$$D^F = \sum_{A\subseteq X} \hat{D}(A)\cdot\prod_{x\in A} F(x).$$

Then

$$\begin{aligned}
\left|\mathrm{E}[D^{F_s}] - \mathrm{E}[D^R]\right| &= \left|\textstyle\sum_{A\subseteq X}\hat{D}(A)\cdot\left(\mathrm{E}\prod_{x\in A}F_s(x) - \mathrm{E}\prod_{x\in A}R(x)\right)\right| \\
&= \left|\textstyle\sum_{A\subseteq X, A\neq\varnothing}\hat{D}(A)\cdot\mathrm{E}\prod_{x\in A}F_s(x)\right| \\
&\le \textstyle\sum_{A\subseteq X, A\neq\varnothing}|\hat{D}(A)|\cdot\left|\mathrm{E}\prod_{x\in A}F_s(x)\right| \\
&\le \varepsilon\cdot\textstyle\sum_{A\subseteq X, A\neq\varnothing}|\hat{D}(A)|.
\end{aligned} \tag{17}$$

By the Cauchy–Schwarz inequality and Parseval's identity the last expression is at most $\varepsilon\sqrt{2^q-1}$. Therefore

$$\left|\Pr[D^{F_s}\text{ accepts}] - \Pr[D^R\text{ accepts}]\right| = \tfrac{1}{2}\cdot\left|\mathrm{E}[D^{F_s}] - \mathrm{E}[D^R]\right| \le \sqrt{2^q-1}\cdot\varepsilon/2.$$

□

Generalizing this argument to adaptive distinguishers, it follows that if $\{F_s\}$ is (q,ε)-biased then every adaptive distinguisher that makes at most q queries has advantage at most $2^q \cdot \varepsilon/2$. To prove this, the distinguisher is modeled as a decision tree over variables $F_s(x)$ of depth at most q. The sum of the absolute values of the Fourier coefficients of a decision tree is upper bounded by the number of its leaves, which is at most 2^q. The bound then follows by a calculation similar to (17).

There are several efficient constructions of $(2^k,\varepsilon)$-biased families $F_s\colon \{0,1\}^k \to \{0,1\}$ of size polynomial in k and $\log 1/\varepsilon$ [106, 5, 26]. As in the case of bounded independence, some of them are cryptographically insecure.

We do not know of any generic methods for obtaining cryptographically secure PRFs that provably exhibit the small bias property. In the case of the GGM construction, it would be interesting to understand which property of the underlying PRG is sufficient to guarantee that the PRF has small bias. As an initial step in this direction, we suggest the problem of constructing an efficient PRG G so that the GGM construction instantiated with G has small bias. ⓘ

Regarding concrete constructions, Miles and Viola [101] prove that one of their proposed PRFs is $(3,2^{-\Omega(k)})$-biased. It would be interesting to prove that the other constructions described in Section 4 have the small bias property. ⓘ

7.3 Space-Bounded Distinguishers

A distinguisher is space-bounded if its amount of memory is bounded by some (possibly sublinear) efficient function of the PRF key size. An algorithm with m bits of memory can be modeled as a branching program of width 2^m whose input is the truth table of the function to be distinguished.

Definition 15 (Pseudorandomness against bounded space)**.** A function family $\{F_s\colon \{0,1\}^k \to \{0,1\}\}$ is ε-pseudorandom against space m if the distinguishing advantage of any branching program of width 2^m is at most ε.

The access mode of the branching programs can be sequential, random, oblivious, or unrestricted, corresponding to the notion of sequential, weak, nonadaptive, and general PRFs, respectively.

The pseudorandom generators of Nisan [114] and Impagliazzo, Nisan, and Wigderson [78] can be viewed as function families of key length $O(k^2+km+k\log(1/\varepsilon))$ and size polynomial in k, m, and $1/\varepsilon$ that are ε-pseudorandom against space m with sequential access.

In a permutation branching program, the answer to each query induces a permutation of the states. Reingold, Steinke, and Vadhan [127] give a PRF of key length and size polynomial in k, 2^m, and $\log 1/\varepsilon$ that is ε-pseudorandom against space m permutation branching programs with nonadaptive *read-once* access. There is little hope of removing the read-once and permutation restrictions at the same time: By Barrington's theorem [20], polynomial-size branching program families of width 5 have the same computational power as the circuit class NC^1. A PRF against NC^1 with an unconditional proof of security would imply an explicit circuit lower bound against this class, thereby resolving a long-standing open problem in computational complexity.

Raz [121] proved that the function $F_s(x) = \langle s,x\rangle = s_1x_1 + \cdots + s_nx_n \bmod 2$ is $2^{-o(n)}$-weakly pseudorandom against branching programs of width $2^{o(n^2)}$ and length $2^{o(n)}$. This result was generalized to a larger class of functions [122] (see also Moshkovitz and Moshkovitz [104]).

7.4 Correlation with Polynomials

I don't like polynomials.
They are mysterious.
ODED GOLDREICH (2000s)

One can attempt to detect nonrandom behavior in a function by looking for correlations with some structured class of functions. In this context, the class of low-degree polynomials over a finite field is important in many areas of the theory of computing. On the algorithmic side, there are efficient methods for detecting low-degree correlations in several interesting parameter regimes. On the complexity-theoretic side, functions that correlate with some low-degree polynomial capture several interesting classes of computations, in particular bounded-depth circuits with AND, OR, and PARITY gates [123, 132].

Here we focus on polynomials over the binary field $\mathbb{F}_2$. The results can be extended to other finite fields, but the efficiency of the tests worsens as the field size becomes larger.

Definition 16 (Proximity to polynomials)**.** The function $F\colon \mathbb{F}_2^k \to \mathbb{F}_2$ is δ-*close to degree* d if

$$\Pr_{x \leftarrow \mathbb{F}_2^k}[F(x) \neq p(x)] \leq \delta$$

for some polynomial $p\colon \mathbb{F}_2^k \to \mathbb{F}_2$ of degree at most d.

A standard counting argument shows, for example, that the probability of a random function being $1/3$-close to degree $k/3$ is at most $2^{-\Omega(2^k)}$. We now describe some settings in which low-degree correlation can be tested efficiently.

Exact representation. In the extreme setting $\delta = 0$, we are interested in an exact representation of F as an $\mathbb{F}_2$-polynomial of degree at most d:

$$F(x) = \sum_{S\colon |S| \leq d} \tilde{F}(S) \cdot \prod_{i \in S} x_i. \tag{18}$$

Every input–output pair $(x, F(x))$ then reveals a linear dependence between the coefficients $\tilde{F}(S)$. Given sufficiently many such linear dependences, the values $\tilde{F}(S)$ can be learned via linear algebra. As there are $\binom{k}{\leq d}$ values to be learned, at least this many queries to F are required. This number of queries is also sufficient: For example, if x ranges over all $\binom{k}{\leq d}$ strings of Hamming weight at most d, then the system of equations (18) has full row rank and the coefficients $\tilde{F}(S)$ are uniquely determined.

Using a few additional queries, F can even be learned from independent random samples of the form $(x, F(x))$: $O(2^d \binom{k}{\leq d})$ such pairs are sufficient to ensure full row rank with constant probability. In Section 7.5 we will analyze a more general variant of this test. Since every function has a unique canonical multilinear expansion, it can in principle be verified that a given candidate PRF implementation resists these types of attacks.

High correlation. Low-degree polynomials have a dual characterization: A function p has degree at most d if and only if $\sum_{x \in A} p(x) = 0$ for all affine subspaces A of dimension $d+1$.

This characterization provides a more efficient test for exact representation of F by a polynomial of degree at most d. Moreover, it extends to testing correlation in the regime where δ is smaller than 2^{-d-2}.

The test chooses a random $a \leftarrow \mathbb{F}_2^k$ and outputs $\sum_{x \in A+a} F(x)$ for an arbitrary $(d+1)$-dimensional affine subspace A. Assuming F is δ-close to a polynomial p of degree at most d, it follows from union bounds that

$$\begin{aligned}\Pr\Big[\sum_{x \in A+a} F(x) \neq 0\Big] &\leq \Pr\Big[\sum_{x \in A+a} p(x) \neq 0\Big] + \Pr[F(x) \neq p(x) \text{ for some } x \in A+a] \\ &\leq 0 + \sum_{x \in A} \Pr[F(x+a) \neq p(x+a)] \\ &= \delta \cdot 2^{-d-1},\end{aligned}$$

while $\Pr[\sum_{x \in A+a} R(x) \neq 0] = 1/2$ for a random function R.

The learner for exact degree-d representation from random input–output samples can also be used in a regime of very high correlation: If F is $\delta = 1/O(2^d \binom{k}{\leq d})$-close to degree d, the learner outputs the unique polynomial p that is δ-close to F with constant probability.

Noticeable correlation. In the regime $\delta \geq 2^{-d-1}$ there may be more than one polynomial of degree at most d that is δ-close to F. This imposes additional difficulties in the design and analysis of correlation tests.

The Gowers test. A natural way to test for correlation is to evaluate the expression $\sum_{x \in A} p(x)$ on a *random* affine subspace A of dimension $d+1$. It is more convenient for the analysis to also allow degenerate subspaces (of lower dimension): The distinguisher chooses $a_0, \ldots, a_{d+1}$ independently and uniformly at random from $\mathbb{F}_2^k$, sets $A = \{a_0 + a_1 x_1 + \cdots + a_{d+1} x_{d+1} \colon x_1, \ldots, x_{d+1} \in \mathbb{F}_2\}$, and outputs $\sum_{x \in A} p(x)$.

Theorem 8. *The Gowers test distinguishes functions that are δ-close to degree d from a random function with advantage at least $2 \cdot (2\delta - 1)^{2^{d+1}}$.*

Proof. Gowers [70] showed that if F is δ-close to degree d then

$$\mathop{\mathrm{E}}_{A}\Big[\prod_{x \in A} (-1)^{F(x)}\Big] \geq (2\delta - 1)^{2^{d+1}}.$$

On the other hand, $\mathrm{E}_A[\prod_{x \in A} (-1)^{R(x)}] = 0$ for a random function R. Therefore,

$$\begin{aligned}&\Pr\Big[\sum_{x \in A} F(x) = 0\Big] - \Pr\Big[\sum_{x \in A} R(x) = 0\Big] \\ &\qquad = 2 \cdot \Big(\mathrm{E}\Big[\prod_{x \in A} (-1)^{F(x)}\Big] - \mathrm{E}\Big[\prod_{x \in A} (-1)^{R(x)}\Big]\Big) \geq 2 \cdot (2\delta - 1)^{2^{d+1}}.\end{aligned}$$

□

The analysis is essentially tight, as can be seen by instantiating F to a random function with $\delta 2^k$ ones. Owing to the doubly exponential dependence on d in Theorem 8, the efficiency of the Gowers test degrades rapidly with the degree.

The cylinder product test described in Section 7.6 extends the Gowers test to a larger class of functions. The two have identical soundness guarantees.

In Section 7.5 we describe a test of Razborov and Rudich [124] for correlation with rational functions, of which polynomials are a special case. The Razborov–Rudich and Gowers tests have incomparable soundness guarantees.

Weak pseudorandomness. If only random input–output samples are available, the problem of detecting correlation with linear functions (i.e., degree-1 polynomials) is polynomially equivalent to learning noisy parities [55]. For this purpose, the algorithms of Blum, Kalai, and Wasserman and Lyubashevsky discussed in Section 6.3 can be applied. We do not know of any results for higher degree. ⑦

7.5 Correlation with Rational Functions

A rational function is a ratio of two polynomials with the convention that $0/0$ may represent any value.

Definition 17. A function $F\colon \mathbb{F}_2^k \to \mathbb{F}_2$ has *rational degree* at most r if there exist polynomials p and q of degree at most r, not both identically zero, such that

$$F(x)\cdot q(x) = p(x) \qquad \text{for all } x \text{ in } \mathbb{F}_2^k. \tag{19}$$

Rational degree generalizes polynomial degree, which corresponds to the special case $q \equiv 1$. Representation by rational functions is in fact equivalent to one-sided representation by polynomials in the following sense:

Proposition 14 ([11])**.** *F has rational degree at most d if and only if there exists a nonzero polynomial P of degree at most d such that $P(x) = 0$ whenever $F(x) = b$ for some $b \in \{0,1\}$.*

As in the case of polynomials, we say f is δ-close to rational degree at most d if there exists r of rational degree at most d such that $\Pr[f(x) \neq r(x)] \leq \delta$.

We describe and analyze two tests for correlation with rational functions of low degree. The first one applies to the high-correlation regime and can distinguish even weakly pseudorandom functions. The second one, due to Razborov and Rudich [124], is for strong pseudorandom functions but works even when the proximity parameter δ is close to $1/2$.

Exact representation and high correlation. Functions of low rational degree cannot even be weakly pseudorandom:

Proposition 15. *If $F_s\colon \mathbb{F}_2^k \to \mathbb{F}_2$ has rational degree at most d for all s then it is not a $(\mathrm{poly}(2^d\binom{k}{\leq d}), \frac{3}{4})$-weak PRF.*

Proof. The test accepts if there exist p and q that are consistent with the equations (19) on $m = 4\cdot 2^d\left(\binom{k}{\leq d}+1\right)$ samples x. This is a linear system in the coefficients of p and q, so the existence of a nonzero solution can be decided by a circuit of size polynomial in m. If $F = F_s$, a nonzero solution always exists. If, on the other hand, F is a random function, the following claim applies:

Claim 21. *If p, q are polynomials of degree at most d, not both zero, then for a random function R and a random input x, $\Pr_{R,x}[R(x)\cdot q(x) = p(x)] \leq 1 - 2^{-d-1}$.*

Proof. Since p and q are not both zero, it must be that p is nonzero or p and q are different. If p is nonzero,

$$\Pr[R(x)\cdot q(x) \neq p(x)] \geq \Pr[p(x)\neq 0]\cdot \Pr[R(x) = 0 \mid p(x)\neq 0] \geq 2^{-d}\cdot\tfrac{1}{2} = 2^{-d-1},$$

where the inequality $\Pr[p(x)\neq 0] \geq 2^{-d}$ follows by the Schwarz–Zippel lemma. If p and q are different, then $\Pr[R(x)\cdot q(x) \neq p(x)] = \Pr[(R(x)+1)\cdot q(x) \neq p(x)+q(x)]$, and the same argument applies to the functions $R+1$, q, and $q+p$. □

By independence and a union bound over all pairs (p,q), the probability that a random function passes the test is at most

$$\left(2^{2\binom{k}{\leq d}} - 1\right)\cdot\left(1-2^{-d-1}\right)^m \leq 2^{2\binom{k}{\leq d} - m\cdot 2^{-d-1}} \leq \tfrac{1}{4}$$

by the choice of m. □

Akavia et al. [3] conjecture that the construction (16) instantiated with g equal to the tribes function XORed with an additional input bit is a weak PRF. Their conjecture is false as g can be seen to have rational degree at most $O(\log k)$. By Proposition 15, the resulting function family is not a $(k^{O(\log k)}, \frac{1}{3})$-weak PRF.

The following theorem shows, more generally, that weak PRF constructions in the class $\mathrm{DNF} \circ \oplus$ cannot be too secure.

Theorem 9. *If $F_s \colon \{0,1\}^k \to \{0,1\}$ is an OR of at most t ANDs of parities of literals for all s then it is not a $(\mathrm{poly}(tk \cdot 2^{\log t \cdot \log k}), \frac{1}{3})$-weak PRF.*

sketch. The distinguisher accepts if (1) F passes the test in Proposition 15 with $d = \log_2 t + 2$ or (2) the number of x such that $F(x) = 0$ is at most $2^k/3$. From the above proof and large deviation bounds, it follows that a random function is accepted with probability at most $1/3$.

When $F = F_s$, we consider two possibilities. If all AND terms of F_s have fan-in more than $\log_2 t + 2$, then by a union bound, $F_s(x)$ is nonzero with probability at most $1/4$ over the choice of x. By a large deviation bound, F_s is then rejected by test (2) with probability at most $1/3$. If, on the other hand, F_s contains an AND term with fan-in at most $\log_2 t + 2$, then this term is a nonzero polynomial of degree at most $\log_2 t + 2$ that evaluates to one whenever F_s does. By Proposition 14, F_s has rational degree $\log_2 t + 2$ and it is rejected by test (1) with probability one. □

Proposition 15 in fact holds under the weaker assumption that F_s is $o\big(1/2^d \binom{k}{\leq d}\big)$-close to rational degree d, as the probability that such a function triggers a false negative in the test is vanishingly small. Towards understanding the security of (16) it would be interesting to investigate the approximate rational degree of AC^0 function families. Akavia et al. show that the tribes function on n inputs is $\Omega(2^d)$-far from polynomial degree d for every $d \leq n - \omega(\log n)$. Does a similar property hold for some AC^0 function family with respect to rational degree? ⍰

Noticeable correlation. Testing for correlation with functions of low rational degree can be reduced to testing for exact representation by a rational function of degree close to $k/2$.

Proposition 16. *If F is δ-close to rational degree at most r then the linear space of solutions to* (19) *has dimension at least $\binom{k}{\leq d-r} - \delta \cdot 2^k$.*

Proof. If F is δ-close to rational degree r, then there exist polynomials p, q of degree at most r such that $F(x) \cdot q(x) \neq p(x)$ for at most $\delta 2^k$ inputs $x \in \mathbb{F}_2^k$. Consider the linear space Z of polynomials of degree at most $d - r$ that vanish on these inputs. This space has dimension at least $\binom{k}{\leq d-r} - \delta \cdot 2^k$. On the other hand, for every $z \in Z$, it holds that $F(x) \cdot p(x) z(x) = q(x) z(x)$ on all inputs $x \in \mathbb{F}_2^k$, so all pairs of the form $(pz, qz) \colon z \in Z$ are solutions to (19). □

We now restrict our attention to the regime $d = (k - o(\sqrt{k}))/2$. Proposition 16 then has the following asymptotic behavior for every $\delta < 1/2$ and sufficiently large k: If $r = o(\sqrt{k})$ then the dimension of the solution space is at least $(\frac{1}{2} - \delta - o(1)) \cdot 2^k$. In particular, if δ is bounded away from $\frac{1}{2}$ then the system has at least one solution. This gives the following reduction from approximate to exact rational degree:

Corollary 1. *For every $\delta < 1/2$ there exists an $\varepsilon > 0$ such that for sufficiently large k, if $F \colon \mathbb{F}_2^k \to \mathbb{F}_2$ is δ-close to rational degree at most $\varepsilon\sqrt{k}$, then F has rational degree at most $(k - \varepsilon\sqrt{k})/2$.*

On the other hand, (19) is a linear system of 2^k equations in $(1 - o(1)) \cdot 2^k$ unknowns. We may conjecture that, for a reasonable fraction of functions F, the equations should exhibit few linear dependencies and

so the system should have no solution. Razborov and Rudich write that they have no easy proof of this conjecture. It was recently observed by Swastik Kopparty and the first author that this property follows from the asymptotic optimality of Reed-Muller codes under random erasures, which was recently proved by Kudekar et al. [88].

Proposition 17. *For every $\varepsilon > 0$, the probability that a random function $R\colon \mathbb{F}_2^k \to \mathbb{F}_2$ has rational degree at most $(k - \varepsilon\sqrt{k})/2$ approaches zero for large k.*

Proof. Kudekar et al. show that for every $\varepsilon > 0$, if E is a uniformly random subset of $\mathbb{F}_2^k$ (representing a set of erasures), every polynomial P of degree at most $d = (k - \varepsilon\sqrt{k})/2$ is uniquely determined by the evaluations $(x, P(x))\colon x \notin E$ with probability approaching one for large k. In particular, when $P \equiv 0$, the only polynomial of degree at most d whose zeros cover the set $\overline{E}$ is the zero polynomial with probability approaching one.

For a random function R, the sets $R^{-1}(0)$ and $R^{-1}(1)$ are uniformly random subsets of $\mathbb{F}_2^k$. Therefore the probability that there exists a nonzero polynomial of degree at most d whose zeros cover $R^{-1}(0)$ or $R^{-1}(1)$ approaches zero for large k. By Proposition 14, this is exactly the probability that R has rational degree at most d. □

We obtain the following consequence regarding the correlation of pseudorandom functions to functions of low rational degree.

Theorem 10. *For every $\delta < \frac{1}{2}$ there exists $\varepsilon > 0$ such that for sufficiently large k, if $F_s\colon \mathbb{F}_2^k \to \mathbb{F}_2$ is δ-close to rational degree at most $(k - \varepsilon\sqrt{k})/2$ for all s, then $\{F_s\}$ is not a $(2^{O(k)}, \frac{1}{2})$-PRF.*

Proof. By Corollary 1 and Proposition 17, the probability that a random function $R\colon \mathbb{F}_2^k \to \mathbb{F}_2$ is δ-close to rational degree at most $(k - \varepsilon\sqrt{k})/2$ is at most $\frac{1}{2}$ for sufficiently large k. Correlation with rational functions is testable in time $2^{O(k)}$ (the time it takes to solve (19)). It follows that F_s and R can be distinguished in size $2^{O(k)}$ with advantage $\frac{1}{2}$. □

Razborov and Rudich prove a weaker version of Proposition 17. They show that for k odd and $d = (k-1)/2$ the linear space of solutions to (19) has dimension at most $\frac{1}{4}2^k$ with probability at least $\frac{1}{2}$ over the choice of a random R.[10] Together with Proposition 16, this establishes the conclusion of Theorem 10 under the stronger assumption $\delta < \frac{1}{4}$.

For completeness, we give an elementary proof of an even weaker dimension bound, which yields the conclusion of Theorem 10 under the assumption $\delta < \frac{1}{2}\log_2 \frac{4}{3} \approx 0.2075$.

Proposition 18. *For k odd and $d = (k-1)/2$ the linear space (p, q) of solutions to* (19) *has dimension at most $\frac{1}{2}\log_2 \frac{3}{2} \cdot 2^k + 1$ with probability at least $\frac{1}{2}$ over the choice of a random F.*

Proof. For a fixed pair q, p and a random F, the probability that Fq equals r is zero if $q(x) = 0$ and $p(x) = 1$ for any x, and $2^{-|\{x\colon q(x)=1\}|}$ if not. If p and q are chosen independently at random, we have

$$\Pr_{F,p,q}[Fq = p] = \mathop{\mathrm{E}}_{p,q}\left[\mathbf{1}(q(x) = 1 \text{ or } p(x) = 0 \text{ for all } x) \cdot 2^{-|\{x\colon q(x)=1\}|}\right].$$

Let $B = \{x\colon |x| < k/2\}$ and $\overline{B} = \{x\colon |x| > k/2\}$. By the Cauchy–Schwarz inequality, we can write

$$\Pr[Fq = r] \le \sqrt{\mathrm{E}_{q,r}[Z(B)]} \cdot \sqrt{\mathrm{E}_{q,r}[Z(\overline{B})]},$$

[10] In fact, they only show this holds for the q-component of the solution space, which is sufficient.

where

$$Z(S) = \mathbf{1}(q(x) = 1 \text{ or } p(x) = 0 \text{ for all } x \in S) \cdot 2^{-2|\{x \in S:\, q(x)=1\}|}.$$

By symmetry, $\mathrm{E}[Z(B)] = \mathrm{E}[Z(\overline{B})]$, so $\Pr[Fq = p] \le \mathrm{E}[Z(B)]$.

Every polynomial of degree less than $k/2$ represents a unique function from B to $\mathbb{F}_2$. (The coefficients of an interpolating polynomial for a given function can be calculated iteratively in order of increasing set size. Since the dimensions of the space of functions and the space of polynomials are both 2^{k-1}, the correspondence is one-to-one.) Therefore the values of q and r at different points in B are mutually independent and

$$\mathrm{E}[Z(B)] = \prod_{x \in B} \mathrm{E}[Z(\{x\})].$$

The value of $Z(\{x\})$ is $1/2$ conditioned on $q(x) = 0$ and $1/4$ conditioned on $q(x) = 1$, from where $\mathrm{E}[Z(\{x\})] = 3/8$ and $\mathrm{E}[Z(B)] = (3/8)^{2^{k-1}}$. It follows that the expected number of solutions (q, r) to $Fq = r$ is at most

$$2^{2^k} \cdot \Pr_{F,q,r}[Fq = r] \le 2^{2^k} \cdot \left(\frac{3}{8}\right)^{2^{k-1}} = 2^{\frac{1}{2}\log_2 \frac{3}{2} \cdot 2^k}.$$

By Markov's inequality, the probability that the number of solutions is more than $2^{\frac{1}{2}\log_2 \frac{3}{2} \cdot 2^k + 1}$ is less than half, implying the bound on the dimension of the solution space. □

7.6 Correlation with Cylinder Products

Cylinder products are functions that exhibit a "product structure" with respect to a fixed partition of the inputs. They play a central role in the study of low-complexity circuits and communication protocols. Pseudorandom synthesizers and their higher-dimensional analogues turn out to be closely related to cylinder products.

Definition 18 (Cylinder product)**.** Let $D_1, \ldots, D_d$ be any finite domains. A function $F : D_1 \times \cdots \times D_d \to \{-1, 1\}$ is a *d-cylinder product*[11] if it can be written as a product $F = f_1 \cdots f_d$ where the function f_i does not depend on its i-th input.

In particular, 2-cylinder products are the functions of the form $F(x, y) = f(x) \cdot g(y)$. It is easily verified that such functions satisfy the relation

$$F(x_1, y_1) \cdot F(x_2, y_1) \cdot F(x_1, y_2) \cdot F(x_2, y_2) = 1$$

for all $x_1, x_2 \in D_1$ and $y_1, y_2 \in D_2$. In general, for a d-cylinder product F, the following expression vanishes for all $x_1, y_1 \in D_1, \ldots, x_d, y_d \in D_d$:

$$\prod_{\sigma_1 \in \{x_1, y_1\}, \ldots, \sigma_d \in \{x_d, y_d\}} F(\sigma_1, \ldots, \sigma_d). \tag{20}$$

The cylinder product test [52, 120, 136] evaluates (20) on uniformly random inputs $x_1, y_1, \ldots, x_d, y_d$. We next give an analogue of Theorem 8 for cylinder intersections.

Theorem 11. *The cylinder product test distinguishes functions that are δ-close to a d-cylinder from a random function with advantage at least $2 \cdot (2\delta - 1)^{2^d}$.*

[11] This is the class Π_d^* of Viola and Wigderson [136] (Section 3.1), who explain the close relation with the cylinder intersections of Babai, Nisan, and Szegedy [14].

The cylinder product test is more general than the Gowers test for the purpose of distinguishing from a random function: A degree-$(d-1)$ polynomial from $\mathbb{F}_2^k$ to $\mathbb{F}_2$ is a d-cylinder product with respect to any product partition $\mathbb{F}_2^{k_1} \times \cdots \times \mathbb{F}_2^{k_d}$ of the input domain (with $k_1, \ldots, k_d > 0$).

The cylinder product test generalizes to functions that take values on the complex unit circle. (This requires conjugating the entries with an odd number of x_i in (20).)

On the complexity of synthesizers. As the 2-cylinder product test can be implemented by a two-query distinguisher for a pseudorandom synthesizer, Theorem 11 has the following corollary:

Corollary 2. *If F if $\frac{1}{2}(1-(\varepsilon/2)^{1/4})$-close to some 2-cylinder then F is not a $(O(1),2,\varepsilon)$-pseudorandom synthesizer.*

As 2-cylinders over $\mathbb{Z}_2^n \times \mathbb{Z}_2^n$ include all linear functions, it follows that all Fourier coefficients of a synthesizer must have negligible magnitude.

Akavia et al. [3] provide evidence that all function families in the class $\mathrm{AC}^0 \circ \oplus$ of polynomial-size, constant-depth AND/OR circuits with a bottom layer of PARITY gates have a Fourier coefficient of magnitude at least $\exp(-\mathrm{poly}\log n)$. Thus, it is conceivable that $\mathrm{AC}^0 \circ \oplus$ circuits can compute weak PRFs (see Section 6.3) but not synthesizers. In contrast, Proposition 5 indicates weak PRFs can only be *more* complex than synthesizers.

To resolve this apparent contradiction, recall that the complexity of a weak PRF family $F_s(x)$ is the maximum complexity of the function F_s over all fixings of the key s. The induced synthesizer $S(s,x) = F_s(x)$ is a function of both the key and the input of the original PRF. The function S could have high complexity, but reduce to a function of lower complexity once s is fixed.

This phenomenon is exemplified by the LWR problem from Section 4.2: For every fixing of the key $\mathbf{s}$, the function $F_{\mathbf{s}}(\mathbf{a}) = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_p$ (see (11)) has a large Fourier coefficient over $\mathbb{Z}_q^k$ (the coefficient $\hat{F}_{\mathbf{s}}(\mathbf{s})$), while all the Fourier coefficients of the corresponding synthesizer are small.

7.7 Statistical Queries

Random input–output pairs $(x, F(x))$ can be used as samples to estimate statistics $\mathrm{E}\,\phi = \mathrm{E}_x[\phi(x, F(x))]$ for various real-valued functions ϕ. If ϕ is bounded in the range $[-1,1]$, then about $1/\varepsilon^2$ samples are needed to estimate the statistic within error ε with constant probability. Many known algorithms for learning from random examples operate in this manner: The algorithm computes estimates of $\mathrm{E}\,\phi$ for all ϕ in some fixed class of real-valued functions Φ and outputs the value of some function of these estimates only. The algorithm of Linial, Mansour, and Nisan for low-weight Fourier learning is one such example.

The statistical query learning model [83] captures this class of algorithms. The following pseudorandomness property is necessary and sufficient for functions to be hard to learn in this model [49]. It postulates that any statistic of the queried function should be close in value to what is expected for a random function.

Definition 19 (Pseudorandomness against statistical queries)**.** The family $\{F_s\}$ is (ε, δ)-pseudorandom against statistical queries Φ if, with probability at least $1-\delta$ over the choice of s,

$$\left| \mathop{\mathrm{E}}_{x}[\phi(x, F_s(x))] - \mathop{\mathrm{E}}_{x,R}[\phi(x, R(x))] \right| \leq \varepsilon$$

for all ϕ in Φ.

For Boolean-valued functions, pairwise independence is sufficient to ensure pseudorandomness against statistical queries.

Lemma 4. *Let Φ be any set of $[-1,1]$-valued functions. If $F_s\colon \{0,1\}^k \to \{-1,1\}$ is pairwise independent then it is $(\sqrt{2|\Phi|/\delta 2^k}, \delta)$-pseudorandom against Φ.*

Proof. Any statistical query can be written as a combination of two *correlation queries* [49]:

$$\begin{aligned}\mathop{\mathrm{E}}_x[\phi(x,F(x))] &= \mathop{\mathrm{E}}_x\left[\phi(x,-1)\cdot\frac{1-f(x)}{2}+\phi(x,1)\cdot\frac{1+f(x)}{2}\right]\\ &= \tfrac{1}{2}\mathop{\mathrm{E}}_x[\phi(x,1)\cdot F(x)] - \tfrac{1}{2}\mathop{\mathrm{E}}_x[\phi(x,-1)\cdot F(x)] + \tfrac{1}{2}\mathop{\mathrm{E}}_x[\phi(x,1)+\phi(x,-1)].\end{aligned}$$

The last term is independent of F, so we can bound the distinguishing advantage of ϕ by

$$\begin{aligned}\bigl|\mathop{\mathrm{E}}_x[\phi(x,F_s(x))] - \mathop{\mathrm{E}}_{x,R}[\phi(x,R(x))]\bigr| &\le \tfrac{1}{2}\bigl|\mathop{\mathrm{E}}_x[\phi(x,1)\cdot F_s(x)] - \mathop{\mathrm{E}}_{x,R}[\phi(x,1)\cdot R(x)]\bigr|\\ &\quad + \tfrac{1}{2}\bigl|\mathop{\mathrm{E}}_x[\phi(x,-1)\cdot F_s(x)] - \mathop{\mathrm{E}}_{x,R}[\phi(x,-1)\cdot R(x)]\bigr|\\ &= \tfrac{1}{2}\bigl|\mathop{\mathrm{E}}_x[\phi(x,1)\cdot F_s(x)]\bigr| + \tfrac{1}{2}\bigl|\mathop{\mathrm{E}}_x[\phi(x,-1)\cdot F_s(x)\bigr|.\end{aligned}$$

It therefore suffices to bound $|\mathrm{E}_x[\psi(x)\cdot F_s(x)]|$ for an arbitrary set of $2|\Phi|$ functions $\psi\colon \{0,1\}^k \to [-1,1]$. We apply the second-moment method. Since every bit of F_s is uniformly distributed, $\mathrm{E}_s\,\mathrm{E}_x[\psi(x)\cdot F_s(x)]$ equals zero. For the second moment, write

$$\begin{aligned}\mathop{\mathrm{E}}_s\bigl[\mathop{\mathrm{E}}_x[\psi(x)\cdot F_s(x)]^2\bigr] &= \mathop{\mathrm{E}}_s\bigl[\mathop{\mathrm{E}}_{x,y}[\psi(x)F_s(x)\cdot\psi(y)F_s(y)]\bigr]\\ &= \mathop{\mathrm{E}}_{x,y}\bigl[\psi(x)\psi(y)\mathop{\mathrm{E}}_s[F_s(x)\cdot F_s(y)]\bigr]\\ &= \mathop{\mathrm{E}}_{x,y}[\psi(x)\psi(y)\mathbf{1}(x=y)]\\ &\le \Pr[x=y].\end{aligned}$$

This is the collision probability of the uniform distribution, which equals $1/2^k$. By Chebyshev's inequality, the probability that $|\mathrm{E}_s[\psi(x)\cdot F_s(x)]|$ exceeds $\sqrt{2|\Phi|/\delta 2^k}$ is at most $\delta/2|\Phi|$. The lemma follows by taking a union bound over all ψ. □

It would be interesting to investigate if an analogous statement holds for adaptive distinguishers. ⑦

A variant of this proof appears in the work of Akavia et al. [3]. In particular, since the function family $H_{A,b}(x) = Ax+b$ is pairwise independent, construction (16) is pseudorandom against statistical queries.

8 Contemporary Constructions

The dishwasher is a gift of nature!
SILVIO MICALI (1980's)

We present two recently-proposed extensions of PRFs and describe some of their applications. A *key-homomorphic* PRF allows for the efficient evaluation of $F_{s_1+s_2}(x)$ given the values $F_{s_1}(x)$ and $F_{s_2}(x)$. In a *puncturable* PRF the adversary obtains code for evaluating F_s everywhere but at a single input $\mathring{x}$ of its choice and cannot distinguish the value $F_s(\mathring{x})$ from a random one.

Puncturable PRFs are an indispensable tool in applications of *indistinguishability obfuscation*, an intriguing concept that has attracted recent interest. Sahai and Waters used puncturable PRFs together with indistinguishability obfuscation to convert certain private-key encryption schemes into public-key ones. We provide a self-contained treatment of their result.

8.1 Key-Homomorphic PRFs

Key-homomorphic PRFs were introduced by Naor, Pinkas, and Reingold [107], who constructed, in the random oracle model, a very simple key-homomorphic PRF family assuming the DDH problem is hard.

Definition 20 (Key-homomorphic PRF)**.** Let $\mathbb{S}$ and $\mathbb{G}$ be Abelian groups. We say that a family $\{F_s\colon \{0,1\}^k \to \mathbb{G}\}$ of functions, indexed by $s \in \mathbb{S}$, is *key homomorphic* if for every $s_1, s_2 \in \mathbb{S}$ and every $x \in \{0,1\}^k$, it holds that

$$F_{s_1}(x) + F_{s_2}(x) = F_{s_1+s_2}(x).$$

Recently, Boneh et al. [42] constructed the first (almost) key-homomorphic PRF without random oracles. The construction is based on the LWE problem, and builds upon ideas used in the non-key-homomorphic LWE-based PRFs of Banerjee, Peikert, and Rosen [17], and specifically on the reduction from LWE to LWR (Proposition 6). The Boneh et al. construction was subsequently generalized by Banerjee and Peikert [16], resulting in higher efficiency and tighter security reductions.

Constructions in the random oracle model. Let $\mathbb{G}$ be a finite cyclic group of prime order q and let $H\colon \{0,1\}^k \to \mathbb{G}$ be a function modeled as a random oracle. Define the function $F_s : \{0,1\}^k \to \mathbb{G}$, keyed by $s \in \mathbb{Z}_q$, as

$$F_s(x) = H(x)^s \in \mathbb{G}.$$

Since $F_{s_1}(x) \cdot F_{s_2}(x) = H(x)^{s_1+s_2} = F_{s_1+s_2}(x)$ then F_s is key homomorphic. Naor, Pinkas, and Reingold [107] proved that $\{F_s\}$ is a PRF family in the random oracle model, assuming DDH is hard in $\mathbb{G}$ (see Section 4.1 for a description of DDH).

Similarly, it is possible to construct (almost) key-homomorphic PRFs from the LWR problem in the random oracle model [42]. Let $p < q$ and let $H\colon \{0,1\}^k \to \mathbb{Z}_q$ be a function modeled as a random oracle. Define the function $F_s : \{0,1\}^k \to \mathbb{Z}_p$ as

$$F_s(x) = \lfloor \langle H(x), \mathbf{s} \rangle \rceil_p, \tag{21}$$

where $\lfloor x \rceil_p$ equals $\lfloor (p/q) \cdot x \bmod q \rceil \bmod p$. The function can be shown to be a secure PRF in the random oracle model, assuming the LWR problem is hard (see Section 4.2 for a description of LWR). Because rounding is not linear, the function F_s is not actually key homomorphic. However it is *almost* key homomorphic in that

$$F_{s_1}(x) + F_{s_2}(x) - F_{s_1+s_2}(x) \in \{-1, 0, 1\}.$$

This relaxed property turns out to be sufficient for many applications.

Application I: Distributed PRFs. Distributed PRFs support splitting of the secret key among n servers so that at least t servers are needed to evaluate the PRF. Evaluating the PRF is done without reconstructing the key at a single location. This can be useful, for instance, in mitigating the risk of master key leakage, as described in Section 1.1 (in the context of key derivation).

Key-homomorphic PRFs give a simple, one-round solution to this problem. For instance, for n-out-of-n sharing, server i stores a random key s_i and the overall PRF key is $s = s_1 + \cdots + s_n$. To evaluate $F_s(x)$ the client sends x to all servers and each server responds with $y_i = F_{s_i}(x)$. The client combines the results to obtain $F_s(x)$ using the key-homomorphism property.

For t-out-of-n sharing, the client first homomorphically multiplies the responses from the key servers by the appropriate Lagrange coefficients and then applies key homomorphism to add the results. This still works with an almost key-homomorphic PRF as long as the PRF range is sufficiently larger than the error term. The homomorphism error is eliminated by setting the output to the high-order bits of the computed value $F_s(x)$.

Application II: Proxy re-encryption. Given a ciphertext encrypted under one symmetric key, we would like to enable a proxy to transform the ciphertext to an encryption under a different symmetric key without knowledge of either key. To this end, the proxy is provided with a short re-encryption token t. Consider a ciphertext of the form $(r, F_s(r)+m)$ where F_s is a key-homomorphic PRF. To re-encrypt from key s to key s', one sends the re-encryption token $t = -s+s'$ to the proxy, who computes $F_t(r)$ and adds it to $F_s(r)+m$ to obtain $F_{s+t}(r)+m = F_{s'}(r)+m$. This also works with an almost key-homomorphic PRF except that here we pad each message m with a small number of zeros on the right to ensure that the small additive error term does not affect the encrypted plaintext after several re-encryptions.

Constructions without random oracles. The Boneh et al. PRF [42] is indexed by two public matrices $\mathbf{A}_0, \mathbf{A}_1 \in \{0,1\}^{n\times n}$, both sampled uniformly at random. The (secret) key for the PRF is a vector $\mathbf{s} \in \mathbb{Z}_q^n$. The PRF $F_{\mathbf{s}}\colon \{0,1\}^k \to \mathbb{Z}_p^n$ is defined as

$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s}^{\mathsf{T}} \cdot \prod_{i=1}^{k} \mathbf{A}_{x_i} \right\rceil_p . \tag{22}$$

The function $F_{\mathbf{s}}$ satisfies $F_{\mathbf{s}_1}(x) + F_{\mathbf{s}_2}(x) - F_{\mathbf{s}_1+\mathbf{s}_2}(x) \in \{-1,0,1\}^n$. It is thus almost key homomorphic in the same sense as the function F_s from (21).

The Banerjee–Peikert (BP) almost key-homomorphic PRF is a generalization of the Boneh et al. construction. A basic tool underlying the construction is the *bit decomposition* operator, which allows one to control the magnitude of individual entries in a matrix. Let $\ell = \lfloor \log q \rceil$, and for each $a \in \mathbb{Z}_q$, identify it with its unique integer residue in $\{0,\ldots,q-1\}$. Define the bit decomposition function $\mathbf{d}\colon \mathbb{Z}_q \to \{0,1\}^\ell$ as

$$\mathbf{d}(a) = (x_0, x_1, \ldots, x_{\ell-1}),$$

where $a = \sum_{i=0}^{\ell-1} x_i 2^i$ is the binary representation of a. Similarly, define the function $\mathbf{D}\colon \mathbb{Z}_q^{n\times m} \to \{0,1\}^{n\ell\times m}$ by applying $\mathbf{d}$ entry-wise.

For a full (but not necessarily complete) binary tree T, let $|T|$ denote the number of its leaves. If $|T| \geq 1$, let $T.l$, $T.r$ denote the left and right subtrees of T, and for a string $x \in \{0,1\}^{|T|}$ write $x = (x_l, x_r)$ for $x_l \in \{0,1\}^{|T.l|}$ and $x_r \in \{0,1\}^{|T.r|}$.

Public parameters: Moduli $q \gg p$, matrices $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_q^{n\times n\ell}$, and a binary tree T

Function key: A random $\mathbf{s} \in \mathbb{Z}_q^n$

Function evaluation: On input $x \in \{0,1\}^{|T|}$ define $F = F_{\mathbf{s}}\colon \{0,1\}^{|T|} \to \mathbb{Z}_p^n$ as

$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s}^{\mathsf{T}} \cdot \mathbf{A}_T(x) \right\rceil_p ,$$

where the function $\mathbf{A}_T\colon \{0,1\}^{|T|} \to \mathbb{Z}_q^{n\times n\ell}$ is defined recursively as

$$\mathbf{A}_T(x) = \begin{cases} \mathbf{A}_x, & \text{if } |T| = 1 \\ \mathbf{A}_{T.l}(x_l) \cdot \mathbf{D}(\mathbf{A}_{T.r}(x_r)), & \text{otherwise} \end{cases}$$

Size: $\mathrm{poly}(k,n)$

Depth: $O(s(T))$

Figure 10: The Banerjee–Peikert key-homomorphic PRF

The BP function generalizes the function of Boneh et al. This can be seen by setting public parameters $\mathbf{B}_b = \mathbf{D}(\mathbf{A}_b)$ and a left-spine tree T (as in Figure 11.(a)), which (after a minor adaptation) yields the construction $F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s}^{\mathsf{T}} \cdot \prod_i \mathbf{B}_{x_i} \right\rceil_p$ from (22).

Sequentiality and expansion. In terms of efficiency, the cost of computing $F_{\mathbf{s}}(x)$ is dominated by the evaluation of $A_T(x)$. Since linear operations over $\mathbb{Z}_q$ can be computed by depth-one (unbounded fan-in) arithmetic circuits, the circuit depth of the construction is proportional to the maximum nesting depth of $\mathbf{D}(\cdot)$ terms when one unwinds A_T. This is the *sequentiality*, $s(T)$, which measures the right depth of the tree, i.e., the maximum number of right edges over all root-to-leaf paths.

For security based on the hardness of the LWE problem, the public parameters $\mathbf{A}_0, \mathbf{A}_1$ and the secret key $\mathbf{s}$ are sampled uniformly at random over $\mathbb{Z}_q$. The modulus q and underlying LWE error rate, and hence also the dimension n needed to obtain a desired level of security, are determined by the maximum number of terms of the form $\mathbf{D}(\cdot)$ that are consecutively multiplied when one unwinds the recursive definition of A_T. This is the *expansion* $e(T)$, which measures the left depth of the tree, i.e., the maximum number of left edges over all root-to-leaf paths.

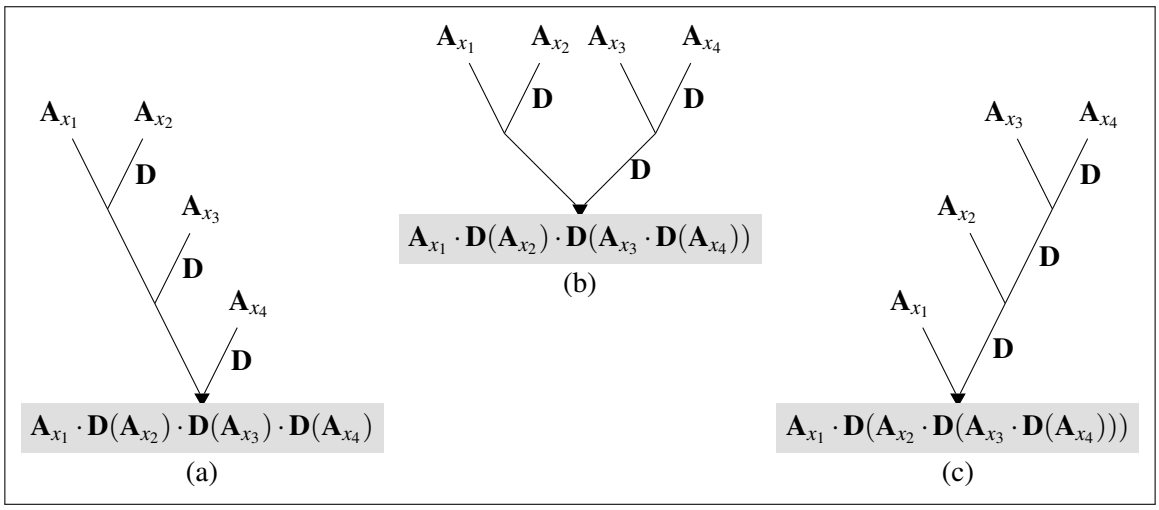

Figure 11: Instantiations of the Banerjee–Peikert PRF. All functions are on four inputs with the following sequentiality (s)/expansion (e) tradeoffs: (a) $s = 1, e = 3$, (b) $s = e = 2$, (c) $s = 3, e = 1$

Theorem 12 ([16])**.** *If the LWE problem is $(t, mn\ell, \varepsilon)$-hard for some B-bounded error distribution then $\{F_{\mathbf{s}}\}$ as defined in Figure 10 is an almost key-homomorphic (t', ε')-pseudorandom function family, where*

$$t' = t - \mathrm{poly}(n, m, k), \quad \textit{and} \quad \varepsilon' = pB(n\ell)^{e(T)}/q + \mathrm{poly}(\varepsilon).$$

The use of a left-spine tree T (as in Figure 11 (a)) yields a maximally parallel instantiation: Its sequentiality is $s(T) = 1$. This instantiation, however, also has maximal expansion $e(T) = |T| - 1$. In Theorem 12, the modulus q and error parameter $1/p$ have to grow exponentially with $e(T)$, so using a tree with large expansion leads to a strong hardness assumption on LWE, and therefore large secret keys and public parameters. Other trees give different sequentiality/expansion tradeoffs (as in Figure 11 (b) and (c)).

Efficiency. The cost of computing $F_{\mathbf{s}}(x)$ is dominated by the evaluation of $A_T(x)$, which can be done publicly without any knowledge of the secret key s. This property can be very useful for the efficiency of certain applications, such as the homomorphic evaluation of $F_{\mathbf{s}}$ given an encryption of s (see Section 1.1).

In addition, if $A_T(x)$ has been computed and all the intermediate matrices saved, then $A_T(x)$ can be incrementally updated for an x' that differs from x in just a single bit. As discussed in Sections 4.1 and 4.2, this can significantly speed up successive evaluations of $F_{\mathbf{s}}$ on related inputs, e.g., in a counter-like mode using a Gray code.

8.2 Puncturable PRFs

I cannot relate to emotional statements.
Which of the words here is incorrect?
LEONID LEVIN (1989)

A function family is puncturable if it can be evaluated at all but a single point $\mathring{x}$ and its value at this point is secret [86, 45]. The GGM construction has this property, so a puncturable PRF can in principle be obtained from any one-way function.

Sahai and Waters [129] show how to build public-key encryption from any puncturable function family using an indistinguishability obfuscator. Their construction has yet to produce encryption schemes that are as practical as available alternatives. Moreover, the existence of efficient and secure indistinguishability obfuscators is currently a subject of debate. Despite these shortcomings, the methodology holds significant conceptual appeal: If indistinguishability obfuscation is possible then public-key encryption can be constructed generically from any one-way function.

Definition 21 (Puncturable function family)**.** A *puncturing* of a function family F_s is a pair of deterministic algorithms Gen and $\mathring{F}$ such that for all s, $\mathring{x}$, and $x \neq \mathring{x}$, $\mathring{F}_{\mathring{s}}(x) = F_s(x)$, where $\mathring{s} = \mathrm{Gen}(s, \mathring{x})$. The puncturing is (t, ε)-*secure* if the distributions $(\mathring{s}, F_s(\mathring{x}))$ and $(\mathring{s}, r)$ are (t, ε)-indistinguishable for every $\mathring{x}$.

Every puncturable function family is a PRF, but the opposite may not hold. A PRF distinguisher can only make black-box queries to the function, while the adversary to a puncturable function family has a circuit that evaluates the PRF everywhere except at the challenge point. It may be interesting to study under which conditions the two notions can be formally separated. ⍰

Proposition 19. *If F_s has a (t, ε)-secure puncturing of size c then it is a $(t - qc, q, q\varepsilon)$-PRF for every q.*

Proof. Assume F_s is not a $(t - qc, q, q\varepsilon)$-PRF. By a hybrid argument the following two games are $(t - qc, \varepsilon)$-distinguishable for some $i \leq q$:

Answer the first $q - i$ queries randomly and the other i according to F_s.

Answer the first $q - i - 1$ queries randomly and the other $i + 1$ according to F_s.

After fixing the first $q - i - 1$ answers to maximize the distinguishing advantage we obtain that the games

F: Answer all i queries according to F_s

R: Answer the first query randomly and the other $i - 1$ according to F_s

are also $(t - qc, \varepsilon)$-distinguishable. Let $\mathring{x}$ be the first query made by the distinguisher D of size at most $t - qc$ and $\mathring{s} = \mathrm{Gen}(s, \mathring{x})$. The following circuit A of size t then breaks the assumed security of puncturing: Given a challenge $(\mathring{s}, y)$, emulate D by answering the first query by y and every subsequent query $x \neq \mathring{x}$ by $\mathring{F}_{\mathring{s}}(x)$. By the functionality of puncturing, $A(\mathring{s}, F_s(\mathring{x})) = D^F$ and $A(\mathring{s}, r) = D^R$, so the distributions $(\mathring{s}, F_s(\mathring{x}))$ and $(\mathring{s}, r)$ are (t, ε)-distinguishable. □

A puncturable PRF. The PRF F_s of Goldreich, Goldwasser, and Micali from Section 3.1 is puncturable. The puncturing is specified recursively by

$$\mathrm{Gen}(s,\mathring{a}\mathring{x}) = \big(\mathring{a}, G_{1-\mathring{a}}(s), \mathrm{Gen}(G_{\mathring{a}}(s),\mathring{x})\big),$$
$$\mathring{F}_{(\mathring{a},y,g)}(ax) = \begin{cases} F_y(x), & \text{if } a \neq \mathring{a} \\ \mathring{F}_g(x), & \text{if } a = \mathring{a}, \end{cases}$$

where $a,\mathring{a} \in \{0,1\}$ and $x,\mathring{x} \in \{0,1\}^{k-1}$. In the base case $k=0$, Gen and Eval output an empty string. The functionality requirement follows from the definition of the GGM pseudorandom function.

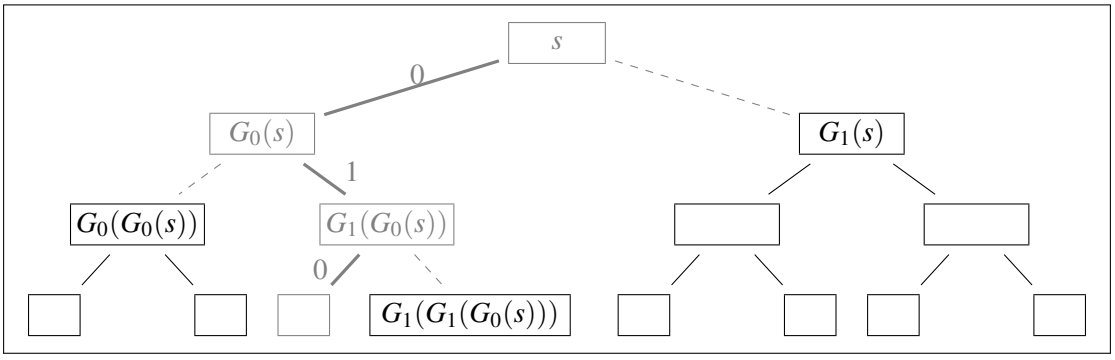

Figure 12: The GGM function family punctured at $\mathring{x} = 010$. The punctured key is $\mathring{s} = (0, G_1(s), 1, G_0(G_0(s)), 0, G_1(G_1(G_0(s))))$

Proposition 20. *If G is a (t,ε)-PRG of size c then $(\mathrm{Gen},\mathring{F})$ is a $(t - O(ck^3), 2k\varepsilon)$-secure puncturing of F_s.*

Proof. We prove the proposition by induction on k. When $k=0$, $(\mathring{s}, F_s(\lambda))$ and $(\mathring{s},r)$ are both distributed like (λ,s), so the two distributions are identical. Here λ is the empty string.

Now assume $(t - O(c(k-1)^3), 2(k-1)\varepsilon)$-security holds for input length $k-1$, namely the distributions

$$(\mathrm{Gen}'(s,\mathring{x}), F'_s(\mathring{x})) \quad \text{and} \quad (\mathrm{Gen}'(s,\mathring{x}), r)$$

are $(t - O(c(k-1)^3), 2(k-1)\varepsilon)$-indistinguishable for all $\mathring{x} \in \{0,1\}^{k-1}$. Here, F'_s is the GGM construction on input size $k-1$ and *Gen′* is the corresponding punctured key-generation algorithm.

Claim 22. *The distributions*

$$(\mathrm{Gen}(s,\mathring{a}\mathring{x}, F_s(\mathring{a}\mathring{x})) = \big(\mathring{a}, G_{1-\mathring{a}}(s), \mathrm{Gen}'(G_{\mathring{a}}(s),\mathring{x}), F'_{G_{\mathring{a}}(s)}(\mathring{x})\big) \quad \textit{and}$$
$$(\mathrm{Gen}(s,\mathring{a}\mathring{x}), r) = \big(\mathring{a}, G_{1-\mathring{a}}(s), \mathrm{Gen}'(G_{\mathring{a}}(s),\mathring{x}), r\big)$$

are $(t - O(ck^3), 2k\varepsilon)$-indistinguishable for every $\mathring{a} \in \{0,1\}$ and $\mathring{x} \in \{0,1\}^{k-1}$.

Proof. Since G is (t,ε)-pseudorandom and Gen′ can be computed using $O(k^2)$ calls to G, these two distributions are $(t - O(ck^2), \varepsilon)$-indistinguishable from

$$\big(\mathring{a}, r_1, \mathrm{Gen}'(r_0,\mathring{x}), F'_{r_0}(\mathring{x})\big) \quad \text{and} \quad \big(\mathring{a}, r_1, \mathrm{Gen}'(r_0,\mathring{x}), r\big), \tag{23}$$

respectively, for random strings r_0 and r_1 of length n. The distributions (23) are $(t - O(c(k-1)^3), 2(k-1)\varepsilon)$-indistinguishable by inductive assumption. The claim follows by the triangle inequality. □

This completes the inductive step. □

From private-key to public-key encryption. Much cryptographic evidence suggests that public-key encryption needs to be based on the hardness of "structured" problems such as discrete logarithms or finding close vectors in lattices. In contrast, private-key encryption follows from the existence of one-way functions, which are seemingly abundant. This divide is explained by theoretical results that rule out a fully-black-box construction of key exchange from one-way functions [79, 77].

On the other hand, there is an appealing blueprint for upgrading encryption from private-key to public-key: Publish an obfuscation of the encryption circuit as the public key. Assuming the encryption circuit can be obfuscated in the "virtual black-box" (VBB) sense, the resulting public-key scheme can be proved secure. Unfortunately, VBB obfuscation is impossible for all but a few rudimentary classes of functions [19].

Sahai and Waters propose applying this transformation to a variant of the encryption scheme (2) that was analyzed in Section 2.4. They show that the resulting public-key scheme is secure under the assumption that the obfuscator satisfies the seemingly weaker security notion of indistinguishability [19]. Garg et al. [59] give a candidate construction of an indistinguishability obfuscator. However this and related candidates were subsequently shown to be insecure. The feasibility of indistinguishability obfuscation is currently a highly active research area. Whether such obfuscation is attainable in its most general form is still uncertain.

Circuits C_0 and C_1 are *functionally equivalent* if $C_0(x) = C_1(x)$ for all x. Let $\mathbf{C}, \mathbf{C}'$ denote the classes of circuits of sizes c and c', respectively.

Definition 22 (Indistinguishability obfuscation [19])**.** We say that a probabilistic function $\mathrm{iO}\colon \mathbf{C} \to \mathbf{C}'$ is a (t,ε)-*indistinguishability obfuscator* if, for all $C \in \mathbf{C}$, it holds that $\mathrm{iO}(C)$ and C are functionally equivalent with probability 1 (*functionality*), and for all pairs of functionally equivalent circuits $C_0, C_1 \in \mathbf{C}$ the distributions $\mathrm{iO}(C_0)$ and $\mathrm{iO}(C_1)$ are (t,ε)-indistinguishable (*indistinguishability*).

The construction of Sahai and Waters relies on the existence of a puncturable family $F_s : \{0,1\}^{2n} \to \{0,1\}^{\ell}$, a PRG $G : \{0,1\}^n \to \{0,1\}^{2n}$, and an indistinguishability obfuscator iO. The starting point is the following variant of the private-key encryption scheme (2):

$$\mathrm{Enc}'_s(m;r) = (G(r), F_s(G(r)) \oplus m), \qquad \mathrm{Dec}'_s(y,z) = F_s(y) \oplus z. \tag{24}$$

The public-key encryption scheme of Sahai and Waters is shown in Figure 13. Its functionality follows from the functionality of indistinguishability obfuscation and the private-key scheme $(\mathrm{Enc}', \mathrm{Dec}')$.

Private key: A key s for the scheme $(\mathrm{Enc}', \mathrm{Dec}')$

Public key: The circuit $pk = \mathrm{iO}(\mathrm{Enc}'_s)$

Encryption/decryption: For message $m \in \{0,1\}^{\ell}$ and randomness $r \in \{0,1\}^n$

$$\mathrm{Enc}_{pk}(m;r) = pk(m;r), \qquad \mathrm{Dec}_s(y,z) = \mathrm{Dec}'_s(y,z).$$

Figure 13: The Sahai–Waters public-key encryption scheme

Proposition 21 ([129])**.** *Suppose that $\{F_s\}$ is a (t,ε)-puncturable family, that G is a (t,ε)-pseudorandom generator, and that* iO *is a (t,ε)-indistinguishability obfuscator, all of size at most c. Then for every two messages $m_0, m_1 \in \{0,1\}^{\ell}$, the following games are $(t - O(c), 6\varepsilon + 2 \cdot 2^{-n})$-indistinguishable:*

E_0: Sample pk, s, r and output $(pk, \mathrm{Enc}_{pk}(m_0;r))$.

E_1: Sample pk, s, r and output $(pk, \mathrm{Enc}_{pk}(m_1;r))$.

Proof. For $b \in \{0,1\}$ consider the following sequence of games:

E_b: Sample random s, r and $pk = \mathrm{iO}(\mathrm{Enc}'_s)$. Output $(pk, G(r), F_s(G(r)) \oplus m_b)$.

Y_b: Sample random $s, \mathring{y}$ and $pk = \mathrm{iO}(\mathrm{Enc}'_s)$. Output $(pk, \mathring{y}, F_s(\mathring{y}) \oplus m_b)$.

H_b: Sample random $s, \mathring{y} \notin \mathrm{Im}(G)$ and $pk = \mathrm{iO}(\mathrm{Enc}'_s)$. Output $(pk, \mathring{y}, F_s(\mathring{y}) \oplus m_b)$.

$\mathring{F}_b$: Sample random $s, \mathring{y} \notin \mathrm{Im}(G)$ and $\mathring{pk} = \mathrm{iO}(\mathring{\mathrm{Enc}}_{\mathring{s}})$. Output $(\mathring{pk}, \mathring{y}, F_s(\mathring{y}) \oplus m_b)$.

R: Sample random $s, \mathring{y} \notin \mathrm{Im}(G), r$ and let $\mathring{pk}$ be as in $\mathring{H}_b$. Output $(\mathring{pk}, \mathring{y}, r)$.

In the game $\mathring{F}_b$, we set $\mathring{s} = \mathrm{Gen}(s, \mathring{y})$, and $\mathring{\mathrm{Enc}}_{\mathring{s}}(m; r) = (G(r), \mathring{F}_{\mathring{s}}(G(r)) \oplus m)$.

Claim 23. *E_b and Y_b are $(t - c, \varepsilon)$-indistinguishable.*

Proof. The claim follows from the pseudorandomness of G and from the fact that iO is of size at most c. □

Claim 24. *Y_b and H_b are $(\infty, 2^{-n})$-indistinguishable.*

Proof. The distribution H_b is Y_b conditioned on $\mathring{y}$ not landing in the image of G. By a union bound, the probability of this event is at most

$$\Pr[\mathring{y} \in \mathrm{Im}(G)] \leq \sum_r \Pr[\mathring{y} = G(r)] \leq 2^n \cdot 2^{-2n} = 2^{-n}. \qquad \square$$

□

Claim 25. *H_b and $\mathring{F}_b$ are (t, ε)-indistinguishable.*

Proof. Fix s and $\mathring{y}$ to maximize the distinguishing advantage. After fixing, the distributions reduce to $\mathrm{iO}(\mathrm{Enc}'_s)$ and $\mathrm{iO}(\mathring{\mathrm{Enc}}_{\mathring{s}})$ (plus some fixed bits). Owing to the assumption $\mathring{y} \notin \mathrm{Im}(G)$, the circuits Enc'_s and $\mathring{\mathrm{Enc}}_{\mathring{s}}$ are functionally equivalent, so indistinguishability follows from the security of iO. □

Claim 26. *$\mathring{F}_b$ and R are $(t - O(c), \varepsilon)$-indistinguishable.*

Proof. Fix $\mathring{y}$ to maximize the distinguishing advantage. By the security of puncturing, $(\mathring{s}, F_s(\mathring{y}))$ and $(\mathring{s}, r)$ are (t, ε)-indistinguishable. Then the distributions

$$(\mathrm{iO}(\mathring{\mathrm{Enc}}_{\mathring{s}}), \mathring{y}, F_s(\mathring{y}) \oplus m_b) \quad \text{and} \quad (\mathrm{iO}(\mathring{\mathrm{Enc}}_{\mathring{s}}), \mathring{y}, r \oplus m_b)$$

are $(t - O(c), \varepsilon)$-indistinguishable. These are identical to $\mathring{F}_b$ and R, respectively. □

The proposition follows by applying Proposition 1 to the sequence $E_0, Y_0, H_0, \mathring{F}_0, R, \mathring{F}_1, H_1, Y_1, E_1$. □

The proof generalizes to the stronger adaptive security notion in which the challenge messages m_0 and m_1 are chosen after observing the public key.

Constrained PRFs and witness PRFs. *Constrained PRFs* extend puncturable PRFs in that they allow for more general constraints than mere puncturing [43, 86, 45]. A constrained PRF can be evaluated at any input x that satisfies some constraint circuit $C(x)$, but the constrained key reveals no information about the PRF values at points that do not satisfy the constraint to a computationally bounded adversary.

Brakerski and Vaikutanatan [46] construct an LWE-based function of constraint PRFs. Boneh and Waters [43] consider a more general definition in which security holds even if the adversary is given multiple constrained keys derived from different circuits. They give candidate constructions for restricted circuit classes whose security is based on strong hardness assumptions related to multilinear maps. ⑦

An even more intriguing notion is that of a *witness PRF* [140], which can be seen as a nondeterministic analogue of a constrained PRF. Here the constrained key allows for evaluation of $F_s(x)$, but only if the evaluator is also given a witness w that satisfies the constraint circuit $C(x,w)$. If no such w exists, the adversary obtains no information about the value $F_s(x)$. Witness PRFs are sufficient to realize some of the most interesting applications of indistinguishability obfuscation. What makes them intriguing is the possibility of constructing them from more standard hardness assumptions than the ones currently used to construct obfuscation. ⑦

Open Questions

In the established physical sciences ... a rich intellectual structure has been uncovered that reveals at any time a wide range of unsolved problems or puzzles. Solutions to these provide increased understanding of the field and further enrich the structure. As long as successful problem solving continues, progress is close to being guaranteed. The possibility of almost routine progress of this nature appears to be a fundamental aspect of science. Even if it is not the most celebrated aspect, it may be the most characteristic one.

LESLIE VALIANT (*Circuits of the Mind*, 1999)

Acknowledgements

This survey is dedicated to Oded Goldreich, a towering and charismatic figure, who has inspired generations of researchers through his limitless passion and devotion to intellectual inquiry. Oded has been and continues to be a dear mentor to us all, and sets a very high bar to aspire to. Without theory there would be chaos, and without Oded Goldreich there would be no theory as we know it.

We are deeply indebted to Shafi Goldwasser, Siyao Guo, Silvio Micali, Moni Naor, Chris Peikert, Omer Reingold, and Vinod Vaikuntanathan for invaluable advice. Last but not least, thanks to Yehuda Lindell for this wonderful initiative and for confidently steering the project into safe harbor.

Alon Rosen's work on this project was supported by ISF grant no. 1255/12, NSF-BSF Cyber Security and Privacy grant no. 2014/632, and by the ERC under the EU's Seventh Framework Programme (FP/2007-2013) ERC Grant Agreement no. 307952. Andrej Bogdanov's work was supported by Hong Kong RGC GRF grant CUHK14208215.

References

[1] William Aiello and Ramarathnam Venkatesan. Foiling birthday attacks in length-doubling transformations - Beneš: A non-reversible alternative to Feistel. In *Proceedings of EUROCRYPT '96*, volume 1070, pages 307–320, 1996.

[2] Adi Akavia. Solving hidden number problem with one bit oracle and advice. In *Advances in Cryptology - CRYPTO 2009: 29th Annual International Cryptology Conference*, pages 337–354, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[3] Adi Akavia, Andrej Bogdanov, Siyao Guo, Akshay Kamath, and Alon Rosen. Candidate weak pseudorandom functions in $\mathrm{AC}^0 \circ \mathrm{MOD}_2$. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 251–260, 2014.

[4] N. Alon and J.H. Spencer. *The Probabilistic Method*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2008.

[5] Noga Alon, Oded Goldreich, Johan Hastad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992.

[6] Jacob Alperin-Sheriff and Daniel Apon. Dimension-preserving reductions from LWE to LWR. *IACR Cryptology ePrint Archive*, 2016:589, 2016.

[7] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 57–74, 2013.

[8] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 657–677, 2015.

[9] Benny Applebaum. Bootstrapping obfuscators via fast pseudorandom functions. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, pages 162–172, 2014.

[10] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.

[11] Benny Applebaum and Shachar Lovett. Algebraic attacks against random local functions and their countermeasures. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1087–1100, 2016.

[12] Yuriy Arbitman, Moni Naor, and Gil Segev. Backyard cuckoo hashing: Constant worst-case operations with a succinct representation. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 787–796, 2010.

[13] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP (1)*, pages 403–415, 2011.

[14] László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.

[15] Abhishek Banerjee, Hai Brenner, Gaëtan Leurent, Chris Peikert, and Alon Rosen. SPRING: fast pseudorandom functions from rounded ring products. In *Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers*, pages 38–57, 2014.

[16] Abhishek Banerjee and Chris Peikert. New and improved key-homomorphic pseudorandom functions. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 353–370, 2014.

[17] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 719–737, 2012.

[18] Boaz Barak, Oded Goldreich, Shafi Goldwasser, and Yehuda Lindell. Resettably-sound zero-knowledge and its applications. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 116–125, 2001.

[19] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6:1–6:48, May 2012.

[20] David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *J. Comput. Syst. Sci.*, 38(1):150–164, 1989.

[21] Mihir Bellare. New proofs for NMAC and HMAC: security without collision-resistance. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, pages 602–619, 2006.

[22] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 1–15, 1996.

[23] Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 194–211, 1989.

[24] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, pages 341–358, 1994.

[25] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, pages 62–73, New York, NY, USA, 1993. ACM.

[26] Avraham Ben-Aroya and Amnon Ta-Shma. Constructing small-bias sets from algebraic-geometric codes. *Theory of Computing*, 9:253–272, 2013.

[27] Michael Ben-Or and Nathan Linial. Collective coin flipping, robust voting schemes and minima of banzhaf values. In *FOCS*, pages 408–416, 1985.

[28] Eli Ben-Sasson, Alessandro Chiesa, Michael Forbes, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. On probabilistic checking in perfect zero knowledge. *Electronic Colloquium on Computational Complexity (ECCC)*, 23(156), 2016.

[29] Itay Berman, Iftach Haitner, Ilan Komargodski, and Moni Naor. Hardness preserving reductions via cuckoo hashing. In *TCC*, pages 40–59, 2013.

[30] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *J. Cryptology*, 4(1):3–72, 1991.

[31] Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer, 1993.

[32] Nir Bitansky, Ran Canetti, Henry Cohn, Shafi Goldwasser, Yael Tauman Kalai, Omer Paneth, and Alon Rosen. The impossibility of obfuscation with auxiliary input or a universal simulator. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 71–89, 2014.

[33] Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a nash equilibrium. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:1, 2015.

[34] John Black and Phillip Rogaway. Ciphers with arbitrary finite domains. In Bart Preneel, editor, *Topics in Cryptology - CT-RSA 2002, The Cryptographer's Track at the RSA Conference, 2002, San Jose, CA, USA, February 18-22, 2002, Proceedings*, volume 2271 of *Lecture Notes in Computer Science*, pages 114–130. Springer, 2002.

[35] A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1):35–52, 1998.

[36] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Advances in cryptology (CRYPTO'93)*, pages 278–291. Springer, 1994.

[37] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, July 2003.

[38] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13(4):850–864, 1984.

[39] Andrej Bogdanov. Small bias requires large formulas. *Electronic Colloquium on Computational Complexity (ECCC)*, 24(91), 2017.

[40] Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 209–224, 2016.

[41] Andrej Bogdanov and Hoeteck Wee. A stateful implementation of a random function supporting parity queries over hypercubes. In *Proceedings of the 8th International Workshop on Randomization and Computation, RANDOM 2004*, pages 298–309, 2004.

[42] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 410–428, 2013.

[43] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, pages 280–300, 2013.

[44] Ravi Boppana. The average sensitivity of bounded-depth circuits. *Inf. Proc. Letters*, 63(5):257–261, 1997.

[45] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, pages 501–519, 2014.

[46] Zvika Brakerski and Vinod Vaikuntanathan. Constrained key-homomorphic prfs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 1–30, 2015.

[47] Mark Braverman. Poly-logarithmic independence fools bounded-depth boolean circuits. *Communications of the ACM*, 54(4):108–115, 2011.

[48] Hai Brenner, Lubos Gaspar, Gaëtan Leurent, Alon Rosen, and François-Xavier Standaert. FPGA implementations of SPRING - and their countermeasures against side-channel attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, pages 414–432, 2014.

[49] Nader H. Bshouty and Vitaly Feldman. On using extended statistical queries to avoid membership queries. *The Journal of Machine Learning Research*, 2:359–395, 2002.

[50] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 235–244, 2000.

[51] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 10:1–10:24. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[52] Fan R. K. Chung and Prasad Tetali. Communication complexity and quasi randomness. *SIAM J. Discrete Math.*, 6(1):110–123, 1993.

[53] Aloni Cohen and Saleet Klein. The GGM PRF is a weakly one-way family of functions. *IACR Cryptology ePrint Archive*, 2016:610, 2016.

[54] Yevgeniy Dodis, Jonathan Katz, John Steinberger, Aishwarya Thiruvengadam, and Zhe Zhang. Provable security of substitution-permutation networks. Cryptology ePrint Archive, Report 2017/016, 2017. `http://eprint.iacr.org/2017/016`.

[55] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM J. Comput.*, 39(2):606–645, 2009.

[56] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Proceedings on Advances in cryptology—CRYPTO '86*, pages 186–194, London, UK, UK, 1987. Springer-Verlag.

[57] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, pages 303–324, 2005.

[58] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[59] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Hiding secrets in software: a cryptographic approach to program obfuscation. *Commun. ACM*, 59(5):113–120, 2016.

[60] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a nash equilibrium. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 579–604, 2016.

[61] Oded Goldreich. Two remarks concerning the Goldwasser-Micali-Rivest signature scheme. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 104–110, 1986.

[62] Oded Goldreich. Towards a theory of software protection and simulation by oblivious rams. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 182–194, 1987.

[63] Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, New York, NY, USA, 2000.

[64] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, New York, NY, USA, 1 edition, 2008.

[65] Oded Goldreich. The GGM construction does NOT yield correlation intractable function ensembles. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 98–108, 2011.

[66] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *FOCS*, pages 464–479, 1984.

[67] Oded Goldreich, Shafi Goldwasser, and Asaf Nussboim. On the implementation of huge random objects. *SIAM J. Comput.*, 39(7):2761–2822, 2010.

[68] Oded Goldreich and Rafail Ostrovsky. Software protection and simulation on oblivious rams. *J. ACM*, 43(3):431–473, 1996.

[69] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 553–562, 2005.

[70] W. T. Gowers. A new proof of Szemerédi's theorem. *Geom. Funct. Anal.*, 11(3):465–588, 2001.

[71] Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *SIAM J. Comput.*, 42(3):1405–1430, 2013.

[72] Prahladh Harsha and Srikanth Srinivasan. On polynomial approximations to AC^0. *CoRR*, abs/1604.08121, 2016.

[73] Johan Hastad. *Computational limitations of small-depth circuits*. MIT Press, 1987.

[74] Johan Hastad. The shrinkage exponent of de Morgan formulas is 2. *SIAM J. Comput.*, 27(1):48–64, 1998.

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

[76] Pavel Hubáček and Eylon Yogev. Hardness of continuous local search: Query complexity and cryptographic lower bounds. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1352–1371, 2017.

[77] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 134–147. IEEE Computer Society, 1995.

[78] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 356–364. ACM, 1994.

[79] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 44–61, 1989.

[80] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 433–442. ACM, 2008.

[81] Jeffrey C Jackson. An efficient membership-query algorithm for learning dnf with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997.

[82] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC, 2nd edition, 2014.

[83] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[84] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, USA, 1994.

[85] Michael Kharitonov. Cryptographic lower bounds for learnability of boolean functions on the uniform distribution. *J. Comput. Syst. Sci.*, 50(3):600–610, 1995.

[86] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 669–684, 2013.

[87] Joe Kilian and Phillip Rogaway. How to protect DES against exhaustive key search. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 252–267, 1996.

[88] Shrinivas Kudekar, Santhosh Kumar, Marco Mondelli, Henry D. Pfister, Eren Sasoglu, and Rüdiger L. Urbanke. Reed-muller codes achieve capacity on erasure channels. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 658–669, 2016.

[89] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, December 1982.

[90] Leonid A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.

[91] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993.

[92] Shachar Lovett and Emanuele Viola. Bounded-depth circuits cannot sample good codes. *Computational Complexity*, 21(2):245–266, 2012.

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

[94] Anna Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 597–612, 2002.

[95] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *Proceedings of the 9th International Workshop on Randomization and Computation, RANDOM 2005*, pages 378–389, 2005.

[96] Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. SWIFFT: A modest proposal for FFT hashing. In *FSE*, pages 54–72, 2008.

[97] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.

[98] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, EUROCRYPT '93, pages 386–397, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[99] Ueli M. Maurer, Krzysztof Pietrzak, and Renato Renner. Indistinguishability amplification. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 130–149. Springer, 2007.

[100] Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 120–130, 1999.

[101] Eric Miles and Emanuele Viola. Substitution-permutation networks, pseudorandom functions, and natural proofs. *J. ACM*, 62(6):46:1–46:29, December 2015.

[102] Ben Morris. Improved mixing time bounds for the thorp shuffle. *Combinatorics, Probability & Computing*, 22(1):118–132, 2013.

[103] Ben Morris, Phillip Rogaway, and Till Stegers. How to encipher messages on a small domain. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 286–302. Springer, 2009.

[104] Dana Moshkovitz and Michal Moshkovitz. Mixing implies lower bounds for space bounded learning. *Electronic Colloquium on Computational Complexity (ECCC)*, 24(17), 2017.

[105] Elchanan Mossel, Ryan O'Donnell, and Rocco P. Servedio. Learning juntas. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 206–212, New York, NY, USA, 2003. ACM.

[106] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput*, 22:838–856, 1993.

[107] Moni Naor, Benny Pinkas, and Omer Reingold. Distributed pseudo-random functions and kdcs. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 327–346, 1999.

[108] Moni Naor and Omer Reingold. From unpredictability to indistinguishability: A simple construction of pseudo-random functions from macs (extended abstract). In *Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '98, pages 267–282, London, UK, UK, 1998. Springer-Verlag.

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

[110] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.

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

[112] Moni Naor, Omer Reingold, and Alon Rosen. Pseudo-random functions and factoring (extended abstract). In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 11–20, 2000.

[113] Moni Naor and Eylon Yogev. Bloom filters in adversarial environments. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 565–584, 2015.

[114] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[115] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, October 1994.

[116] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, New York, NY, USA, 2014.

[117] National Institute of Standards and Technology. Announcing request for candidate algorithm nominations for the Advanced Encryption Standard (AES). http://csrc.nist.gov/archive/aes/pre-round1/aes_9709.htm, 1997.

[118] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342, 2009.

[119] L. Pitt and M. K. Warmuth. Reductions among prediction problems: On the difficulty of predicting automata. In *Structure in Complexity Theory Conference, 1988. Proceedings., Third Annual*, pages 60–69, Jun 1988.

[120] Ran Raz. The BNS-Chung criterion for multi-party communication complexity. *Computational Complexity*, 9(2):113–122, 2000.

[121] Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 266–275, 2016.

[122] Ran Raz. A time-space lower bound for a large class of learning problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 24(20), 2017.

[123] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Matematicheskie Zametki*, 41(4):598–607, 1987.

[124] Alexander A. Razborov and Steven Rudich. Natural proofs. In *STOC*, pages 204–213, 1994.

[125] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):1–40, 2009. Preliminary version in STOC 2005.

[126] John H. Reif and Stephen R. Tate. On threshold circuits and polynomial computation. *SIAM J. Comput.*, 21(5):896–908, 1992.

[127] Omer Reingold, Thomas Steinke, and Salil Vadhan. *Pseudorandomness for Regular Branching Programs via Fourier Analysis*, pages 655–670. Springer Berlin Heidelberg, 2013.

[128] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179, 1978.

[129] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 475–484, 2014.

[130] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[131] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[132] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *STOC*, pages 77–82, 1987.

[133] Avishay Tal. Tight bounds on the fourier spectrum of AC^0. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:174, 2014.

[134] S. Vadhan and J. Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. Technical Report TR11-141, Electronic Colloquium on Computational Complexity, 2011.

[135] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984.

[136] Emanuele Viola and Avi Wigderson. Norms, XOR lemmas, and lower bounds for polynomials and protocols. *Theory of Computing*, 4(1):137–168, 2008.

[137] Andrew C. C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

[138] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 1–10, 1985.

[139] M. Zhandry. How to construct quantum random functions. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 679–687, Oct 2012.

[140] Mark Zhandry. How to avoid obfuscation using witness PRFs. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, pages 421–448, 2016.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il