SoK: On Shallow Weak PRFs

Christina Boura, Geoffroy Couteau, Léo Perrin, Yann Rotella

IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2025, No. 3, pp. 289–336. DOI:10.46586/tosc.v2025.i3.289-336

SoK: On Shallow Weak PRFs

A Common Symmetric Building Block for MPC Protocols

Christina Boura[1], Geoffroy Couteau[1], Léo Perrin[2] and Yann Rotella[3]

[1] IRIF, CNRS, Université Paris Cité, Paris, France
christina.boura@irif.fr, couteau@irif.fr
[2] Inria, Paris, France
leo.perrin@inria.fr
[3] Université Paris-Saclay, UVSQ, CNRS, Laboratoire de Mathématiques de Versailles, Versailles, France
yann.rotella@uvsq.fr

Abstract. A growing number of advanced cryptographic protocols and constructions rely on symmetric primitives known as weak pseudo-random functions (wPRFs). These functions differ significantly from traditional PRFs: they operate in constrained models where inputs are sampled uniformly at random and are not chosen by the adversary. In practice, many of these functions are implemented as shallow, non-iterated constructions with simple circuit representations.

This Systematization of Knowledge (SoK) provides a unified view of shallow wPRFs (swPRFs), which we define as wPRFs computable by low-depth circuits and primarily used in different secure computation protocols. We identify and classify four main families of swPRFs—alternating moduli wPRFs, Goldreich's PRG family, and the VDLPN and EALPN constructions—presenting formal definitions, algorithmic descriptions, known variants, cryptanalytic results, and concrete parameter sets for each.

In addition to surveying the literature, our goal is to shift the focus from asymptotic analyses to concrete cryptanalysis. To this end, we provide a set of cryptanalytic challenges along with reference `SAGE` implementations for all the primitives discussed. We aim to encourage the symmetric cryptography community—particularly cryptanalysts—to rigorously evaluate the practical security levels offered by swPRFs, as concrete analyses are currently lacking. Given their growing use in high-level protocols and constructions, any cryptanalytic breakthrough on these primitives could directly affect the security of the broader cryptographic systems that rely on them.

Keywords: shallow weak PRFs · alternating moduli · Goldreich's PRG · VDLPN · EALPN

1 Introduction

Symmetric cryptography has provided other areas of cryptography and information security with a wide range of secure algorithms, from high-throughput stream ciphers to lightweight message authentication codes, and from low-latency block ciphers to zero-knowledge-friendly hash functions. Meanwhile, another class of symmetric primitives has been gaining ground: *shallow weak pseudo-random functions (swPRFs)*, which have so far mostly escaped the attention of the symmetric cryptography community.

These functions are symmetric in nature in the sense that they map a secret input to secure output using a procedure that depends only on a secret key, much like the "usual" pseudo-random functions. However, these primitives must also satisfy different properties.

Received: 2025-06-01 Accepted: 2025-08-01 Published: 2025-09-25

Security Model. They are *weak* in the sense that the adversary is only allowed to perform random queries, and that the number of queries is severely limited. To put it differently, they are only intended to be secure in the known-plaintext setting, and under a low number of queries.

Input/Output Sizes. The input and key of such functions typically consist of several hundred bits, while the output can be as small as a single bit. This stands in contrast to traditional PRFs, which are generally expected to produce outputs of a size comparable to their inputs.

Cost Model. The main difference with classical constructions lies in the *shallow* requirement: these functions are not iterated. They consist essentially of a single round.

It should then come as no surprise that such primitives look very different from what is usually considered in symmetric cryptography. For example, the mod-2/mod-3 function introduced in [BIP+18] is fully specified by the following equation:

$$F_k(x) := \left(\sum_{i=0}^{n-1} k_i x_i \mod 2 + \sum_{i=0}^{n-1} k_i x_i \mod 3 \right) \mod 2, \quad \text{for } x, k \in \mathcal{Z}_2^n \ ,$$

where $\mathcal{Z}_n$ denotes the set $\{0, \ldots, n-1\}$. We provide more details about this specific function in Section 4. As we can see, none of the usual building blocks are present: no round function, no S-box layer, no Feistel round... And yet the security goals are very reminiscent of what is usually expected in symmetric cryptography.

To better understand these functions, we must first introduce their formal definition. This reveals one of the main challenges of this work: the vocabulary and notions used in complexity theory and symmetric cryptography differ. We begin by describing the security notion that shallow weak PRFs (swPRFs) are expected to satisfy, using both languages.

1.1 Security Definitions

The different definitions of PRFs intend to capture a similar concept that was already explicitly mentioned in the seminal work of Luby and Rackoff [LR88]. Consider a family of functions $h_k : \mathcal{Z}_2^{64} \to \mathcal{Z}_2^{64}$, where k is a secret key. The intuitive notion to capture is, in their words, the following one.

> Say that we have two black boxes, one of which computes a fixed randomly chosen function from the set of functions of $\mathcal{Z}_2^{64}$ and the other computes h_k for a fixed randomly chosen k. Then no algorithm which examines the boxes by feeding inputs to them and looking at the outputs can obtain, in a "reasonable" time, any "significant" idea about which box is which.

We need to limit the power of the attackers using a definition of what "reasonable" means as an unbounded adversary could always just brute-force the key. Similarly, the output of the distinguishing algorithm has to be "significant". For example, systematically returning "random function" would be correct sometimes, but would not be interesting at all.

Starting from this insight, two different definitions of "reasonable" have eventually solidified. The first was already present in [LR88], and relies on complexity classes. The other, more common in modern symmetric cryptography, relies on precise upper bounds.

Complexity Theory View. A pseudorandom function (PRF), as originally introduced in [GGM84], is defined as follows.

Definition 1 (PRF in complexity theory)**.** A PRF is a family of efficiently computable keyed functions such that no polynomial-time adversary can distinguish the input-output

behavior of a randomly chosen function from the family from that of a uniformly random function with the same domain and range.

Informally, the "reasonable" power of the adversary lies in the *asymptotic* complexity of their attack: it has to *scale* polynomially at most (and not exponentially).

This definition is, in essence, based on the notion of *computational indistinguishability*. Two families of sets $\{S_n\}_{n>0}$ and $\{S'_n\}_{n>0}$ are computationally indistinguishable if, for any probabilistic polynomial-time algorithm, the advantage in determining whether x is from S_n or S'_n is a negligible function of n. This is denoted by $S_n \approx_c S'_n$.

In this setting, it is easy to define a *weak* pseudorandom function (wPRF). Such a function offers a relaxed guarantee: it is only required to be indistinguishable from a truly random function when the adversary is restricted to querying the function on uniformly random inputs.

Definition 2 ((Weak) Pseudorandom Function (wPRF, PRF), [GGM84, NR95])**.** Let $\lambda \in \mathbb{N}$ be a security parameter. A (weak) pseudorandom function with domain $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$, key space $\mathcal{K} = \{\mathcal{K}_\lambda\}_{\lambda\in\mathbb{N}}$, and range $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$, consists of the following two polynomial-time algorithms:

- $\mathsf{KeyGen}(1^\lambda) \to \mathsf{msk}$, a probabilistic algorithm that takes as input the security parameter λ, and outputs a master secret key $\mathsf{msk} \in \mathcal{K}$, and
- $\mathsf{Eval}(\mathsf{msk}, x) \to y$, a deterministic algorithm that, given as input a secret master key msk and a value $x \in \mathcal{X}$, outputs a value $y \in \mathcal{Y}$.

We say that the pair $(\mathsf{KeyGen}, \mathsf{Eval})$ is a **weak pseudorandom function (wPRF)** if for any Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$ and any polynomially bounded number $Q \in \mathbb{N}$, representing the maximal number of authorised samples, it holds that

$$\left\{ \left((x_i, \mathsf{Eval}(\mathsf{msk}, x_i))_{i\in[Q]} \right) \middle| \begin{array}{c} \mathsf{msk} \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda) \\ \forall i \in [Q] : x_i \leftarrow\!\$\ \mathcal{X} \end{array} \right\} \approx_c \left\{ \left((x_i, y_i)_{i\in[Q]} \right) \middle| \begin{array}{c} \forall i \in [Q] : \\ x_i \leftarrow\!\$\ \mathcal{X},\ y_i \leftarrow\!\$\ \mathcal{Y} \end{array} \right\}.$$

In this context, the main goal of the designers of such primitives is to identify the asymptotic behavior of the best attack techniques in order to determine whether a polynomial-time adversary can gain an advantage.

Symmetric Cryptography View. In modern symmetric cryptography, PRFs are usually not intended to work for any block size n. Instead, the input and output size are fixed, and while the security definition also relies on a game, it differs in a crucial way: there is no asymptotic reasoning, we instead explicitly upper-bound the probability of success of a distinguishing algorithm Dist that is allowed at most q queries to the function under scrutiny.

Definition 3 (PRF security)**.** Let n and k be positive integers. Let $F = \{F_K\}_{K\in\mathcal{Z}_2^k}$ be a keyed function family with $F_K : \mathcal{Z}_2^n \to \mathcal{Z}_2^n$. We say that F achieves (q, T, ε)-PRF security if for any distinguisher Dist running in time at most T and making at most q queries, the distinguishing advantage is at most ε, i.e.,

$$\left| \Pr\left[\mathsf{Dist}^{F_K} = 1 \right] - \Pr\left[\mathsf{Dist}^{\mathcal{R}} = 1 \right] \right| \leq \varepsilon,$$

where the key K is sampled uniformly at random from $\mathcal{Z}_2^k$, and $\mathcal{R}$ is a uniformly random function from $\mathcal{Z}_2^n$ to $\mathcal{Z}_2^n$.

Since the parameters of symmetric crypto-systems are fixed (e.g., the AES operates on 128 bits), such definitions are better suited to their study.

In this description, the attacker is allowed to do *chosen plaintext* queries, meaning that the distinguishing algorithm Dist can choose on which x to evaluate the black box. It is however possible to instead restrict the attacker to *known plaintexts*, i.e., to enforce that the inputs x for the black box are chosen uniformly using an external source of randomness. In this context, we can provide an alternative definition of a weak PRF that is not asymptotic.

Definition 4 (wPRF security)**.** Let n, k, and q be positive integers. A family of functions $F_K : \mathcal{Z}_2^n \to \mathcal{Z}_2^n$, indexed by a key $K \in \mathcal{Z}_2^k$, is said to achieve (q, T, ε)-weak PRF security if, for every distinguisher Dist running in time at most T and making at most q queries on inputs $x_1, \ldots, x_q \in \mathcal{Z}_2^n$ drawn independently and uniformly at random, we have:

$$\Big| \Pr\left[\mathsf{Dist}\left(x_1, \ldots, x_q, F_K(x_1), \ldots, F_K(x_q)\right) = 1\right] - \Pr\left[\mathsf{Dist}\left(x_1, \ldots, x_q, \mathcal{R}(x_1), \ldots, \mathcal{R}(x_q)\right) = 1\right] \Big| \leq \varepsilon \ .$$

where the key K is sampled uniformly at random from $\mathcal{Z}_2^k$, and $\mathcal{R}$ is a uniformly random function from $\mathcal{Z}_2^n$ to $\mathcal{Z}_2^n$.

Concrete Security. In practice, defining and interpreting the runtime of the adversary remains a subtle issue for which we are not aware of any fully satisfactory answer. In many symmetric-key attack scenarios, such as differential or linear cryptanalysis, the dominating step is a final exhaustive search over a part of the key space. As a result, the total number of calls to the function under attack is often used as a proxy for the time complexity. Security claims are then made against adversaries performing up to 2^k such evaluations, assuming that the final brute-force phase is unavoidable.

However, this reasoning breaks down for attack techniques based on algebraic methods, such as solving systems of equations. In such cases, the bottleneck is not a brute-force stage but rather the resolution of the underlying mathematical problem, which may not involve any calls to the primitive at all after a preprocessing phase. Comparing the performance of such attacks to brute-force-style security claims is therefore problematic.

Because most of the attacks applicable to the weak PRFs we consider do not rely on exhaustive search, we argue that it is more relevant to express the time complexity in terms of the number of elementary operations, rather than just oracle calls. That is, we adopt the view that security should be measured in terms of the total cost (time or operations) required to distinguish the function from a random oracle.

In the rest of this paper, we aim to move beyond asymptotic reasoning and instead adopt a concrete perspective: for a given target security level λ, we consider a weak PRF secure if no adversary can distinguish it from a random function with success probability greater than $\varepsilon = T/2^\lambda$ in less than T operations[1]. This includes both oracle queries and any algebraic or pre-processing steps the adversary may perform; however, a separate bound q on the total number of oracle queries allowed to the adversary is also typically assumed (in real-world contexts, the number of queries corresponds to the number of samples observed by the adversary. It is not directly under their control and is often, in practice, much smaller than 2^λ). The set of operations considered can depend on the context, but one must be clear and exact on what is being counted.

We note that in several existing proposals, the chosen parameters are justified post hoc to avoid trivial breakage, and the resulting security level is sometimes arbitrarily bounded (e.g., “80 bits”) to compensate for performance issues. This practical “patching” approach

[1]This is the traditional way to measure concrete security of symmetric primitives [DS09]. However, it was recently challenged by Micciancio and Walter [MW18] who showed that $\varepsilon = \sqrt{T/2^\lambda}$ is more accurate for decision primitives, such as wPRFs. Nevertheless, we are typically interested in the setting where ε is close to 1, in which case both measures coincide.

highlights the need for a more rigorous study based on concrete cryptanalysis, something we actively advocate for by writing this survey.

1.2 Practical Relevance

While figuring out the existence (or inexistence) of secure wPRFs in some complexity class could be seen as a problem of purely theoretic significance, it recently received renewed attention because of the very practical need for such functions in some protocols. We will provide more details on this topic further in Section 3.

In these contexts, the "symmetric" definitions of (weak) PRF security take precedence since the parameters are then fixed. This change of paradigm, from complexity theory to computer security, implies several significant changes.

Irrelevance of Asymptotic Reasoning. While figuring out the asymptotic behaviour of an adversary is the correct way to investigate complexity classes, it is hardly relevant when moving to concrete computer security since the aim in this case is not to let a parameter go to infinity, it is instead to fix it to a specific value. Thus, the attacks that were relevant asymptotically might be of lesser interest, while new attack vectors that were previously cast aside because they are exponential may instead become a threat. In a nutshell, an attack with a runtime equivalent to $2^{n/3}$ basic operations is, in practice, far more threatening than one with a runtime of n^{10} similar operations when $n = 128$, where they amount roughly to $2^{42.7}$ and 2^{70}, respectively.

Performance Trade-Offs. When a function is intended to be implemented, adding structure can become necessary, or at least welcome, to decrease the cost of its implementation. On the other hand, such structure could potentially open new attack directions. The investigation of such trade-offs becomes much more relevant when leaving the world of complexity theory.

1.3 High-level View of Shallow wPRFs and Their Cryptanalysis

Among the various cryptanalytic techniques, differential [BS91] and linear cryptanalysis [Mat94] are two of the most fundamental and powerful attacks in symmetric cryptography. Differential cryptanalysis relies on pairs of inputs that satisfy a specific difference and is therefore most relevant in a chosen-plaintext setting. However, this requirement makes it ill-suited for the analysis of shallow weak PRFs: due to input size constraints and data limitations, the probability of observing a pair with a desired input difference is too low for the technique to be applicable.

In contrast, linear cryptanalysis is generally applicable in a known-plaintext setting, making it a more natural approach in our context. Nevertheless, in the case of shallow weak PRFs, the structure of the "noise", which corresponds to the error term in the linear approximation, is deterministic and derived from both the input and the secret key in a nonlinear way. As such, generic linear cryptanalysis is unlikely to succeed without carefully modeling the noise generation process specific to each construction.

This observation highlights a broader point: all the primitives studied in this paper can be viewed, at a high level, as instances of the Learning Parity with Noise (LPN) problem, with the important distinction that the noise is not random but deterministically generated from the secret and the input. These constructions are explicitly designed to resist attacks that abstract the noise as random with fixed bias. Hence, any successful cryptanalysis must go beyond the generic LPN model and incorporate the properties of the noise function.

The same phenomenon arises in the context of algebraic attacks. The general strategy is to collect enough equations involving the secret key and attempt to solve the resulting nonlinear system, often via linearization. All the constructions discussed here are built

to resist such generic attack methods, typically by ensuring that the algebraic degree is sufficiently high. As with decoding-based attacks, the actual shape and structure of the equations depend heavily on the specific definition of each wPRF. Thus, any meaningful algebraic analysis must be carefully adapted to the specific design, as generic bounds and arguments are insufficient to analyze the security.

This situation is actually very close to classical block cipher cryptanalysis: differential cryptanalysis is a general technique applicable theoretically to almost any design, yet the actual strategy and refinements to be used vary widely depending on the cipher's structure. Similarly, while the constructions presented here share high-level goals and abstractions, their internal design choices differ significantly—necessitating distinct, dedicated cryptanalytic approaches. As a conclusion, attacking shallow weak PRFs requires building upon generic techniques while carefully taking into account the algebraic, statistic and other properties of each primitive. A difficult challenge lies in identifying input patterns that are likely to occur and which lead to a detectable distinguishing property. This suggests that analysing these constructions from a combinatorial point of view could be an interesting direction.

1.4 Our Contributions

Our aim in this Systematization of Knowledge (SoK) paper is to describe *shallow* wPRF (swPRFs)—what they are and why they are relevant in practice. This involves both a detailed survey of the literature and a translation of their security goals away from asymptotics. To this end, we also propose cryptanalysis challenges, along with `SAGE` reference implementations of the swPRFs we describe: practical security can only be studied through cryptanalysis, and our goal is to simplify this analysis.

Outline. The rest of the article is organized as follows. Section 2 introduces preliminary notions on shallow weak PRFs, while Section 3 discusses their intended applications and outlines the associated cost and security models. We then present several families of shallow weak PRFs, providing both mathematical and algorithmic descriptions, known variants, any known cryptanalysis results and proposed parameter sets that serve as targets for cryptanalysis. Specifically, we cover four distinct families, which are detailed in Sections 4,5,6, and 7, respectively. A `SAGE` implementation of each of these families can be found in

https://github.com/lpp-crypto/PCF-cryptanalysis/tree/main/implementations

to simplify the task of future cryptanalysts.

2 General Background

2.1 Definitions and Notations

Mathematics. As stated before, we denote by $\mathcal{Z}_n$ the ring consisting of the integers in $\{0, \ldots, n-1\}$. We also denote by $[N]$ the set $\{1, \ldots, N\}$, and $[a, b]$ the interval $\{a, a+1, \ldots, b\}$.

A function mapping $\mathcal{Z}_2^n$ to $\mathcal{Z}_2$ is a *binary* or *Boolean* function. Every such function has a unique representation as a multivariate polynomial over $\mathcal{Z}_2$ in the variables $(x_0, \ldots, x_{n-1})$, expressed as a sum of monomials of the form $\prod_{i=0}^{n-1} x_i^{u_i}$ with $u_i \in \mathcal{Z}_2$. The *Hamming weight* of a vector $(u_0, \ldots, u_{n-1}) \in \mathcal{Z}_2^n$ is the number of non-zero entries in the vector. The *algebraic degree* of a Boolean function is the maximum number of variables appearing in any monomial with a non-zero coefficient in its polynomial representation.

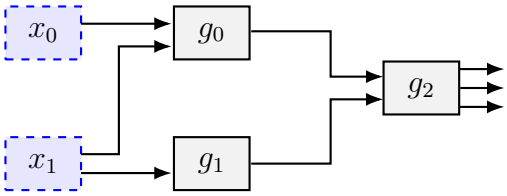


Figure 1: Graphical representation of a small circuit with two input nodes. The circuit has depth 2. The gate g_0 has indegree 2 and outdegree 1; g_1 has both indegree and outdegree equal to 1; and g_2 has indegree 2 and outdegree 3.

Parameters. In practice, the message length and key lengths of the function we study depend on a security parameter λ. To simplify notations, we will explicitly omit indexing all relevant parameters by λ. For instance, we will write n instead of $n(\lambda)$ and $\mathcal{K}$ instead of $\mathcal{K}_\lambda$ for the key space.

2.2 Circuits

In order to formally define the notion of *shallow* PRF, we need to introduce *circuits*.

Circuits. A circuit is a directed acyclic graph composed of two types of nodes: *input nodes*, which have indegree 0, and *gates*, which have indegree at least 1. The indegree corresponds to the number of edges ending in this node, and the outdegree to the number of edges leaving it.

Each gate with indegree in and outdegree out is labeled by a function $g : \{0,1\}^{\mathsf{in}} \to \{0,1\}^{\mathsf{out}}$. To evaluate a circuit C with n input nodes on an input $x \in \{0,1\}^n$, we proceed as follows: each input node i is labeled with the corresponding bit x_i. Then, for each gate whose in input wires have been assigned values $(z_1, \ldots, z_{\mathsf{in}})$, we compute the output $g(z_1, \ldots, z_{\mathsf{in}})$ and assign its out bits to the outgoing wires of the gate. Gates with outdegree 0 are called *output gates*, and the final output of the circuit is the bitstring composed of the values on these output wires. We define a *circuit class* as a family $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$, where $\mathcal{C}_n$ denotes the set of allowed circuits with n input nodes. Figure 1 presents a very simple example of circuit.

A circuit class typically restricts the set of allowed gate types, and may impose additional structural constraints, such as bounds on indegree, outdegree, depth, size, or overall topology.

Shallow wPRF. A shallow weak PRF (wPRF) is a construction whose evaluation can be performed by a circuit of small depth. Formally, fix a class of circuits $\mathcal{C}$ and a function $s : \mathbb{N} \to \mathbb{N}$. A pair of algorithms $(\mathsf{KeyGen}, \mathsf{Eval})$ is said to define an *s-shallow wPRF over $\mathcal{C}$* if for every secret key msk in the range of KeyGen, the function $f : x \mapsto \mathsf{Eval}(\mathsf{msk}, x)$ can be computed by a circuit in $\mathcal{C}$ of depth at most $s(|x|)$.

The function s expresses how the allowed circuit depth grows with the input size. For instance, if $s(n) = \lfloor \log n \rfloor$ (where $\lfloor \cdot \rfloor$ denotes rounding down), then the wPRF is said to be *logarithmically shallow*. The notion of shallow wPRFs is relevant in contexts where parallel-time complexity is a constraint, since small-depth circuits can be computed in very few parallel steps, assuming enough computational units are available.

3 Shallow wPRFs in Higher-Level MPC Protocols

The existence of pseudorandom functions computable by low-depth circuits has a long history, with strong ties to fundamental questions in complexity theory such as circuit lower bounds [RR94, MV12, SV12], derandomization [NW88, Wil13], and learning theory [Val84, KV94, BCG+21b]. These connections are not the focus of this work and will not be covered, as they are typically of little relevance to cryptanalysis and symmetric cryptography.[2] In this work, we focus on the use of low-depth weak pseudorandom functions in practically-minded higher-level secure computation protocols (we will use the terms secure computation and MPC, short for MultiParty Computation, interchangeably). More precisely, we will be mostly concerned with cryptographic constructions that use a wPRF as an *inner component* within an *outer component*, where the latter is a high-end cryptographic building block (a primitive or a protocol) that uses internally the *circuit description* of the wPRF (that is, we do not consider cryptographic protocols that make a "black-box" use of a wPRF, since such protocols are not sensitive to low-level details of the computational model where the wPRF is computed). In equivalent terms, these constructions compile a wPRF whose circuit description meets specific structural constraints into a cryptographic primitive that fulfills a set of security and efficiency requirements.

Unlike the traditional study of shallow wPRFs, whose focus was generally on low-depth wPRFs in standard computational models of complexity theory[3], the computational models that best capture these target applications are slightly more exotic at first sight. Indeed, as will become clear next, there is a duality between the computational model in which a wPRF is implemented, and the class of functions that the outer cryptographic primitive or protocol can "evaluate" efficiently.

3.1 Why Weak Pseudorandom Functions?

Pseudorandom functions turn a short random key into a virtually unbounded number of pseudorandom strings. Advanced cryptographic protocols often consume a tremendous amount of randomness, as randomness is unavoidable to hide secret data used in complex interactions, but also to guarantee many other standard security properties (e.g., catching cheaters with random challenges, authenticating data with MACs...). As such, that pseudorandom functions can be leveraged in some of these higher-level protocols is not too surprising.

What is perhaps less natural is the focus on *weak* pseudorandom functions: by definition, wPRFs require the participants to be given access to a trusted source of randomness in the first place, as pseudorandomness holds only when the inputs are truly random. However, it turns out that wherever PRFs are used as an inner component in a higher-level MPC protocol, wPRFs also suffice, for in all such applications, the inputs can be easily *preprocessed* (either by the party holding them, or by all participants when they are public). When one is allowed to preprocess inputs, there is an easy way to convert any wPRF into a PRF: given an arbitrary input x, hash x into $\mathsf{H}(x)$ using your favorite hash function (e.g., SHA-256) before feeding it to the wPRF. This general methodology, dubbed the hash-then-evaluate paradigm in [BCE+24], belongs to the large family of heuristic hash-based transforms that are provably secure when the hash is modeled as a random oracle (similar to the Fiat-Shamir transform, the Fischlin transform, the Fujisaki-Okamoto transform, and many more). In addition, the work of [BCE+24] showed that for several

[2] We do note, however, that the tools and results developped in these works have significantly influenced the design of shallow wPRFs that we will cover here. Therefore, on several occasions, we will refer to some of these early works to explain the security rationale underlying some candidates.

[3] That is, the usual "low-depth" classes found in complexity-theory textbooks: AC^0, $\mathsf{AC}^0[\oplus]$, ACC^0, TC^0, and related classes. They capture functions computable by constant-depth circuits operating with a certain set of gates, e.g., unbounded fan-in AND and OR gates for AC^0.

wPRF candidates (in fact, for essentially those covered in this SoK), instantiating H with a PRF whose key is publicly known yields a candidate PRF whose security guarantees are comparable to that of the original wPRF.

Therefore, if one is fine with relying on the random oracle methodology (for higher-level MPC protocols targeting real-world applications, this is usually the case) or on the security arguments outlined in [BCE+24], using a wPRF suffices. Furthermore, building shallow wPRFs turns out to be considerably easier than building shallow PRFs. This is a known general behavior both in theoretical cryptography[4] and in concrete candidates design. In particular, all candidate wPRFs covered in this SoK are provably not PRFs (they are easily broken given chosen queries), and constructing shallow PRFs of a comparable complexity is an open question (or, in the case of Goldreich's PRG, is known to be impossible [Vio13, AR16]).

On computational models and cost metrics. There are two types of restrictions that can be enforced by the computational model, that we will losely refer to as "hard" and "soft" restrictions:

- An outer scheme induces a computational model with a hard restriction when a measure of the circuits in the model (e.g., depth, gate type) cannot exceed a fixed bound. As a simple example, using a linearly-homomorphic encryption scheme (e.g., Paillier encryption [Pai99]) as outer component restricts the inner component (the primitive being homomorphically evaluated) to be computable by a circuit computing a linear combination over a ring; using instead the Boneh-Goh-Nissim encryption scheme [BGN05] restricts the inner component to be computable by a *depth-1* arithmetic circuit with product gates of indegree 2 (i.e., a quadratic polynomial).

- In other settings, the outer scheme induces a soft restriction on the circuits, where some measures (e.g., depth, gate count) can grow arbitrarily but are associated to a *cost* (an efficiency metric of the final protocol, typically computation, communication, or round complexity) that increases as the measure increases. A typical example would be the use of BGV-style fully-homomorphic encryption [Gen09, BGV12]: the inner component can be any Boolean circuit, but the computational cost (measured by noise growth, or number of necessary bootstrappings) grows with the multiplicative depth of the circuit.

Below, we cover the main target applications of these cryptographic constructions. For each target application, we provide some motivation, list the high-end cryptographic building blocks used as outer component to realize the application, and extract the computational model that captures the wPRFs that can be evaluated within each outer component, indicating whether it comes with soft or hard restrictions on the type of circuits that can be computed.

3.2 Oblivious Pseudorandom Functions

An oblivious pseudorandom function (OPRF) [FIPR05] for a pseudorandom function family $\mathsf{PRF} = (\mathsf{KeyGen}, \mathsf{Eval})$ is a two-party protocol where one party holds a key msk in the support of KeyGen, and the other party holds an input x to the PRF. At the end of the protocol, the party holding x should learn $\mathsf{Eval}(\mathsf{msk}, x)$ (and nothing more), while the party holding msk should not learn anything. OPRFs enjoy many applications in settings such as obliviously searching databases [FIPR05], private set intersection and

[4] For example, PRFs cannot exist in AC^0, even by using a constant number of xor and majority gates [Vio13], while wPRFs in AC^0 exist under standard assumptions such as factoring [Kha93].

pattern matching [HL08, KKRT16], password-protected secret sharing and password-authenticated key exchange [JKK14], single sign-on with privacy [BFH+20], cloud key management [JKR19], and many more—see [CHL22] for a recent survey on the topic.

3.2.1 From General Secure Computation

The simplest way to build an OPRF is to take a PRF family $\mathsf{PRF} = (\mathsf{KeyGen}, \mathsf{Eval})$, and to run a secure 2-party protocol that, on input msk from one party and x from the other, securely computes $\mathsf{Eval}(\mathsf{msk}, x)$. Secure 2-party computation (2PC) refers to a general set of techniques involving two participants with private inputs x and y who wish to compute a public function $f(x, y)$ on their joint private inputs (with the output being revealed to one participant, or both). The two main paradigms for modern secure 2-party computation are secret-sharing-based 2PC in the preprocessing model [GMW87, FKOS15, KOS16, DPSZ12, KPR18] and garbled-circuit-based secure computation [Yao86, KS08, ZRE15, WRK17b, WRK17a, DILO22, CWYY23], the latter being typically more expensive but requiring much less rounds of communication than the former, which can make it a better option over high-latency networks. Secure computation is a very active topic: there are many available protocols, and the cost metric can vary significantly from one protocol to the other. The metrics are typically of the "soft restriction" type: 2PC can theoretically evaluate any function. Nevertheless, the cost is mostly driven by the following properties.

Free Linear Operations. Linear operations (e.g., XORs) are typically "for free" in such systems, either because they can be performed locally by the participants in secret-sharing-based 2PC, or due to the free-XOR trick for garbled circuits [KS08]. In contrast, the number of nonlinear gates (e.g., AND gates) matters.

Low Depth. In secret-sharing-based 2PC, the depth of the circuit (the length of the largest path from an input to an output, counted ignoring the "free" linear gates) is a major measure of efficiency, as it translates to a lower bound on the number of messages the participants must exchange. For example, when written as a Boolean circuit with XOR gates and AND gates, the AES block cipher has depth 40 [ARS+15]. When the protocol takes place between distant machines, this incurs a significant latency: a single round trip between relatively close machines (e.g., in the same country) adds 50–100ms [DKLs19]. For distant machines, latency can range from 100–260ms, up to 0.5 seconds in some cases [DKLs19]. A 200ms roundtrip would result in 8s of delay for the AES circuit.

Correlated Randomness. In the preprocessing model, the participants initially run a secure protocol to distribute correlated randomness among them. Different types of correlated randomness can be used to implement efficiently some "complex" gates. Typical examples include gates converting between arithmetic values over different fields [BIP+18], matrix operations [BCG+20c], or some standard non-linear functions used in machine learning such as equality tests, threshold predicates, and ReLU [BGI19, BCG+21a].

As the methods and protocols for efficiently distributing complex forms of correlated randomness evolve rapidly, it is not feasible to pinpoint a specific computational model and cost metric that would faithfully capture the evolving state-of-the-art. Nevertheless, a good rule of thumb is that secure computations shine when evaluating *low-depth* circuits that minimize the number of non-linear gates, and can rely on certain types of more complex gates such as equality tests, greater-than predicates, ReLU, splines, or linear algebra, among others.

The performance gain obtained by using dedicated wPRFs to build OPRFs is significant. In Section 4, we introduce the alternating moduli paradigm which is argued by its authors

in [BIP+18, Table 2] to provide three orders of magnitude of improvement compared to OPRFs built on top of block ciphers such as AES, LowMC [ARS+15], or Rasta [DEG+18].

3.2.2 From TFHE

Fully-homomorphic encryption (FHE) [Gen09] is a natural high-end outer component for building round-optimal OPRFs: using FHE, one user would encrypt x, let the other user compute an encryption of $\mathsf{Eval}(\mathsf{msk}, x)$ homomorphically, and decrypt the result. Unfortunately, FHE is typically considerably slower than general secure computation, and a significant effort is being invested both in optimizing FHE and in optimizing primitives for homomorphic evaluation via FHE (in the setting of ciphers, standard proposals include LowMC [ARS+15], FLIP [MJSC16], Rasta [DEG+18], Kreyvium [CCF+16], or Transistor [BBB+25], among others). Among the best-performing FHE schemes is TFHE [CGGI20]. It features more compact ciphertexts and relatively fast linear operations, as well as a fast *functional bootstrapping* [MP20, Joy21] that takes only a few milliseconds. A simplified circuit model for TFHE is the following (also of the soft restriction type, as TFHE can in principle evaluate any function).

Underlying Algebra. The circuit operates over a small integer ring $\mathcal{Z}_n$ (e.g., $n = 2$, $n = 17$).

Noise. A quantity called "noise" is increased by every operation, and will lead to an incorrect decryption should it become too high. It is then crucial to prevent it from increasing too much.

Programmable Bootstrap (PBS). Bootstrapping is a "refreshing" operation that brings the noise back down to a given base level. In the case of TFHE, bootstrapping is "programmable", meaning that it is possible to evaluate an arbitrary function $f : \mathcal{Z}_n \to \mathcal{Z}_n$ implemented by a lookup table without increasing the cost of the bootstrap. This operation is by far the most expensive, so its total number must be minimized—and it is better if those that are used can be evaluated in parallel.

Free Linear Operations. The evaluation of linear gates of arbitrary indegree is virtually "free" in terms of time, but it incurs a slight increase in the noise.

Of course, a linear function over any integer ring cannot be a wPRF, hence any wPRF in the above circuit model must contain at least one functional gate. Therefore, a natural question to answer is the following.

Given a small integer n (e.g., $n \leq 17$), are there efficient wPRF candidates computable by arithmetic circuits over $\mathcal{Z}_n$ with additions, multiplications by a constant, and a single layer of indegree-1 programmable bootstrapping gates? What is the smallest number of PBS gates required for a secure wPRF to exist in this model?

Recently, the work of [ADDG24] identified that a wPRF of [BIP+18], while not originally designed with this circuit model in mind, is especially well-suited for evaluation within TFHE, as each wPRF evaluation can be computed using solely linear operations together with a *single* layer functional bootstrapping. In fact, it is not too hard to observe that the alternative design of [BIP+18], denoted *alternative mod-2/mod-3 wPRF* in this writeup, is a (single-bit output) wPRF that uses a *single* f-functional gate (we cover this wPRF in Section 4.1). This illustrates a general behavior, also observed in other contexts, that shallow wPRFs that optimize to the extreme in some circuit model tend to also perform well in different circuit models (in the context of block ciphers, a similar observation was made for Kreyvium [CCF+16] that performs well for TFHE in spite of being originally optimized for BGV-style FHE).

3.3 Pseudorandom Correlation Generators and Functions

Pseudorandom correlation generators (PCG) and pseudorandom correlation functions (PCF) are related cryptographic primitives introduced and studied in a recent line of work [BCGI18, BCG+19b, BCG+19a, BCG+20c, BCG+20a, CRR21, BCG+22, BCCD23]. They play an important role in modern secure computation protocols: secure computation considers a scenario where n parties, with respective secret inputs $(x_1, \ldots, x_n)$, wish to jointly compute $f(x_1, \ldots, x_n)$ for some public function f without revealing anything more than the result of the computation. Modern secure computation protocols [GMW87, FKOS15, KOS16, DPSZ12, KPR18] rely on *correlated randomness* to achieve high concrete efficiency: random strings are sampled according to a joint distribution ("correlated") and are securely distributed among the parties prior to the protocol. Securely generating these strings has long been a major bottleneck in these protocols. PCGs and PCFs enable multiple participants, given short correlated keys, to generate long correlated pseudorandom strings *without any communication*, effectively pushing the bulk of the correlated randomness distribution to an offline computation. Below, we will describe two recent paradigms for constructing PCGs and PCFs from weak pseudorandom functions. For both paradigms, and unlike Section 3.2, the metrics are of the "hard restriction" type: the class of circuits handled by the frameworks are highly specific types of very low-depth circuits that can only handle a limited set of gates.

3.3.1 From Function Secret Sharing

Most constructions of PCGs and PCFs (in fact, all those cited above) follow a common template that combines a cryptographic primitive called *function secret sharing* (FSS) with either a pseudorandom generator (for PCGs) or a wPRF (for PCFs). Roughly, an FSS for a function class $\mathcal{F}$ is a high-level cryptographic primitive that allows sharing any function $f \in \mathcal{F}$ into a pair of shares (f_0, f_1) such that two parties with a common input x and respective shares f_0, f_1 can compute $y_b = \mathsf{Eval}(f_b, x)$ for $b = 0, 1$ such that $y_0 + y_1 = f(x)$. In other words, an FSS allows sharing a function $f \in \mathcal{F}$ such that from the shares of f, two parties can obtain additive shares of $f(x)$ for any input x. At the heart of PCG and PCF constructions are variants of the following (semi-formal) theorem.

Theorem 1 (Theorem 5.3 in [BCG+20a])**.** *Let $\mathcal{R} = \{\mathcal{R}_n\}_{n\in\mathbb{N}}$ denote a family of commutative rings. Let $\mathsf{PRF} = (\mathsf{KeyGen}, \mathsf{Eval})$ denote a weak pseudorandom function and let $\mathcal{F}$ denote the family of all functions $x \mapsto c \cdot \mathsf{Eval}(\mathsf{msk}, x)$ for some constant c and PRF key msk. If there exists an FSS for $\mathcal{F}$, then there exists a PCF for the OT correlation.*

The "OT correlation" (short for oblivious transfer correlation) is, without going into details, the most standard correlation required by secure computation protocols. Hence, the theorem above says that PCFs for useful correlations can be constructed, provided that there are wPRFs that fit into a class of functions for which there exists an FSS scheme.[5] In turn, the class of functions for which we know *efficient* FSS schemes[6] is fairly limited [BGI16], and corresponds to the set of functions that are linear combinations of "interval functions", as formally defined below.

Definition 5. Fix a ring $\mathcal{R}$ and an integer $N \in \mathbb{N}$. Let $\mathcal{I} = \mathcal{I}_N(\mathcal{R}) \subset \mathcal{R}^N$ denote the set of *interval functions* $f_{a,b,\Delta} : [N] \to \mathcal{R}$, parameterized by $a, b \in [N]$ with $a \le b$, and $\Delta \in \mathcal{R}$, and defined as:

$$f_{a,b,\Delta}(x) = \begin{cases} \Delta & \text{if } x \in [a, b], \\ 0 & \text{otherwise.} \end{cases}$$

[5] More formally, we need *scalar multiples* of wPRFs to fit in the class, that is, functions of the form $x \to c \cdot \mathsf{Eval}(\mathsf{msk}, x)$ for some scalar c.

[6] Theoretically, there exist advanced constructions of FSS for all polynomial-time functions from strong forms of fully-homomorphic encryption [DHRW16] that are known to exist from the LWE assumption. However, these results are purely of theoretical interest.

For a given $t \in \mathbb{N}$, define $\mathcal{F}_{t,N}(\mathcal{R})$ as the set of functions of the form

$$(x_1, \ldots, x_t) \mapsto L(f_1(x_1), \ldots, f_t(x_t)),$$

where each $f_i \in \mathcal{I}$ and $L : \mathcal{R}^t \to \mathcal{R}$ is a linear function.

Equivalently, the class $\mathcal{F}_{t,N}(\mathcal{R})$ can be viewed as the class of functions computable by a depth-2 circuit with indegree-1 comparison gates on the bottom (that output a fixed payload if the input is above, or below, a given threshold) and a t-ary linear combination gate on top. In the following, we will slightly abuse the definition and view $\mathcal{F}_{t,N}(\mathcal{R})$ both as a class of functions and as the corresponding class of circuits. The following informal lemma summarizes what is known about the existence of efficient FSS schemes for the class $\mathcal{F}$ (a more formal statement can be found in [BGI16]).

Lemma 1 (Informal)**.** *Assume the existence of a length-doubling pseudorandom generator. There exists an FSS for the class $\mathcal{F}_{t,N}(\mathcal{R})$ where the shares (f_0, f_1) of a function $f \in \mathcal{F}_{t,N}(\mathcal{R})$ have size $t \cdot (\lambda \cdot \log N + \log |\mathcal{R}|)$ and the cost of evaluating a share on an input x is dominated by $2t(\log N + (\log |\mathcal{R}|)/\lambda)$ invocations of the PRG.*

In practice, the pseudorandom generator is typically instantiated using AES, and the cost of evaluating the FSS becomes dominated by $2t(\log N + (\log |\mathcal{R}|)/\lambda)$ calls to AES, which is extremly efficient (for moderate values of t) using modern hardware with support for AES instructions. FSS constructions for more complex classes are known, but they make a heavy use of public-key cryptography and are considerably less efficient. Therefore, past works on PCGs and PCFs have focussed on developping PRGs and wPRFs compatible with the FSS schemes promised by Lemma 1—that is, computable by a depth-2 circuit in the class $\mathcal{F}$ over a suitable ring (typically, the field $\mathbb{F}_{2^\lambda}$, where the reader can think of λ as being 128):

Are there weak pseudorandom functions computable by a circuit in $\mathcal{F}_{t,N}(\mathbb{F}_{2^\lambda})$ for $\lambda = 128$ and for some value $t \ll N$?

In the PCF-from-FSS compiler, N translates to an upperbound on the target number of samples from the correlation (for a wPRF, N must be superpolynomial), and t is a parameter that influences both the PCF key size and the evaluation time. The works of [BCG+20a, CD23, BCG+22] introduced respectively the VDLPN-wPRF, and the EALPN-wPRF, as candidate solutions to the above challenge (with a value t polylogarithmic in N). We cover these candidates in Section 6 and Section 7 respectively.

3.3.2 From Constrained Pseudorandom Functions

Recently, an alternative paradigm for building PCFs has emerged in [BCM+24, CDD+24]. This paradigm replaces the outer FSS component with a constrained pseudorandom function (CPRF): a pseudorandom function equipped with a special "constraining" algorithm that, on the input of a constraint C (a function with binary output from a target class of constraints $\mathcal{C}$) and a PRF key msk, outputs a *constrained key* k_C that allows evaluating the PRF on all inputs x such that $C(x) = 1$, but leaks no information about the PRF output when $C(x) = 0$. The PCFs built from this paradigm enjoy a number of appealing procedures compared to the traditional paradigm: most notably, they come with a *public key setup* (a one-round protocol for securely generating the PCF keys), while efficient (one-round or multi-round) key distribution protocols for earlier PCF constructions are still an open problem. At the heart of this alternative paradigm is the following theorem:

Theorem 2. *Let* $\mathsf{PRF} = (\mathsf{KeyGen}, \mathsf{Eval})$ *denote a weak pseudorandom function with binary output and let $\mathcal{F}$ denote the family of all functions $x \to \mathsf{Eval}(msk, x)$ and $x \to 1 - \mathsf{Eval}(\mathsf{msk}, x)$ for some PRF key* msk*. If there exists a CPRF for the class of constraints $\mathcal{F}$, then there exists a PCF for the OT correlation.*

The work of [BCM+24] additionally introduced an efficient CPRF construction where the cost of evaluating the CPRF (and therefore the PCF) is dominated by a single exponentiation over an elliptic curve (on a modern laptop and for the fastest OpenSSL curves, this can be done in about 8μs). This CPRF builds upon the seminal Naor-Reingold PRF [NR97], and is restricted to constraints computable by an *inner-product membership* (IPM) predicate.

Definition 6. Fix a security parameter λ, integers $B, n \in \mathbb{N}$, and let $g : \{0,1\}^\lambda \to [B]^n$ be a function and $S \subset \mathbb{N}$ be a finite set. Let $\mathcal{IPM} = \mathcal{IPM}_{B,n}(S, g)$ denote the set of *inner-product membership predicates over* S, i.e., the functions $P_y : [B]^n \to \{0,1\}$ for $y \in [B]^n$ defined as follows:

$$P_y : x \mapsto \begin{cases} 1 \text{ if } \langle g(x), y\rangle \in S & \triangleright \text{ The inner product is computed over } \mathbb{N} \\ 0 \text{ otherwise .} \end{cases}$$

In other words, after fixing some public "preprocessing" function g on the input, and a subset S, an IPM predicate is a depth-3 circuit with a g-gate at the bottom (that takes a single input x and has n outputs), an arbitrary n-ary linear combination gate in the middle, and an indegree-1 membership-test gate on top. A formal statement of the following informal lemma can be found in [BCM+24].

Lemma 2 (Informal)**.** *Assume the power-DDH assumption over a group $\mathbb{G}$ of prime order p. Then there exists a CPRF for the class of constraints $\mathcal{IPM} = \mathcal{IPM}_{B,n}(S, g)$ where the size of a constrained key is $(n + |S|) \cdot \log |\mathbb{G}|$, and the cost of evaluating the CPRF is dominated by n multiplications over $\mathcal{Z}_p$ and one exponentiation over $\mathbb{G}$.*

Above, the number of multiplications can be reduced to the Hamming weight of $g(x)$. Then, to obtain efficient PCFs, it remains to find wPRFs that can be computed by an inner-product membership predicate, called IPM-wPRF in [BCM+24]:

Are there weak pseudorandom functions computable by a circuit from $\mathcal{IPM}_{B,n}(S, g)$ where B is some integer bound and S is a small subset of integers?

In the PCF-from-CPRF compiler, a larger S translates to a larger key size (as it grows linearly with $|S|$). The work of [BCM+24] showed that two wPRFs from previous works are actually IPM-wPRFs: the Boneh-Ishai-Passelègue-Sahai-Wu, or alternating moduli wPRF [BIP+18] and the Goldreich-Applebaum-Raykov wPRF [Gol00, AR16]. We discuss the two candidates in Section 4 and Section 5, respectively.

3.4 VOLE-Based Zero-Knowledge Proofs

Zero-knowledge proofs (ZKP) are fundamental cryptographic primitives that allow proving properties about a system that depends on secret values without revealing these values. While they were mainly an object of theoretical interest after their introduction in [GMR89], four decades of research have led to a plethora of highly practical ZKP that are now routinely used in real-world systems, from authentication to anonymous blockchains and cryptocurrencies (a recent example is Google's wallet [Goond]).

Definition. An NP-language is a set of strings $\mathcal{L} \subset \{0,1\}^*$ such that there is an algorithm $\mathcal{R}_\mathcal{L}$ (the *relation*) that can verify proofs of membership to $\mathcal{L}$ in polynomial-time: a string x is in $\mathcal{L}$ if and only if there exists a *witness* w, of size polynomial in x, such that $\mathcal{R}_\mathcal{L}(x, w)$ returns "accept" in time polynomial in $|x|$. A zero-knowledge proof for an NP-language $\mathcal{L}$ is a two-party secure computation protocol between a prover and a verifier with the following characteristics:

- Both parties have as common input a word $x \in \mathcal{L}$, and the (honest) prover additionally holds a *witness* for x—that is, a string w such that $\mathcal{R}_{\mathcal{L}}(x, w) = 1$.
- At the end of the interaction, the verifier outputs 1 ("accept") or 0 ("reject").
- An honest prover (with a witness w) should always cause the verifier to accept. However, if $x \notin \mathcal{L}$ (hence no valid witness w exists), no prover can possibly cause the verifier to output 1 with non-negligible probability.
- Eventually, the proof is *zero-knowledge*, meaning that it leaks no information about w (beyond the fact that $x \in \mathcal{L}$). This is formalized by requiring that the view of the verifier (the transcript of its interaction with the prover) could have been simulated without the witness w, and the simulation is indistinguishable from the real transcript (hence, in essence, the verifier cannot tell the transcript apart from a transcript computed independently of w).

Efficient zero-knowledge proof systems. Modern zero-knowledge proof systems require tradeoffs and compromises. Typically, they will handle restricted types of algebraic statements [Sch90, GS08], or be fully generic and optimal on almost any aspect, but incur an often prohibitive computational overhead on the prover side [BCC+16, Gro16, AHIV17, BBB+18, BBHR18, BCR+19, MBKM19, Set20, CBBZ23] (the citations are a short sample from the vast SNARK/STARK ecosystem of proofs). Building on the seminal work of [IKOS07], a recent line of work has put forth *VOLE*[7]*-based zero-knowledge* as a promising middle ground [BCGI18, WYKW21, DIO21, YSWW21, BMRS21]: VOLE-based ZKP can handle arbitrary statements, have reasonable proof sizes (though higher than SNARKs/STARKs for large statements), and small prover costs, confering them an overall highly competitive scalability. VOLE-based ZKP are deployed in real-world applications (e.g., TLSNotary [tls]), and in commercial solutions (such as zkPass [zkp] or zkTLS [pri]).

A typical "pain point" of zero-knowledge proofs are *mixed statements*: proofs of statements that mix arithmetics over different structures (a typical example is neural network training, that often mixes arithmetic over large fields with Boolean operations). Mixed statements have been the focus of a significant attention, both in general secure computation [DSZ15, GYKW24, BCG+21a] and in zero-knowledge proofs [CGM16, BBMH+21, OKMZ24, ABBS25]. A recent work [ABBS25] observed that shallow wPRFs provide an ideal tool to handle mixed arithmetic in VOLE-based zero-knowledge proofs. At the high level, the key idea is the following: in VOLE-based zero-knowledge, the secret witness is authenticated to the verifier via an information-theoretic homomorphic MAC. Previous works had observed that when the secret witness must be used over two different structures, e.g., the field $\mathbb{F}_2$ and a larger field $\mathbb{F}_p$, it suffices to have pairs of identical bits MAC-authenticated both in $\mathbb{F}_2$ and $\mathbb{F}_p$ to securely "switch" between the two structures. The work of [ABBS25] observes that an efficient way to achieve this is by first authenticating a short wPRF key over both fields and proving the equality. This is a mixed statement, so it can be expensive, but it is crucially independent of the size of the full statement since the key is short: it will get amortized away. Then, identical *pseudorandom* bits are generated over both fields by evaluating the wPRF and proving (separately over each field) the correct evaluation. To make this procedure efficient, the key is to find a wPRF that admits short proofs of correct evaluation *both over* $\mathbb{F}_2$ *and* $\mathbb{F}_p$.

Cost model. The basic cost model of modern VOLE-based zero-knowledge proofs is the usual "pay-per-product" MPC cost model, inherited from Quicksilver [YSWW21] (the

[7] Short for Vector Oblivious Linear Evaluation: a mechanism that distributes vectors $\vec{u}, \vec{v}$ to the prover and $\Delta, \vec{u} + \Delta \cdot \vec{v}$ to the verifier, where Δ acts as a MAC to "authenticate" the vector $\vec{v}$.

underlying proof system used in these works): when proving that an arithmetic circuit C (with additions and multiplications over a field $\mathbb{F}$) evaluates to a certain value on a secret witness w, all linear operations (additions, multiplications by a constant) are for free (they do not incur any overhead in the proof size), while multiplications increase the proof size. Therefore, the wPRF must be of the standard "MPC-friendly" type: its circuit description must minimize the number of multiplications.

In the setting of mixed statements, however, a subtlety arises: given a statement mixing operations over two fields $(\mathbb{F}, \mathbb{F}')$, the same wPRF must be computed by an arithmetic circuit over each of these fields. One must therefore be careful, as a low multiplication-gate count over one field does not typically translate to a low multiplication-gate count over the other field—in fact, the opposite is often true: linear operations over a field $\mathbb{F}_p$ are highly nonlinear over a field $\mathbb{F}_q$ when p, q are coprime [Raz87, Smo87] (this behavior is actually a crucial component of the alternating moduli paradigm, that we cover in Section 4). However, when $\mathbb{F}, \mathbb{F}'$ are prime-order fields, a simple observation is that it suffices to have a wPRF computable by a circuit with a small number of multiplications *over the integers*[8]: the natural arithmetization of this circuit (replacing additions and multiplications over the integers with the field operations) yields a valid arithmetic circuit for the wPRF over any prime-order field. In turn, any wPRF computable by a *d-local* Boolean circuit (i.e., such that each output bit depends on at most d input bits) can be arithmetized into an arithmetic circuit over the integers with at most d multiplications. Combining these two observations, we get the following target cost model.

Underlying Algebra. The circuit operates over the Boolean field $\mathbb{F}_2$.

Locality. The circuit is "d-local" for some small quantity d: every output bit depends on at most d input bits.

The question below summarizes the quest for the best possible candidate:

Given a target bound $n(\lambda)$ on the key size (for a security parameter λ) and a target number $N(\lambda)$ of samples, what is the smallest d such that there exist weak pseudorandom functions with key size $n(\lambda)$ computable by a d-local Boolean circuit?

The study of local PRGs and wPRFs has a rich history in cryptography. We cover the main candidates in Section 5.

4 Alternating Moduli Constructions

In 2018, Boneh, Ishai, Passelègue, Sahai, and Wu introduced a new design paradigm for constructing wPRFs, which was later described in the literature as the *alternating moduli paradigm* [BIP+18]. Constructions based on this paradigm are low-depth, but still exhibit a high algebraic degree. They are mathematically simple, built by combining linear functions over different moduli.

The original paper [BIP+18] presented two such constructions: a primary one outputting elements in $\mathcal{Z}_3$, and an alternative design that ultimately attracted more attention from the community. This second construction, referred to as the *alternative mod-2/mod-3 weak PRF* and labeled as Construction 6.3 in the original paper, is the only one from this work that we describe below.

[8] The circuit operates directly on the input bits of the wPRF, and must output integers that are guaranteed to be bits as well.

4.1 Original Construction ([BIP+18])

Let n denote the key length and the input length. The alternative mod-2/mod-3 wPRF as defined in [BIP+18] is a function $F : \mathcal{Z}_2^n \times \mathcal{Z}_2^n \to \mathcal{Z}_2$, where the key space is $\mathcal{K} = \mathcal{Z}_2^n$, the domain is $\mathcal{X} = \mathcal{Z}_2^n$, and the output space is $\mathcal{Y} = \mathcal{Z}_2$. For a key $k \in \mathcal{Z}_2^n$, we use the notation $F_k(x)$ to represent the function $F(k, x)$.

In its original formulation, F_k is defined as follows:

$$F_k(x) := \left(\sum_{i=0}^{n-1} k_i x_i \mod 2 + \sum_{i=0}^{n-1} k_i x_i \mod 3 \right) \mod 2, \quad \text{for } x \in \mathcal{Z}_2^n. \tag{1}$$

As we can see, we first compute the inner product between the key k and the input x. Then, this inner product is evaluated both modulo 2 and modulo 3, and the two results are finally added modulo 2 to produce the final result.

Alternative representation. The above construction can be implemented by a depth-2 circuit: the first layer of this circuit consists of a MOD_2 gate and a MOD_3 gate, both connected to all inputs x_i such that $k_i = 1$ (the MOD_3 gate is additionally connected to a "constant-2": it checks whether $\sum_i k_i x_i + 2$ is divisible by 3), followed by a second layer with a single two-input MOD_2-gate (by inspection, $F_k(x) = 1$ if and only if $\sum_i k_i x_i$ is 0 mod 2 and 1 mod 3, or $\sum_i k_i x_i$ is 1 mod 2 and 0 or 2 mod 3).

However, as the designers point out, the above description can have the following alternative representation, that stems from the observation that $F_k(x) = 1$ if and only if the inner product $\langle k, x \rangle \mod 6 = \sum_{i=0}^{n-1} k_i x_i \mod 6 \in \{3, 4, 5\}$. For this, denote by $\lfloor \cdot \rceil_2 : \mathcal{Z}_6 \to \mathcal{Z}_2$ the rounding operator defined as:

$$\lfloor u \rceil_2 = \begin{cases} 0 & \text{if } u \in \{0, 1, 2\} \ , \\ 1 & \text{if } u \in \{3, 4, 5\} \ . \end{cases}$$

Then, following the above remark, F_k can alternatively be defined as

$$F_k(x) = \left\lfloor \sum_{i=0}^{n-1} k_i x_i \mod 6 \right\rceil_2 . \tag{2}$$

One of the security arguments put forward by the designers of the above construction is that it cannot be approximated by any low-degree polynomial. This is supported by the classical results of Razborov and Smolensky [Raz87, Smo87], which show that for two distinct primes p and q, the mod p function cannot be approximated by low-depth mod q circuits. While the moduli p and q can, *a priori*, be chosen arbitrarily (as long as they are distinct), the most natural and simplest choice is $p = 2$ and $q = 3$. Although the versatility of the construction allows for different choices tailored to specific applications (e.g., in [DMMS21, HMM+23], the authors choose $p = 2^{31} - 1$), the concrete security level may vary depending on the specific instantiation of p and q. Therefore, dedicated cryptanalysis is required to evaluate the security of the construction under different choices of moduli.

4.2 Variants and Generalizations

The alternating moduli design paradigm quickly gained popularity as an approach for constructing weak shallow PRFs. Several subsequent works proposed generalized or alternative versions of the construction presented in the previous subsection. In this part, we describe two of the most notable and interesting variants.

4.2.1 A Generalized Construction [DGH+21]

In 2021, Dinur, Goldfeder, Halevi, Ishai, Kelkar, Sharma, and Zaverucha revisited the alternative mod-2/mod-3 constructions from [BIP+18] and proposed a generalized version. Their focus was primarily on the alternative variant (the one presented in the previous subsection), as having an output over $\mathcal{Z}_2$ is more convenient in many applications. The proposed generalization extends the original design in two main directions. First, the secret key is no longer a vector in $\mathcal{Z}_2^n$, but a matrix $K \in \mathcal{Z}_2^{m\times n}$. Second, a public compression matrix $B \in \mathcal{Z}_2^{t\times m}$ is applied before producing the output. This generalized construction has, among others, the advantage of outputting multiple bits at once instead of a single bit, resulting in improved throughput.

More precisely, the proposed construction is parameterized by positive integers n, m, and t, with the constraints $m \geq n$ and $m \geq t$. These parameters respectively represent the input vector size, the size of the intermediate vector(s), and the output vector size. As in the original design, they are functions of a security parameter λ.

The variant described in [DGH+21] defines a function $F : \mathcal{Z}_2^{m\times n} \times \mathcal{Z}_2^n \to \mathcal{Z}_2^t$, where the key space is $\mathcal{K} = \mathcal{Z}_2^{m\times n}$, the input space is $\mathcal{X} = \mathcal{Z}_2^n$, and the output space is $\mathcal{Y} = \mathcal{Z}_2^t$. For a key $K \in \mathcal{K}$, we write $F_K(x)$ to denote $F(K, x)$.

The first step of the construction is to compute an intermediate vector w as follows:

$$w = \big((Kx \bmod 2) + ((Kx \bmod 3) \bmod 2)\big) \bmod 2 \ .$$

The final output is then computed as

$$F_K(x) = Bw \bmod 2 \in \mathcal{Z}_2^t \ ,$$

where $B \in \mathcal{Z}_2^{t\times m}$ is a public compression matrix, which can be chosen either uniformly at random or as a full-rank random circulant matrix. This matrix can also be seen as a generator matrix for a linear code that has high minimum distance. Its application permits to resolve correlation issues between the key and the output, that would for example exploit linear biases in the output. Moreover, introducing the matrix B helps prevent decoding-based attacks such as BKW [BKW03], leading to significantly better parameter choices—for instance, achieving exponential rather than sub-exponential security. In addition, K itself can be structured; this is precisely the approach taken in some instantiations where K is chosen to be circulant.

It is straightforward to verify that this construction is a generalization of the alternative scheme proposed in [BIP+18]. Specifically, the original candidate described in the previous section corresponds to a special case of the construction in [DGH+21], obtained by setting $m = 1$, $t = 1$, and choosing $B = 1$.

4.2.2 A Further Generalization [APRR24]

In 2024, Alamati, Policharla, Raghuraman, and Rindal proposed a further generalization of the constructions in [BIP+18] and [DGH+21]. Let n, m, and t be non-negative integers. Both the input x and the secret key k are elements of $\mathcal{Z}_2^n$. Let $A \in \mathcal{Z}_2^{m\times n}$ and $B \in \mathcal{Z}_3^{t\times m}$ be public, random matrices.

The construction, named the $(\mathbb{F}_2, \mathbb{F}_3)$-wPRF, proceeds in two steps. First, compute

$$z = k \odot x := (k_0x_0, k_1x_1, \ldots, k_{n-1}x_{n-1}) \in \mathcal{Z}_2^n \ .$$

Then, compute the output as

$$F_k(x) = B(Az \bmod 2) \bmod 3 \in \mathcal{Z}_3^t \ ,$$

where, as before, we write $F_k(x)$ for a key $k \in \mathcal{Z}_2^n$ to denote $F(k, x)$. As with the previous variants, the authors claim that the moduli 2 and 3 can be replaced with any distinct prime numbers p and q.

The core idea behind this generalization is to allow for a better mixing of the terms $k_i x_i$. In the previous alternating moduli constructions, these terms were combined in a highly structured manner. For example, in the construction from [DGH+21], the matrix-vector product Kx mixes the key and input variables in a very regular way. In contrast, the use of a random matrix A in this construction results in random linear combinations over the variables composing the shared input vector $k \odot x$. As a result, each product term $k_i x_i$ is reused multiple times throughout the computation, leading to a higher diffusion. On the other hand, as we briefly discuss in the next subsection, combining the key and input variables using an AND operation introduces a strong bias in the output, which can significantly weaken the security of the construction.

The authors also propose a variant, named the *reversed-moduli* or $(\mathbb{F}_3, \mathbb{F}_2)$-variant, that produces binary output. For this variant, the key would be an element of $\mathcal{Z}_3^n$, but in practice it is seen as an element of $\mathcal{Z}_2^n$. In this version, the vector z is first interpreted as an element of $\mathcal{Z}_3^n$. The function output is then computed as:

$$F_k(x) = B(Az \bmod 3) \bmod 2 \in \mathcal{Z}_2^t \ ,$$

where $A \in \mathcal{Z}_3^{m \times n}$ and $B \in \mathcal{Z}_2^{t \times m}$.

4.3 Security Arguments and Cryptanalysis

To date, two papers have analyzed the security of the original construction of [BIP+18] that permit to better estimate the security level and guide the choice of parameters.

First, in [CCKK21], Cheon, Cho, Kim, and Kim proposed a distinguishing attack with a complexity of $\mathcal{O}(2^{0.21n})$, where n is the input length, by exploiting a statistical weakness of the construction. The authors used the observation that the function F_k can be interpreted as operating over the space $\mathcal{Z}_6$. However, due to the binary nature of both the key k and the input x, the function does not uniformly cover the entire space $\mathcal{Z}_6$. This distinguisher requires however a large number of samples to succeed and performs better when the Hamming weight of the key is low. To mitigate this issue, the authors suggested using keys with high Hamming weight as a patch. For instance, they recommended using keys with a Hamming weight of 310 when $n = 384$.

Later, in [JMN23], Johansson, Meier, and Nguyen introduced a different distinguishing attack against the original alternating modulo construction of [BIP+18]. This attack has a better computational complexity than that of [CCKK21], achieving $\mathcal{O}(2^{0.166n})$, where n is the input length. Interestingly, unlike the attack in [CCKK21], this approach, which can be interpreted as a differential attack, performs better when the Hamming weight of the key is high. This attack also suggests that $n = 384$ does not potentially offer a sufficient security margin.

Recently, the security of the generalized construction introduced in [APRR24] was analyzed in [AR25]. The authors present a key-recovery attack with a complexity of $\mathcal{O}(2^{\lambda/2} \log_2 \lambda)$ where λ is the security parameter against the first (standard) variant described in that work, and an attack of complexity $\mathcal{O}(2^{0.84\lambda})$ against the reversed-moduli variant. Both attacks rely on a similar principle: they exploit the fact that key and input bits are combined using the AND operation, which causes any input bit multiplied by a zero key bit to have no influence on the output. This weakness leads to collisions after processing approximately $2^{\lambda/2}$ inputs, leading to a key-recovery attack. Note that this attack is not applicable to [DGH+21], as the first operation is a matrix–vector multiplication rather than a bit-by-bit product of the components of a vector (the key) with those of the input (x).

4.4 Use of the Alternating Moduli Construction within Cryptographic Protocols

The original alternating moduli construction [BIP+18] has been incorporated into at least two cryptographic protocols, each proposing a concrete, and basically identical, set of parameters. The approach of [BCM+24] for constructing Pseudorandom Correlation Functions (PCFs) for Oblivious Transfer (OT) correlations suggests using the construction with a key and input length of $n = 770$ to achieve a security level of $\lambda = 128$ bits. The choice of n was guided by taking into account the attacks of [CCKK21] and [JMN23]. Similarly, the framework proposed in [CDD+24] recommends using the same construction, but with a key and input length of $n = 768$ instead of $n = 770$, for the simple reason that 768 is a multiple of 32 and therefore more suitable for software implementations.

For other types of applications, the generalized construction from [DGH+21] has been used in re-keying protocols, such as those proposed in [DMMS21] and [HMM+23]. The core idea behind these approaches is to use the secret matrix as a master key, generate randomness within a secure chip, and treat the output as a secret ephemeral symmetric key—even in the presence of an attacker who can observe linear leakages from the output. The motivation is that masking the matrix-vector product is relatively efficient, especially when working over the prime field $\mathbb{F}_p$ with $p = 2^{31} - 1$. In the proposal from [DMMS21], the authors argue that modulus 3 is not a suitable choice, and instead suggest using a 4×4 matrix over $\mathbb{F}_{2^{31}-1}$. However, the security of this approach depends on the assumed leakage model, and under certain models it may not be secure. In [HMM+23], the authors provide a more detailed cryptanalysis by explicitly bounding the number of leakages accessible to an attacker. Their analysis supports the same parameter sets as proposed in [DMMS21].

More recently, [ADDG24] introduced a novel application domain for this shallow weak PRF by combining it with the TFHE scheme [CGGI20], a fully homomorphic encryption construction based on programmable bootstrapping to reduce noise. In this context, the authors target a security level of $\lambda = 128$ bits and set $n = 256$, interpreting the key as a square matrix of dimension n.

4.5 Concrete Targets for Cryptanalysis

In this section, we provide explicit parameter sets for the alternative mod-2/mod-3 construction [BIP+18] and for the generalization proposed in [DGH+21], both of which we consider to be as good targets for cryptanalysis. All parameter sets aim for a 128-bit security level.

For the first construction, the parameter $n = 384$ proposed in the original paper is now considered insecure, based on the cryptanalysis results summarized in Section 4.3. As discussed in Section 4.4, the use of $n = 768$ was later proposed in [CDD+24] as an alternative. However, this parameter may offer more than 128 bits of security, and a dedicated cryptanalysis is required to accurately estimate its concrete security level.

Concerning the generalized construction given in [DGH+21], its authors propose two parameter sets: one described as *aggressive* and the other as more *conservative*. For a target security level of λ bits, the aggressive setting uses $n = m = 2\lambda$, while the conservative one sets $n = m = 2.5\lambda$. In both cases, the output size is chosen as $t = \lambda / \log_2(3)$. For the standard 128-bit security level, this corresponds to $n = m = 256$ in the aggressive case and $n = m = 320$ in the conservative one, with $t \approx 81$ in both settings. The authors further impose a concrete upper bound of $N = 2^{40}$ on the number of samples available to an attacker under a single key.

Table 1 summarizes the proposed parameter sets, where N denotes the maximum number of queries allowed under a single key. The upper bound $N = 2^{44.5}$ corresponds to the data complexity of the attack presented in [JMN23], and is also referenced in Nguyen's thesis [Ngu25] (p. 110). The parameters proposed in [CDD+24] and [BCM+24] are directly

based on the results of [JMN23].

Table 1: Parameters for the alternative mod-2/mod-3 construction [BIP+18] and the [DGH+21] generalization, intended as targets for cryptanalysis.

Variant	Parameters	N	Reference
Section 4.1 ([BIP+18])	$n = 768$	$2^{44.5}$	[CDD+24, JMN23, Ngu25]
Section 4.2.1 ([DGH+21])	$n = m = 256$, $t = 81$	2^{40}	[DGH+21]
	$n = m = 320$, $t = 81$	2^{40}	

4.6 Implementation

The algorithmic description of the alternative mod-2/mod-3 wPRF (Section 4.1) and of the generalization given in [DGH+21] are provided in Figure 2.

Algorithm 1 The mod-2/mod-3 wPRF

Input: $k, x \in \mathcal{Z}_2^n$
$u \leftarrow \sum_{i=0}^{n-1} k_i \cdot x_i \mod 6$
if $u \in \{0, 1, 2\}$ **then**
  return 0
else
  return 1
end if

Algorithm 2 Mod-6 version of the [DGH+21] construction

Input: $K \in \mathcal{Z}_2^{m \times n}$, $x \in \mathcal{Z}_2^n$, $B \in \mathcal{Z}_2^{t \times m}$
Output: $y \in \mathcal{Z}_2^t$
$u \leftarrow Kx \mod 6 \in \mathcal{Z}_6^m$
for $i = 1$ to m **do**
  if $u_i \in \{3, 4, 5\}$ **then**
    $u_i \leftarrow 1$
  else
    $u_i \leftarrow 0$
  end if
end for
$y \leftarrow Bu \mod 2 \in \mathcal{Z}_2^t$
return y

Figure 2: Algorithms of the two alternating moduli constructions proposed as targets for cryptanalysis

5 Goldreich's Pseudorandom Generator

Twenty-five years ago, Goldreich asked the question of whether it is possible to securely construct a very simple function that is hard to invert [Gol00]. By *very simple*, we mean a function whose description is transparent and can be expressed using a concise, closed-form formula. This question led to the design of a minimalistic and elegant one-way function. Despite its simplicity, this function is conjectured to offer strong cryptographic guarantees, and has, over the years, become a central object of study in the theory of cryptography and complexity.

5.1 Original Reference

Goldreich's proposal for constructing a one-way function is based on a very simple and elegant idea. Let $n, m \in \mathbb{N}$, and let $(\sigma_i)_{1 \leq i \leq m}$ be a sequence of ordered subsets of $\{1, \ldots, n\}$, each of size $d(n)$. The parameter $d(n)$ is referred to as the *locality* of the construction. Let $P : \{0,1\}^{d(n)} \to \{0,1\}$ be a fixed Boolean predicate (i.e., a function on $d(n)$ input bits).

For any input $x \in \{0,1\}^n$, and for each $i \in \{1,\ldots,m\}$, define $x[\sigma_i] \in \{0,1\}^{d(n)}$ as the projection of x onto the coordinates specified by σ_i :

$$x[\sigma_i] = \left(x_{\sigma_i(1)},\ x_{\sigma_i(2)},\ \ldots,\ x_{\sigma_i(d(n))}\right).$$

The candidate one-way function $F : \{0,1\}^n \to \{0,1\}^m$ is then defined as:

$$F(x) = \left(P(x[\sigma_1]),\ P(x[\sigma_2]),\ \ldots,\ P(x[\sigma_m])\right).$$

We define the *stretch* $\mathsf{s} \geq 1$ as the number in the expression $m = \mathcal{O}(n^{\mathsf{s}})$, where m denotes the output length of the function. A fundamental question in the study of Goldreich's construction is: how large can the stretch be for a fixed locality parameter d, while still maintaining security?

Goldreich's one-way function is defined by a fixed predicate and a public collection of access patterns. It can be instantiated to yield various cryptographic primitives, such as pseudo-random functions [AR16] or pseudorandom generators with output size polynomial in the input (i.e., $m \in \mathcal{O}(n^{\mathsf{s}})$ for $\mathsf{s} > 1$), since the existence of local pseudorandom generators follows from the existence of one-way random local functions [App12].

5.2 Goldreich's One Way Function as a Weak PRF

Given a locality parameter $d(n)$, the standard way to select the subsets $(\sigma_i)_{i \leq m}$ is to independently pick m uniformly random size-$d(n)$ subsets of $\{1,\ldots,n\}$. We briefly overview the rationale behind this choice. It is convenient to view the ordered subsets as describing a bipartite graph with n nodes on the left (numbered from 1 to n), m nodes on the right (numbered from 1 to m). For each right node $i \leq m$, we draw an edge between i and all left nodes in σ_i; the resulting bipartite graph is right $d(n)$-regular: all its right nodes have degree exactly $d(n)$.

For Goldreich's proposal to be hard to invert, it is necessary that the underlying bipartite graph is sufficiently *expanding*[9]: independently of the predicate, whenever the bipartite graph is not sufficiently expanding, the candidate can be broken in polynomial-time via so-called *shrinking set attacks* [AIK08, Zic17]. In turn, being a good expander is conjectured to also be a *sufficient* property for Goldreich's proposal to be secure (assuming that a suitable predicate is used) [AR16]. While this conjecture was disproved in [OST19] when m is very large (superpolynomial in n) by exhibiting a specific expander bipartite graph for which Goldreich's construction fails to be secure, it remains plausible for smaller stretches, and for the vast majority of expanders.

In general, we do not know of *explicit* good bipartite expanders for Goldreich's construction. However, it is a well known application of the probabilistic method that a random $d(n)$-regular bipartite graph is a good expander with high probability: asymptotically, a random $d(n)$-regular bipartite graph fails to be a good expander with probability $1/n^{O(d(n))}$. Whenever $d(n)$ is a growing function of n, this translates to a negligible failure probability. In addition, the "neighbour function" of a random bipartite graph (the function N that maps a right node to the set of its neighbors) has high circuit complexity, which provably circumvents the attack of [OST19], making random bipartite graphs a well-motivated choice for Goldreich's construction. Concretely, given fixed parameters n, m, one can set a target failure probability ε and infer the locality d that will suffice to resist shrinking set attacks with probability $1 - \varepsilon$ over the choice of the random graph. A simple and concrete procedure to select parameters was outlined in Ünal's recent work [Üna23a].

Given a random bipartite graph and a suitable predicate, Goldreich's construction is widely conjectured to be not only one-way, but also pseudorandom: for a uniformly random

[9] A bipartite graph is (e,α)-expanding if for every subset $S \subseteq [n]$ with $|S| \leq e$, denoting $N(S) \subseteq [m]$ the set of all neighbors of the nodes in S, it holds that $|N(S)| > \alpha \cdot |S|$.

seed $x \in \{0,1\}^n$, the output sequence $y = F(x) \in \{0,1\}^m$ should be computationally indistinguishable from a uniformly random string of the same length. In fact, in some parameter regimes, the pseudorandomness of Goldreich's construction is *implied* by its one-wayness [AR16]. Hence, constructions based on Goldreich's template are generally referred to as Goldreich's *pseudorandom generator*. An additional benefit of using a random bipartite graph is that it provides an alternative view of Goldreich's PRG as a weak pseudorandom function: view the input x (the seed of the PRG) as a key k for the weak PRF, and view each random size-$d(n)$ subset σ_i as a random input to the wPRF (a random string can be parsed as the description of a random fixed-sized subset by fixing a suitable parsing method). Then, given a predicate P, the candidate wPRF becomes

$$F_x(\sigma) = P(x[\sigma]).$$

The proof that the one-wayness of Goldreich's proposal implies that the above function is a weak PRF (again, for suitable parameters and predicate) is from Applebaum and Raykov [AR16]. Therefore, the above weak PRF is sometimes referred to as the Goldreich-Applebaum-Raykov (GAR) weak PRF; see, e.g., [BCM+24, CDD+24].

5.3 Predicates for Goldreich's Proposal

While this construction may initially seem purely theoretical, its structure and constraints provide valuable insights for practical cryptographic applications. In particular, the *locality* parameter $d(n)$ serves as a measure of the circuit's arithmetic depth or of the complexity of the circuit implementing the functions, and plays an important role in balancing efficiency and security. One of the most compact instantiations of this proposal uses locality $d = 5$ and the predicate

$$P_5(x_1, x_2, x_3, x_4, x_5) = x_1 + x_2 + x_3 + x_4 x_5 \ .$$

More generally, natural candidates for the predicate P in Goldreich's construction include the classes of XOR-MAJ and XOR-THR functions. These predicates operate on disjoint subsets of the input: the elements of the first subset are combined using XOR, while those of the second are processed by a nonlinear function, such as a majority or threshold function. The outputs of these two components are then XORed to produce the final result.

Threshold function. Let $\mathsf{w}(x)$ denote the Hamming weight of a vector $x \in \{0,1\}^t$. For a given threshold $\mathsf{th} \in [1, t+1]$, the threshold function $\mathsf{THR}_{\mathsf{th}} : \{0,1\}^t \to \{0,1\}$ is defined as:

$$\mathsf{THR}_{\mathsf{th}}(x) = \begin{cases} 0 & \text{if } \mathsf{w}(x) < \mathsf{th}, \\ 1 & \text{otherwise.} \end{cases}$$

Majority function. The majority function $\mathsf{MAJ} : \{0,1\}^t \to \{0,1\}$ is the special case of the threshold function where the threshold is set to half the input length (rounded up):

$$\mathsf{MAJ}(x) = \mathsf{THR}_{\lceil t/2 \rceil}(x) \ .$$

The $\mathsf{XOR}_{\ell_1} - \mathsf{MAJ}_{\ell_2}$ predicate. Let $u = (u_1, \ldots, u_\ell) \in \{0,1\}^\ell$ be the input to the predicate, and let $\ell_1 + \ell_2 = \ell$ be a partition of the input. This predicate applies an XOR over the first ℓ_1 input bits and a majority over the remaining ℓ_2 bits, and XORs the two results:

$$P(u) = \mathsf{XOR}_{\ell_1} - \mathsf{MAJ}_{\ell_2}(u) = \mathsf{XOR}(u_1, \ldots, u_{\ell_1}) \oplus \mathsf{MAJ}(u_{\ell_1+1}, \ldots, u_\ell) \ .$$

A similar definition can be given for the $\mathsf{XOR}_{\ell_1} - \mathsf{THR}_{\mathsf{th},\ell_2}$ predicate. These predicates are the two most popular choices, as they offer both high algebraic immunity [Cou03, CM03] and a high resiliency order [Sie84]. Their cryptographic properties have been thoroughly analyzed in [Méa21, Méa22].

5.4 Cryptanalysis

5.4.1 Generic Cryptanalysis

Before focusing on the properties of specific predicates, it is useful to recall that some forms of generic cryptanalysis apply regardless of the exact function used. A generic cryptanalytic result—meaning it does not rely on specific properties of the predicate—was proposed by Bogdanov and Qiao [BQ09] in the case of linear stretch, i.e., when $m = \lambda n$ for some constant λ. They show that an adversary can succeed by finding a string x' that is close to the secret x in Hamming distance. This work led to a subexponential-time attack [App15], which runs in time $\mathsf{poly}(n) \cdot 2^{2\varepsilon n}$ if $m \in \Omega(n/\varepsilon^{2d})$.

Naturally, the choice of predicate plays a crucial role in resisting such attacks. At a minimum, the predicate should be non-degenerate: it must be unbiased and not linear. In the next section, we describe additional cryptographic criteria that a predicate should satisfy to ensure stronger resistance.

In particular, two main classes of attacks become relevant when the predicate can either be approximated by a linear function with fewer than $d(n)$ variables, or when it has low degree or low algebraic immunity.

5.4.2 Linearity Tests and Resiliency

In 2014, O'Donnell and Witmer [OW14] showed that if the predicate can be approximated by a function that depends on only $c < d(n)$ variables, then the pseudorandom generator can be broken whenever $m \in \Omega(n^{c/2} + n \log n)$. This line of attack was further analyzed in [App15].

This type of attack is particularly efficient when the predicate has a low *resiliency order* [Sie84], a property that quantifies its resistance to such approximations. A Boolean function is said to be t-resilient if it cannot be approximated by a Boolean function on t variables. For example, the predicate P_5 has resiliency order $t = 2$. In [OW14], the authors approximate P_5 using a function with $c = t + 1 = 3$ variables, which leads to an attack whenever the output length m exceeds $n^{1.5}$.

With this criterion and the above attack in mind, it is worth noting that any predicate with locality $d = 4$ would yield an insecure PRG. Indeed, Siegenthaler [Sie84] proved that for any Boolean function f on d variables, $\mathrm{res}(f) + \deg(f) \leq d - 1$. Consequently, when $d = 4$, one cannot simultaneously have a resiliency order greater than 2 and an algebraic degree greater than 2. This implies that the smallest viable locality is $d = 5$. Determining whether this choice leads effectively to a secure construction remains an important theoretical question.

5.4.3 Algebraic Attacks

Unsurprisingly, one of the most powerful classes of attacks on Goldreich's PRG is algebraic in nature. A straightforward observation is that the generator's output can be inverted in polynomial time whenever the output length satisfies $m \in \Omega(n^{\delta})$, where δ is the algebraic degree of the predicate P. Beyond this basic observation, more refined algebraic attacks have been proposed.

A first example appears in [AL16], where the authors introduce the notion of *bit-fixing degree* as an important criterion for resisting linear tests. This quantity refers to the highest degree that the predicate P can attain when some of its input bits are fixed.

Another important criterion discussed in the same work is the *algebraic immunity* of the predicate [Cou03, CM03], which measures the minimal degree of a nonzero Boolean function that annihilates P or $P \oplus 1$. Low algebraic immunity implies that high-degree relations can be transformed into lower-degree equations, potentially leading to more efficient attacks.

This is precisely why functions such as P_5, XOR-MAJ, and XOR-THR are often considered in this context: they offer both good algebraic immunity and high resiliency, as discussed in [Méa21, Méa22]. Another interesting, though theoretical, question is whether there exists a predicate that simultaneously achieves optimal resiliency order and algebraic immunity for any given locality d [DMR23].

Finally, the output of Goldreich's PRG is *a priori* defined over the field with two elements $\mathcal{Z}_2$. However, the construction can naturally be extended to larger alphabets, such as $\mathcal{Z}_q$, for an integer $q > 2$. In 2023, Akin Ünal proposed three distinguishing attacks [Ünal23b], each effective in different parameter regimes. More precisely, if the output length satisfies $m = n^{1+e}$, then the first attack achieves good distinguishing advantage with time and space complexity $n^{\mathcal{O}(n^{1-\frac{e}{\delta-1}})}$, where δ is the algebraic degree of the predicate. A second attack is applicable when the alphabet size satisfies $q \in \mathcal{O}(n^{1-\frac{e}{\delta-1}})$. Finally, a third attack targets the case where the locality is d and the alphabet size q is constant. In this setting, the attack has complexity $2^{\mathcal{O}(n^{1-\frac{e'}{(q-1)d-1}})}$, where $e' < e$, with a corresponding adjustment in the advantage of the distinguisher. In a follow-up work [Ünal23a], Ünal improved over these attacks, as well as over previous shrinking set attacks [AIK08, Zic17], by proposing a way to extend the outputs of $\mathcal{Z}_2$ into larger fields in order to find algebraic relations in the outputs. He also provided a general methodology to select parameters for Goldreich's proposal. Up to replacing the algebraic degree with the notion of algebraic immunity (to account for the existence of low-degree annihilators [AL16]), this is the methodology that was employed in recent works to select concrete parameters [BCM+24, CDD+24, ABBS25].

5.4.4 Instantiations and Concrete Cryptanalysis

Up to this point, all the works discussed above provide insights into whether a polynomial-time attack exists, and offer criteria for determining when the predicate used in Goldreich's construction leads to a secure or insecure instance. As previously mentioned, many modern constructions now aim to target specific applications and are instantiated with concrete parameters. In this context, we need analyses that directly address this challenge and ultimately help build confidence in concrete instantiations for very specific security levels (e.g., 128-bit security).

The first (but slightly different) family of constructions inspired by Goldreich's PRG is the FLIP family of stream ciphers [MJSC16], along with its follow-ups: FiLIP [MCJS19, HMR20] and Elisabeth-4 [HMS23]. However, these proposals suffer from certain weaknesses—primarily algebraic in nature—as highlighted in [DLR16, GBJR23]. In particular, addressing these vulnerabilities requires increasing the size of the secret key to achieve a satisfactory security level.

For constructions without fixed parameters, a guess-and-determine approach is very unlikely to yield a polynomial-time attack. However, it proves to be a powerful tool for designing subexponential-time attacks. More importantly, such approaches can significantly reduce the complexity of attacks when concrete parameters are fixed. This strategy was first applied in [CDM+18], where the "determine" phase consists of a simple Gaussian elimination step. The authors also performed experiments using Gröbner basis techniques. However, as is often the case with algebraic attacks, these methods did not allow the authors to prove or disprove the security of the construction at cryptographic parameter sizes.

Four years later, the above cryptanalysis was improved in [YGJL22] by making

adaptative guesses, which drastically reduce the number of required guesses. Building upon [CDM+18], the authors successfully broke the original parameter sets proposed for the predicate P_5, and provided new ones as open challenges. To date, these revised parameters remain unbroken. In the same work, Yang, Guo, Johansson, and Lentmaier also introduced a very interesting variation: the *guess-and-decode* strategy. While they also considered attacks on other predicates, no detailed analysis was provided in those cases, as each predicate requires a separate, in-depth study to evaluate its resistance to such methods.

5.4.5 Bit-Fixing Correlation Attack

While resiliency and algebraic immunity have long been regarded as the two most important criteria for selecting a secure predicate, recent work [FLLL24] has shown that these properties alone are not sufficient. In particular, for the XOR-MAJ and XOR-THR predicates—which exhibit good resiliency and algebraic immunity—the authors of [FLLL24] observed a new vulnerability: if certain input bits are guessed to be fixed to 0 or 1, the resulting output can be interpreted as a system of noisy linear equations, where the noise level is significantly lower than what is induced by the original function. This attack becomes increasingly effective as the number of input bits to the majority or threshold function grows. This article highlights the strength of such an approach and emphasizes the necessity of predicate-specific cryptanalysis. This work also naturally raises the question of identifying predicates that resist algebraic, linearity-based attacks and naturally guess-and-determine strategies at the same time.

Table 2 summarizes several known subexponential attacks. In this table, d denotes the locality of the PRG, s is the stretch—meaning that the output size satisfies $m = \mathcal{O}(n^{\mathsf{s}})$—and n is the size of the seed.

Table 2: Subexponential and concrete cryptanalysis. The first technique applies to any predicate with locality d, while the three others specifically target the P_5 predicate.

Technique	Complexity	Reference
Hamming-distance test (x' close to x)	$2^{\mathcal{O}(n^{1-\frac{\mathsf{s}-1}{2d}})}$	[App15, BQ09]
Guess-and-determine	$2^{\mathcal{O}(n^{2-\mathsf{s}})}$	[CDM+18]
Adaptive guess-and-determine	$\mathcal{O}(n^2 \cdot 2^{\frac{n^{2-\mathsf{s}}}{4}})$	[YGJL22]
Guess-and-decode	$\mathcal{O}(n^{\mathsf{s}} \cdot 2^{\frac{n^{2-\mathsf{s}}}{4}})$	[YGJL22]

5.5 Targets for Cryptanalysis

The high versatility of Goldreich's PRG raises several important questions. In particular, it remains unclear how to choose an optimal predicate that achieves the best possible stretch s for the smallest possible locality $d(n)$.

Arguably, as also discussed above, the simplest and most natural instantiation is based on the P_5 predicate, for which concrete parameter sets are proposed in [YGJL22] and [ABBS25].

For constructions based on XOR-MAJ predicates, several parameter sets have been proposed in [ABBS25] and [BCM+24]. However, some of these were later broken by the bit-fixing correlation attack introduced in [FLLL24]. For the affected instances, we propose new parameters. Additional instantiations have also been suggested in [CDD+24].

All these parameter sets are summarized in Table 3.

Table 3: Challenge parameter sets for Goldreich's PRG targeting a 128-bit security level.

Seed size (n)	Stretch (s)	Predicate (P)	Reference
1024	1.02	P_5	[YGJL22]
2048	1.10	P_5	
4096	1.19	P_5	
8192	1.19	P_5	[ABBS25]
1024	2	XOR_4-MAJ_7	
512	4.5	XOR_{10}-MAJ_{64}	[BCM+24, FLLL24]
894	3.4	XOR_{10}-MAJ_{64}	[CDD+24]
2048	3	XOR_{10}-MAJ_{64}	

5.6 Algorithmic Description

The PRG differs from the other shallow weak PRFs presented in this survey in that it does not rely on a secret key. Instead, the adversary is given the subsets σ_i, which are sampled uniformly at random from all subsets of size d. We denote A_d^n the set of all ordered subsets of $\mathcal{Z}_n$ of size d. These subsets play a similar role to the inputs x in the other constructions: they are public, but not adversarially chosen. The actual secret in this case is the seed, which was previously denoted by x. For clarity and consistency with the description of the other wsPRFs, we now refer to the seed as k, and the subsets as x.

Algorithm 3 The Goldreich's PRF/PRG
Parameters: n (seed size), m (output size), d (locality) and $P : \{0,1\}^d \to \{0,1\}$ (predicate).

Input: $k \in \{0,1\}^n$, $x \in (A_d^n)^m$
$y \leftarrow \varepsilon$ ▷ ε denotes the empty string of length 0
for $i = 1$ to m **do**
  $z_i \leftarrow \{k_{x_{i,j}}\}_{j\in\{1,\dots,d\}}$
  $y \leftarrow y \,||\, P(z_i)$
end for
return y

6 VDLPN

The candidates introduced in this section and in the next (Section 7) are specifically designed for use in constructions of pseudorandom correlation functions that follow the paradigm outlined in Section 3.3.1. These constructions rely on the class $\mathcal{F}_{t,N}(\mathbb{F}_{2^\lambda})$ of depth-2 circuits, consisting of indegree-1 "interval membership" gates at the bottom layer and a t-ary linear combination gate at the top, where N is the target number of samples and t is a parameter to be minimized. In fact, both candidate families lie within the more restricted class $\mathcal{F}_{t,N}(\mathbb{F}_2)$. Recall that Theorem 1 relies on function secret sharing for functions of the form $x \mapsto c \cdot \mathsf{Eval}(\mathsf{msk}, x)$. In constructions based on VDLPN and EALPN, the evaluation $\mathsf{Eval}(\mathsf{msk}, x)$ is performed over $\mathbb{F}_2$, and only the scalar multiplier c lies in the extension field $\mathbb{F}_{2^\lambda}$ (typically with $\lambda = 128$).

In this section, we introduce the VDLPN design. While we provide a direct description below—as a XOR of ANDs of XORs—this representation slightly obscures the connection to the class $\mathcal{F}_{t,N}(\mathbb{F}_2)$. To see the connection, observe that a point function (i.e., a function f_a that outputs 0 everywhere except at $f_a(a) = 1$) is a special case of an interval function where the interval reduces to a single point. Verifying that $f_a(x) = 1$ is equivalent to

checking whether $x = a$, which can be expressed as $\prod_{i=1}^{|x|}(x_i \oplus \bar{a}_i)$, where the $\bar{a}_i$ are the bitwise complements of the bits of a. Therefore, a "XOR of products of XORs" is simply a linear combination (over $\mathbb{F}_2$) of point functions—which, again, are special cases of interval functions.

6.1 Original Reference

The VDLPN design was proposed by Boyle, Couteau, Gilboa, Ishai, Kohl, and Scholl in 2020 [BCG+20b]. The main candidate introduced in their work, that we will refer to as VDLPN (for *variable-density learning parity with noise*), is defined as follows:

$$F_k(x) := \bigoplus_{i=1}^{D} \bigoplus_{j=1}^{w} \prod_{\ell=1}^{i} (x_{i,j,\ell} \oplus k_{i,j,\ell}), \tag{3}$$

where D and w are parameters that depend on the desired security level λ.

An interesting observation is that this function can also be interpreted as a sum of *triangular functions*. Given a sequence of variables $z = (z_1, z_2, z_3, \dots)$, a triangular function is defined as:

$$T(z) := z_1 \oplus z_2 z_3 \oplus z_4 z_5 z_6 \oplus \cdots,$$

where each term in the sum is a product of an increasing number of consecutive variables.

Using this definition, Equation (3) can be rewritten more compactly as:

$$F_k(x) = \bigoplus_{j=1}^{w} T(X_j \oplus K_j),$$

where X_j is the vector $(x_{i,j,\ell})$ with indices $1 \le i \le D$, $1 \le \ell \le i$, and similarly, K_j is the vector $(k_{i,j,\ell})$ over the same index set.

Example (toy instance of the main variant with $w = 2$, $D = 3$). For $w = 2$ and $D = 3$, $F_k(x)$ can be written as

$$F_k(x) = T(Z_1) \oplus T(Z_2),$$

where $T(Z_j) = z_1 \oplus z_2 z_3 \oplus z_4 z_5 z_6$, for $j = 1, 2$ and

$$Z_j = (x_{1,j,1} \oplus k_{1,j,1}, x_{2,j,1} \oplus k_{2,j,1}, x_{2,j,2} \oplus k_{2,j,2}, x_{3,j,1} \oplus k_{3,j,1}, x_{3,j,2} \oplus k_{3,j,2}, x_{3,j,3} \oplus k_{3,j,3}).$$

We provide also an alternative low-complexity variant of the above function, that was equally presented in [BCG+20b], which we now describe in detail. We will call this variant VDLPN*. Let w and D be two positive integers. Both the secret key $k \in \mathcal{Z}_2^{w \times D}$ and the input $x \in \mathcal{Z}_2^{w \times D}$ are viewed as binary matrices, represented as $(k_{j,\ell})_{1 \le j \le w, 1 \le \ell \le D}$ and $(x_{j,\ell})_{1 \le j \le w, 1 \le \ell \le D}$, respectively. The associated function $F_k^*(x)$ is then computed as:

$$F_k^*(x) := \bigoplus_{i=1}^{D} \bigoplus_{j=1}^{w} \prod_{\ell=1}^{i} (x_{j,\ell} \oplus k_{j,\ell}). \tag{4}$$

We observe that Eq. (4) can also be rewritten in a different form—not as a sum of classical triangular functions, but rather as a sum of *somewhat triangular functions* $T^*(z)$, defined as follows for a sequence of variables $z = (z_1, z_2, z_3, \dots)$:

$$T^*(z) := z_1 \oplus z_1 z_2 \oplus z_1 z_2 z_3 \oplus \cdots.$$

Specifically, we can express the function as:

$$F_k^*(x) = \bigoplus_{j=1}^{w} T^*(X_j \oplus K_j),$$

where X_j is the vector $(x_{j,\ell})$ for $1 \leq \ell \leq D$, and similarly, $K_j = (k_{j,\ell})$ over the same index range. In other words, this can be seen as a variant of a triangular function in which variables are reused in higher-degree monomials. This reuse introduces algebraic dependencies between the terms, which complicates attempts to formally prove security.

Example (toy instance of the low-complexity variant with $w = 2$, $D = 3$). In this case,

$$F_k^*(x) = \bigoplus_{j=1}^{2}(x_{j,1} \oplus k_{j,1}) \oplus (x_{j,1} \oplus k_{j,1})(x_{j,2} \oplus k_{j,2}) \oplus (x_{j,1} \oplus k_{j,1})(x_{j,2} \oplus k_{j,2})(x_{j,3} \oplus k_{j,3}).$$

6.2 The VDLPN′ variant

While [BCG+20a] provides a detailed preliminary security analysis of VDLPN, it is purely asymptotic. Couteau and Ducros [CD23] revisited the original analysis of [BCG+20a] to obtain more concrete bounds achieving provable security guarantees against certain classes of attacks. In the course of refining these bounds, they suggested a variant of the original design, dubbed VDLPN′, that enjoys the same applications but comes with much tighter security arguments. Intuitively, this is achieved by shaving the first few terms of the XOR $\bigoplus_{i=1}^{D}$ and replacing them with a random inner product with the key; this circumvent corner cases of the security analysis that caused the bounds to become loose due to bad events that had significant probability of ocurring at the lower levels. Concretely, VDLPN′ is defined as follows:

$$F_k(x) = \bigoplus_{m=1}^{r} x_{0,m} \cdot k_{0,m} \oplus \bigoplus_{i=5}^{D} \bigoplus_{j=1}^{w} \prod_{\ell=1}^{i} (x_{i,j,\ell} \oplus k_{i,j,\ell}) \,, \tag{5}$$

where the $(x_{0,m}, k_{0,m})$ are $2r$ additional random bits sampled as part of the input x and key K respectively (that is, $x = (x_{0,1}, \cdots, x_{0,r}, (x_{i,j,\ell})_{i\in[5,D],j\in[1,w],\ell\in[1,i]})$, and k has an identical distribution), (D, w) are parameters that depend on the desired security level λ, and $r = h(15/2^D) \cdot 2^D + \lambda$ ($h(x) = -x \log x(1-x) \log(1-x)$ denotes the binary entropy function).

The candidate VDLPN′ also admits a similar low-complexity variant as VDLPN, that we denote by VDLPN′*, where the triangular functions are replaced with "shaved" somewhat triangular functions. Denoting

$$T'_5(z) = z_1z_2z_3z_4z_5 \oplus z_1z_2z_3z_4z_5z_6 \oplus \cdots \,,$$

that is, the function T' stripped of its four first terms, we have

$$F_k(x) = \bigoplus_{m=1}^{r} x_{0,m} \cdot k_{0,m} \oplus \bigoplus_{j=1}^{w} \prod_{\ell=1}^{i} T'_5(X_j \oplus K_j). \tag{6}$$

6.3 Targets for Cryptanalysis

In the original paper [BCG+20b], the authors suggested the following parameters to instantiate both versions of VDLPN for achieving λ bits of security: $w = 1.5\lambda$, $D = w/4$, and a restriction on the number of queries allowed to an attacker, limited to $N = 2^D$. However, as noted in [CD23], these parameter choices were purely heuristic and not supported by concrete cryptanalysis.

This initial proposal was later revisited and refined in subsequent works. In [CD23], the authors conducted a detailed security analysis, allowing them to correct several errors from the original paper and to propose concrete parameters for achieving 128-bit security :

$w = 380$, $D = 30$, under the assumption that the adversary is limited to at most 2^D queries. No close formula was however provided. Finally, Ducros proposed in his thesis [Duc24] a refined method that permitted to propose several good sets of parameters (D, w). These parameters are given in Table 4 and we suggest the related instances as targets for cryptanalysis for both VDLPN and VDLPN′.

All variants are assumed to provide the same concrete security level: the differences between the variants amount to tradeoffs between efficiency (the low-complexity variant yields much better PCF constructions) and security arguments (the VDLPN′ variant enjoys much tighter security arguments), but the easiness (or hardness) of providing formal security arguments is not necessarily expected to translate to a concrete weakness (or strength) of the candidate: rather, it reflects limitations of the probability arguments used to rule out certain classes of attacks in settings where the random variables satisfy some complex dependencies.

Table 4: Parameter sets for the VDLPN, VDLPN*, VDLPN′ and VDLPN′* shallow wPRFs, intended as targets for cryptanalysis at the $\lambda = 128$ security level. The parameter r is used only for VDLPN′. Parameters taken from [CD23] and chosen to provably resist attacks from the linear test framework.

r	D	w	$N = 2^D$
391	20	293	2^{20}
466	25	336	2^{25}
541	30	380	2^{30}
616	35	421	2^{35}
691	40	461	2^{40}

6.4 Algorithmic Description

The algorithmic descriptions of the VDLPN shallow wPRF and its low-complexity variant VDLPN* are provided in Algorithm 4 and Algorithm 5, respectively. The algorithmic descriptions of the VDLPN′ shallow wPRF and its low-complexity variant VDLPN′* are provided in Algorithm 6 and Algorithm 7, respectively.

7 EALPN

This shallow wPRF was intended as a successor to the VDLPN wPRF: while it does not strictly subsume it (as VDLPN enjoys other applications), it provides a candidate in the class $\mathcal{F}_{t,N}(\mathbb{F}_2)$ that does not limit itself to point functions. Instead, it fully exploits the flexibility of the class to support general interval functions, yielding a significantly more efficient construction, where efficiency refers to the efficiency of the target high-level MPC application.

7.1 Original Reference

This shallow weak PRF was proposed by Boyle, Couteau, Gilboa, Ishai, Kohl, Resch and Scholl in 2022 [BCG+22]. It was subsequently analyzed in [RRT23]. Fix a target number of samples N. Define GT to be the greater-than predicate: $\mathsf{GT}(a, b) = 1$ iff $a > b$. Then,

$$F_k(x) = \bigoplus_{i=1}^{\ell} \mathsf{GT}(x_i, k_{x'_i}), \tag{7}$$

Algorithm 4 The VDLPN wPRF
Parameters: Integers w, D

Input: $x = (x_{i,j,\ell}) \in \mathcal{Z}_2^{D\times w\times D}$, $k = (k_{i,j,\ell}) \in \mathcal{Z}_2^{D\times w\times D}$
$y \leftarrow 0$
for $i = 1$ to D **do**
  for $j = 1$ to w **do**
    $t \leftarrow 1$
    for $\ell = 1$ to i **do**
      $t \leftarrow t \cdot (x_{i,j,\ell} \oplus k_{i,j,\ell})$
    end for
    $y \leftarrow y \oplus t$
  end for
end for
return y

Algorithm 5 The VDLPN* wPRF
Parameters: Integers w, D

Input: $x = (x_{j,\ell}) \in \mathcal{Z}_2^{w\times D}$, $k = (k_{j,\ell}) \in \mathcal{Z}_2^{w\times D}$
$y \leftarrow 0$
for $i = 1$ to D **do**
  for $j = 1$ to w **do**
    $t \leftarrow 1$
    for $\ell = 1$ to i **do**
      $t \leftarrow t \cdot (x_{j,\ell} \oplus k_{j,\ell})$
    end for
    $y \leftarrow y \oplus t$
  end for
end for
return y

Figure 3: Algorithmic description of the original VDLPN shallow wPRF and its low-complexity variant VDLPN*.

where ℓ and t are parameters that depend on the desired security level λ, and $x = (x_i, x'_i)_{i\leq\ell}$ is sampled from $([1, 5N/t] \times [1, t])^\ell$. We assume that N and ℓ are chosen such that t divides $5N$ and ℓ divides t. We let $w = 5N/t$.

7.2 Security Arguments and Cryptanalysis

The presentation of EALPN in this work differs significantly from the original description given in [BCG+22]. Before outviewing the security arguments for EALPN, we first clarify this discrepancy—highlighting that the alternative perspective adopted in [BCG+22] is particularly helpful for understanding the security rationale.

Equivalence with the original presentation. The original description of EALPN is as follows:

- Pick a random ℓ-*sparse* matrix $H \in \mathbb{F}_2^{N\times 5N}$ (i.e., each row of H is a random weight-ℓ vector[10].
- Sample a weight-t *regular noise vector* e as a concatenation of t length-w random unit vectors, that is e is drawn at random from $\{1, 2, 4, \ldots, 2^{w-1}\}^t$, and each integer 2^i is viewed as a unit vector over $\{0, 1\}^w$ via its binary encoding.
- Define Δ_{5N} to be the $5N$-by-$5N$ matrix filled with ones on and below the main diagonal.
- The EALPN assumption states that no efficient adversary can distinguish $(H, H \cdot \Delta_{5N} \cdot e)$ from (H, y) where $y \leftarrow\$ \mathbb{F}_2^N$.

With this alternative view, EALPN becomes the problem of decoding the *syndrome of a code* whose parity-check matrix is $H \cdot \Delta_{5N}$ and where the codeword is perturbed with a

[10]To simplify certain security arguments, rows are actually sampled from a Bernoulli distribution with parameter $p = \ell/5N$, where each entry is independently set to 1 with probability p, rather than being drawn as fixed-weight vectors. However, [BCG+22, Section 3.4] explicitly conjectures that both variants offer equivalent security guarantees. The fixed-weight variant remains the most convenient for the target application discussed in Section 3.3.1

Algorithm 6 The VDLPN′ wPRF
Parameters: Integers r, w, D

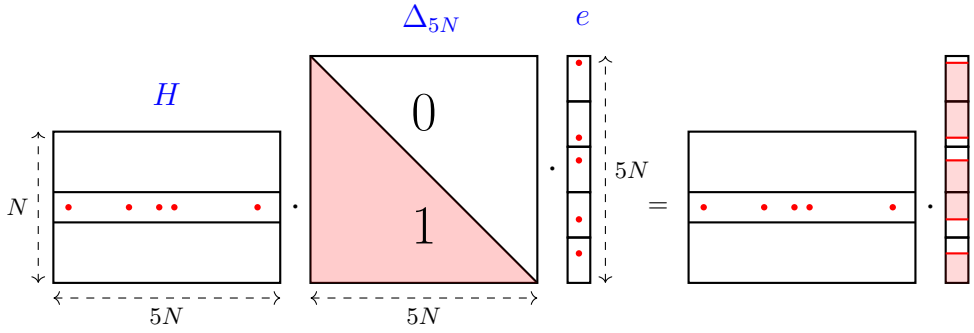
Input: $x = (x_{0,i}, x_{i,j,\ell}) \in \mathcal{Z}_2^s$, $k = (k_{0,i}, k_{i,j,\ell}) \in \mathcal{Z}_2^s$ with $s = r+(D-5)wD$
$y \leftarrow 0$
for $i = 1$ to r **do**
  $y \leftarrow y \oplus x_{0,i} \cdot k_{0,i}$
end for
for $i = 5$ to D **do**
  for $j = 1$ to w **do**
    $t \leftarrow 1$
    for $\ell = 1$ to i **do**
      $t \leftarrow t \cdot (x_{i,j,\ell} \oplus k_{i,j,\ell})$
    end for
    $y \leftarrow y \oplus t$
  end for
end for
return y

Algorithm 7 The VDLPN′* wPRF
Parameters: Integers r, w, D

Input: $x = (x_{0,i}, x_{j,\ell}) \in \mathcal{Z}_2^{r+w\times D}$, $k = (k_{0,i}, k_{j,\ell}) \in \mathcal{Z}_2^{r+w\times D}$
$y \leftarrow 0$
for $i = 1$ to r **do**
  $y \leftarrow y \oplus x_{0,i} \cdot k_{0,i}$
end for
for $i = 5$ to D **do**
  for $j = 1$ to w **do**
    $t \leftarrow 1$
    for $\ell = 1$ to i **do**
      $t \leftarrow t \cdot (x_{j,\ell} \oplus k_{j,\ell})$
    end for
    $y \leftarrow y \oplus t$
  end for
end for
return y

Figure 4: Algorithmic description of the VDLPN′ shallow wPRF and its low-complexity variant VDLPN′*.

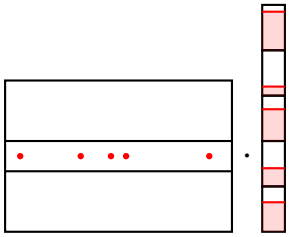
regular weight-w noise e. We first explain why this formulation is equivalent to the formula given in Section 7.1. First, observe that the left-product by Δ_{5N} is the *accumulation* function, that maps $(x_1, \cdots, x_{5N})$ to $(x_1, x_1 \oplus x_2, x_1 \oplus x_2 \oplus x_3, \cdots)$. As e is a weight-t regular vector (i.e., a concatenation of t unit vectors), $\Delta_{5N} \cdot e$ is an alternation of *bands of zeroes* and *bands of ones*. We can therefore rewrite the matrix equations as below: (red dots represent ones, white areas represent zeroes, light-red areas represent areas filled with ones):



In the above equation, the rightmost vector is formed by concatenating, for each block, a vector of the form $(0, \ldots, 0, 1, \ldots, 1)$ (a sequence of 0's followed by a sequence of 1's) when the block is in an odd position (1st, 3rd, 5th, etc.), and a vector of the form $(1, \ldots, 1, 0, \ldots, 0)$ (a sequence of 1's followed by a sequence of 0's) when the block is in an even position.

The next observation is that if we take the odd-numbered blocks of $f = \Delta_{5N} \cdot e$ and *turn them by 180°*, the assumption remains formally identical: indeed, we can divide each row of H in a length-$5N/w$ block and flip identically the odd-numbered blocks (rewriting them right-to-left). Denoting $\tilde{f}$ and $\tilde{H}$ the resulting vector and matrix respectively, we have $H \cdot f = \tilde{H} \cdot \tilde{f}$. But since the rows of H are random weight-ℓ vectors, the distribution

induced on $\tilde{H}$ is identical to the original distribution of H. Therefore, the assumption can be equivalently stated as: $(H, H \cdot \tilde{f})$ should be indistinguishable from $(H, y \leftarrow\!\$\, \mathbb{F}_2^N)$. Now, representing this last version visually:



Above, the vector is a now a concatenation of vectors of the form $(\vec{0}, \vec{1})$ (a vector of 0's concatenated with a vector of 1's). That is, compared to the previous equation, we "reversed" (rewrote in the reverse order) all the even-positioned block so that all blocks are now of the form $(\vec{0}, \vec{1})$ instead of alternating with vectors of the form $(\vec{1}, \vec{0})$. We can rewrite this matrix-vector product as follows: for each red dot in the matrix row, we look at the corresponding block of $\tilde{f}$ and check whether the red dot is before or after the red bar. That is, if the i-th red dot lands at position x_i in the x'_i-th block, and if we denote $k_{x'_i}$ the position of the red bar in this block, we compute

$$\mathsf{GT}(x_i, k_{x'_i})$$

for all of the ℓ nonzero entries of the row of H. Hence, sampling a random row h of H amounts to sampling ℓ pairs $(x_i, x'_i) \leftarrow\!\$\, [1, w] \times [1, t]$ and returning $\bigoplus_{i=1}^{\ell} \mathsf{GT}(x_i, k_{x'_i})$, which matches the description given in Section 7.1.

Security arguments Equipped with the above equivalence, we briefly overview the security argument of [BCG+22]. The argument investigates attacks in the *linear test framework* against the syndrome decoding problem. An attack against EALPN in this framework operates as follows:

- The attacker is given $M = H \cdot \Delta_{5N}$ (the parity-check matrix of the code) and produces a nonzero *test vector* v. This step can run in unbounded time. This *test vector* is chosen to maximize the bias of the distribution $M \cdot e$; a typical choice would be a minimal-weight nonzero vector in the kernel of M. Note that allowing unbounded adversaries here is equivalent to saying that we consider a worst-case choice of v: security in the linear test framework should hold for any possible choice of v.
- The noise vector e is sampled. The attacker's advantage is defined as the *bias* of the distribution induced by $v^\intercal \cdot M \cdot e$ (the bias is the distance to $1/2$ of the probability that $v^\intercal \cdot M \cdot e = 0$).

For λ bits of security, we aim for an advantage of $2^{-\lambda}$ in the above experiment. This abstract game captures the fact that most known attacks on code-based assumptions (such as information set decoding [Pra62], generalized birthday attacks [Wag02, Kir11], the BKW algorithm [BKW00, Lyu05], and many more) amount to (are formally equivalent to) computing a linear function of the syndrome (with coefficients computed solely from the code matrix) and trying to detect a bias in the output.

Then, if $M^\intercal$ generates a code with a large minimum distance (that is, the map $v \to M^\intercal v$ sends all nonzero vectors to vectors with a "large" Hamming weight, typically $\delta \cdot N$ for some constant δ), it is easy to see that no such attack can possibly succeed: if $v^\intercal \cdot M$ has Hamming weight δN, then $(v^\intercal \cdot M) \cdot e$ has bias approximately δ^t. Setting $t = \lambda / \log(\delta)$

suffices to get λ bits of security. Given this observation, proving resistance against most standard attacks on code-based assumptions amounts to proving a bound on the minimum distance of the code generated by $M^\intercal$, which [BCG+22] does:

Lemma 3. *Fix a parameter $\ell = \omega(\log N)$. The code generated by the rows of $M = H \cdot \Delta_{5N}$ has minimum distance at least $\Omega(N)$, with probability at least $1 - N^{-\omega(1)}$ over the choice of H.*

A more precisely quantified version of this lemma is given in [BCG+22, Theorem 3.10]. From there, the authors suggest conservative parameters, chosen to match the parameters for which the theorem guarantees 128 bits of security against linear tests, and aggressive parameters, based on the (empirically observed) assumption that the code generated by M has much better minimum distance than guaranteed by the (somewhat loose) analysis. However, the minimum distances heuristically assumed to hold have been invalidated in a follow-up work [RRT23]. Recent works have relied on an updated version of the agressive parameter set, where $\ell = 7$ is replaced with $\ell = 11$; we report this choice in the challenge parameter sets.

7.3 Parameters

Two sets of parameters are provided for a security level of $\lambda = 128$, one that can be considered as standard and an agressive one. These are summarized in Table 5.

Table 5: Suggested parameters for the EALPN wPRF as targets for cryptanalysis.

t	ℓ	N
85	$3\ln(5N)$	2^{45}
660	11	2^{45}

7.4 Algorithmic Description

Algorithm 8 provides an algorithmic description of the EALPN wPRF.

Algorithm 8 The EALPN wPRF
Parameters: integers N, t, ℓ, $w = \lceil \frac{5N}{t} \rceil$

Input: $x = ((x_1, x'_1), \ldots, (x_\ell, x'_\ell)) \in (\mathcal{Z}_w^* \times \mathcal{Z}_t^*)^\ell$; $k = (k_1, \ldots, k_\ell) \in (\mathcal{Z}_w)^\ell$
$r \leftarrow 0$
for $i = 1$ to ℓ **do**
  if $x_i \geq k_{x'_i}$ **then**
    $r \leftarrow r + 1$
  end if
end for
return $r \bmod 2$

8 Conclusion

The goal of this Systematization of Knowledge (SoK) article was to unify and present four distinct families of weak pseudo-random functions (wPRFs) under a common lens. These families of wPRFs have emerged from the theoretical cryptography community in response to various application needs. Despite their differing origins, all the constructions we examined share a common characteristic: they are shallow, typically consisting of a

Table 6: A summary of the wPRFs we present. The "†" symbol means the primitive is now considered unsafe, while "‡" indicates an "aggressive" set of parameters. The data limit for all such functions is at least 2^{40}, and the intended security level at least 128 bits. "NL op." refers to the non-linear operation.

Description			Spaces			max	NL op.	Best attack
Name	Use Case	Ref.	Input	Key	Output	data		
Alternating	MPC	[BIP+18]	$\mathcal{Z}_2^{384}$†	$\mathcal{Z}_2^{384}$†	$\mathcal{Z}_2$	$2^{44.5}$	mod	ad hoc
			$\mathcal{Z}_2^{768}$	$\mathcal{Z}_2^{768}$				
Moduli		[DGH+21]	$\mathcal{Z}_2^{256}$‡	$(\mathcal{Z}_2^{256})^{256}$‡	$\mathcal{Z}_2^{81}$	2^{40}		
			$\mathcal{Z}_2^{320}$	$(\mathcal{Z}_2^{320})^{320}$				
Goldreich's PRG	(T)FHE	[YGJL22]	A_5^{1024}	$\mathcal{Z}_2^{1024}$	$\mathcal{Z}_2$	1176	P_5	linearization, algebraic
			A_5^{2048}	$\mathcal{Z}_2^{2048}$	$\mathcal{Z}_2$	4389		
			A_5^{4096}	$\mathcal{Z}_2^{4096}$	$\mathcal{Z}_2$	19893		
		[ABBS25]	A_5^{8192}	$\mathcal{Z}_2^{8192}$	$\mathcal{Z}_2$	45387	P_5	
			A_5^{1024}	$\mathcal{Z}_2^{1024}$	$\mathcal{Z}_2$	2^{20}	XOR_4-MAJ_7	
		[BCM+24] [FLLL24]	A_5^{512}	$\mathcal{Z}_2^{512}$	$\mathcal{Z}_2$	$2^{40.5}$	XOR_{10}-MAJ_{64}	
		[CDD+24]	A_5^{894}	$\mathcal{Z}_2^{894}$	$\mathcal{Z}_2$	$2^{33.33}$	XOR_{10}-MAJ_{64}	
			A_5^{2048}	$\mathcal{Z}_2^{2048}$	$\mathcal{Z}_2$	2^{33}		
VDLPN, VDLPN′, VDLPN*, VDLPN′*	MPC	[CD23]	$\mathcal{Z}_2^{5860}$	$\mathcal{Z}_2^{5860}$	$\mathcal{Z}_2$	2^{20}	∧	Syndrome decoding
			$\mathcal{Z}_2^{8400}$	$\mathcal{Z}_2^{8400}$	$\mathcal{Z}_2$	2^{25}		
			$\mathcal{Z}_2^{11400}$	$\mathcal{Z}_2^{11400}$	$\mathcal{Z}_2$	2^{30}		
			$\mathcal{Z}_2^{14735}$	$\mathcal{Z}_2^{14735}$	$\mathcal{Z}_2$	2^{35}		
			$\mathcal{Z}_2^{18840}$	$\mathcal{Z}_2^{18840}$	$\mathcal{Z}_2$	2^{40}		
EALPN	MPC	[BCG+22]	$(\mathcal{Z}_{2^{40.9}}^* \times \mathcal{Z}_{85}^*)^{142}$	$(\mathcal{Z}_{2^{40.9}}^*)^{142}$	$\mathcal{Z}_2$	2^{45}	<	
			$(\mathcal{Z}_{2^{38}}^* \times \mathcal{Z}_{660}^*)^{11}$‡	$(\mathcal{Z}_{2^{38}}^*)^{11}$‡				

single round of computation. Moreover, they operate within weakened security models when compared to standard PRFs or PRPs, notably by restricting the attacker from choosing inputs. More importantly, the inputs to these primitives are sampled uniformly at random.

This uniform randomness of public inputs, combined with the small domain sizes typically proposed, makes differential cryptanalysis largely impractical in this context. As a result, the main attack strategies that remain applicable seem to be linear cryptanalysis, based on exploiting statistical biases in (linear combinations of) the output, and algebraic or combinatorial techniques. However, the field currently lacks a comprehensive understanding of how effective such attacks can be in practice.

In particular, a more comprehensive combinatorial analysis is needed to characterize the conditions under which an attack may succeed based on properties of the public inputs. Indeed, attacks may become feasible when public inputs fall within certain "bad" subsets. Identifying and understanding the structure of these subsets is an interesting and important research direction.

More broadly, there is a remarkable lack of systematic cryptanalysis in this weak-function setting. Security claims for these constructions often rely on heuristic or partial evidence, and the distinction between exponential and polynomial-time attacks becomes blurred once concrete parameters are fixed. Despite this, many of these primitives are being integrated into advanced cryptographic protocols and proposed for real-world deployment. We therefore encourage the symmetric cryptography community to analyze these constructions, and aim to facilitate this task by providing reference implementations for each of them. Targeted and rigorous cryptanalysis is essential before these primitives can be confidently used inside high-level protocols. Bridging the gap between theoretical proposals and concrete security evaluation is not just desirable, it is necessary to ensure the security of the advanced protocols that build upon them.

Acknowledgments

Christina Boura, Léo Perrin and Yann Rotella were partially supported through the France 2030 program under grant agreement No. ANR-22-PECY-0010 Cryptanalyse. Geoffroy Couteau acknowledges the support of the French Agence Nationale de la Recherche (ANR) under the France 2030 ANR Project ANR-22-PECY-003 SecureCompute. This work is supported by the European Research Council (ERC) grants OBELiSC (101115790), and ReSCALE (101041545).

References

[ABBS25] Amit Agarwal, Carsten Baum, Lennart Braun, and Peter Scholl. Low-bandwidth mixed arithmetic in vole-based ZK from low-degree PRGs. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part IV*, volume 15604 of *LNCS*, pages 396–426. Springer, 2025.

[ADDG24] Martin R. Albrecht, Alex Davidson, Amit Deo, and Daniel Gardham. Crypto dark matter on the torus - oblivious PRFs from shallow PRFs and TFHE. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VI*, volume 14656 of *LNCS*, pages 447–476. Springer, Cham, May 2024.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu,

editors, *ACM CCS 2017*, pages 2087–2104. ACM Press, October / November 2017.

[AIK08] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in nc 0. *Computational Complexity*, 17:38–69, 2008.

[AL16] Benny Applebaum and Shachar Lovett. Algebraic attacks against random local functions and their countermeasures. In Daniel Wichs and Yishay Mansour, editors, *48th ACM STOC*, pages 1087–1100. ACM Press, June 2016.

[App12] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 805–816. ACM Press, May 2012.

[App15] Benny Applebaum. The cryptographic hardness of random local functions – survey. Cryptology ePrint Archive, Report 2015/165, 2015.

[APRR24] Navid Alamati, Guru-Vamsi Policharla, Srinivasan Raghuraman, and Peter Rindal. Improved alternating-moduli PRFs and post-quantum signatures. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VIII*, volume 14927 of *LNCS*, pages 274–308. Springer, Cham, August 2024.

[AR16] Benny Applebaum and Pavel Raykov. Fast pseudorandom functions based on expander graphs. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part I*, volume 9985 of *LNCS*, pages 27–56. Springer, Berlin, Heidelberg, October / November 2016.

[AR25] Irati Manterola Ayala and Håvard Raddum. Zeroed out: Cryptanalysis of weak PRFs in alternating moduli. *IACR Trans. Symmetric Cryptol.*, 2025(2):1–15, 2025.

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 430–454. Springer, Berlin, Heidelberg, April 2015.

[BBB+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018.

[BBB+25] Jules Baudrin, Sonia Belaïd, Nicolas Bon, Christina Boura, Anne Canteaut, Gaëtan Leurent, Pascal Paillier, Léo Perrin, Matthieu Rivain, Yann Rotella, and Samuel Tap. Transistor: a TFHE-friendly stream cipher. In Seny Kamara and Yael Tauman Kalai, editors, *CRYPTO 2025, Part V*, volume 16004 of *Lecture Notes in Computer Science*, pages 530–565. Springer, 2025.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *ICALP 2018*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl, July 2018.

[BBMH+21] Carsten Baum, Lennart Braun, Alexander Munch-Hansen, Benoît Razet, and Peter Scholl. Appenzeller to brie: Efficient zero-knowledge proofs for mixed-mode arithmetic and Z2k. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 192–211. ACM Press, November 2021.

[BCC+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Berlin, Heidelberg, May 2016.

[BCCD23] Maxime Bombar, Geoffroy Couteau, Alain Couvreur, and Clément Ducros. Correlated pseudorandomness from the hardness of quasi-abelian decoding. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 567–601. Springer, Cham, August 2023.

[BCE+24] Chris Brzuska, Geoffroy Couteau, Christoph Egger, Pihla Karanko, and Pierre Meyer. Instantiating the hash-then-evaluate paradigm: Strengthening PRFs, PCFs, and OPRFs. In Clemente Galdi and Duong Hieu Phan, editors, *SCN 24, Part II*, volume 14974 of *LNCS*, pages 97–116. Springer, Cham, September 2024.

[BCG+19a] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Peter Rindal, and Peter Scholl. Efficient two-round OT extension and silent non-interactive secure computation. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 291–308. ACM Press, November 2019.

[BCG+19b] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 489–518. Springer, Cham, August 2019.

[BCG+20a] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Correlated pseudorandom functions from variable-density LPN. In *61st FOCS*, pages 1069–1080. IEEE Computer Society Press, November 2020.

[BCG+20b] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Correlated pseudorandom functions from variable-density LPN. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1069–1080. IEEE, 2020.

[BCG+20c] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators from ring-LPN. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 387–416. Springer, Cham, August 2020.

[BCG+21a] Elette Boyle, Nishanth Chandran, Niv Gilboa, Divya Gupta, Yuval Ishai, Nishant Kumar, and Mayank Rathee. Function secret sharing for mixed-mode and fixed-point secure computation. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 871–900. Springer, Cham, October 2021.

[BCG+21b] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Low-complexity weak pseudorandom functions in AC0[MOD2]. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 487–516, Virtual Event, August 2021. Springer, Cham.

[BCG+22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Correlated pseudorandomness from expand-accumulate codes. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 603–633. Springer, Cham, August 2022.

[BCGI18] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector OLE. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 896–912. ACM Press, October 2018.

[BCM+24] Dung Bui, Geoffroy Couteau, Pierre Meyer, Alain Passelègue, and Mahshid Riahinia. Fast public-key silent OT and more from constrained Naor-Reingold. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VI*, volume 14656 of *LNCS*, pages 88–118. Springer, Cham, May 2024.

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Cham, May 2019.

[BFH+20] Carsten Baum, Tore Kasper Frederiksen, Julia Hesse, Anja Lehmann, and Avishay Yanai. PESTO: Proactively secure distributed single sign-on, or how to trust a hacked server. In *2020 IEEE European Symposium on Security and Privacy*, pages 587–606. IEEE Computer Society Press, September 2020.

[BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing: Improvements and extensions. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1292–1303. ACM Press, October 2016.

[BGI19] Elette Boyle, Niv Gilboa, and Yuval Ishai. Secure computation with preprocessing via function secret sharing. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part I*, volume 11891 of *LNCS*, pages 341–371. Springer, Cham, December 2019.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 325–341. Springer, Berlin, Heidelberg, February 2005.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: New simple PRF candidates and their applications. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 699–729. Springer, Cham, November 2018.

[BKW00] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *32nd ACM STOC*, pages 435–440. ACM Press, May 2000.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[BMRS21] Carsten Baum, Alex J. Malozemoff, Marc B. Rosen, and Peter Scholl. Mac'n'cheese: Zero-knowledge proofs for boolean and arithmetic circuits with nested disjunctions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 92–122, Virtual Event, August 2021. Springer, Cham.

[BQ09] Andrej Bogdanov and Youming Qiao. On the security of Goldreich's one-way function. In Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. APPROX 2009, and RANDOM 2009*, volume 5687 of *LNCS*, pages 392–405. Springer, 2009.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 2–21. Springer, Berlin, Heidelberg, August 1991.

[CBBZ23] Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. HyperPlonk: Plonk with linear-time prover and high-degree custom gates. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 499–530. Springer, Cham, April 2023.

[CCF+16] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 313–333. Springer, Berlin, Heidelberg, March 2016.

[CCKK21] Jung Hee Cheon, Wonhee Cho, Jeong Han Kim, and Jiseung Kim. Adventures in crypto dark matter: Attacks and fixes for weak pseudorandom functions. In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 739–760. Springer, Cham, May 2021.

[CD23] Geoffroy Couteau and Clément Ducros. Pseudorandom correlation functions from variable-density LPN, revisited. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part II*, volume 13941 of *LNCS*, pages 221–250. Springer, Cham, May 2023.

[CDD+24] Geoffroy Couteau, Lalita Devadas, Srinivas Devadas, Alexander Koch, and Sacha Servan-Schreiber. QuietOT: Lightweight oblivious transfer with a public-key setup. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part II*, volume 15485 of *LNCS*, pages 197–231. Springer, Singapore, December 2024.

[CDM+18] Geoffroy Couteau, Aurélien Dupin, Pierrick Méaux, Mélissa Rossi, and Yann Rotella. On the concrete security of Goldreich's pseudorandom generator. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 96–124. Springer, Cham, December 2018.

[CGGI20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91, January 2020.

[CGM16] Melissa Chase, Chaya Ganesh, and Payman Mohassel. Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy preserving credentials. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 499–530. Springer, Berlin, Heidelberg, August 2016.

[CHL22] Sílvia Casacuberta, Julia Hesse, and Anja Lehmann. SoK: Oblivious pseudorandom functions. In *2022 IEEE European Symposium on Security and Privacy*, pages 625–646. IEEE Computer Society Press, June 2022.

[CM03] Nicolas Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 345–359. Springer, Berlin, Heidelberg, May 2003.

[Cou03] Nicolas Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 176–194. Springer, Berlin, Heidelberg, August 2003.

[CRR21] Geoffroy Couteau, Peter Rindal, and Srinivasan Raghuraman. Silver: Silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 502–534, Virtual Event, August 2021. Springer, Cham.

[CWYY23] Hongrui Cui, Xiao Wang, Kang Yang, and Yu Yu. Actively secure half-gates with minimum overhead under duplex networks. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 35–67. Springer, Cham, April 2023.

[DEG+18] Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Eik List, Florian Mendel, and Christian Rechberger. Rasta: A cipher with low ANDdepth and few ANDs per bit. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 662–692. Springer, Cham, August 2018.

[DGH+21] Itai Dinur, Steven Goldfeder, Tzipora Halevi, Yuval Ishai, Mahimna Kelkar, Vivek Sharma, and Greg Zaverucha. MPC-friendly symmetric cryptography from alternating moduli: Candidates, protocols, and applications. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 517–547, Virtual Event, August 2021. Springer, Cham.

[DHRW16] Yevgeniy Dodis, Shai Halevi, Ron D. Rothblum, and Daniel Wichs. Spooky encryption and its applications. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 93–122. Springer, Berlin, Heidelberg, August 2016.

[DILO22] Samuel Dittmer, Yuval Ishai, Steve Lu, and Rafail Ostrovsky. Authenticated garbling from simple correlations. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 57–87. Springer, Cham, August 2022.

[DIO21] Samuel Dittmer, Yuval Ishai, and Rafail Ostrovsky. Line-point zero knowledge and its applications. In Stefano Tessaro, editor, *ITC 2021*, volume 199 of *LIPIcs*, pages 5:1–5:24. Schloss Dagstuhl, July 2021.

[DKLs19] Jack Doerner, Yashvanth Kondi, Eysa Lee, and abhi shelat. Threshold ECDSA from ECDSA assumptions: The multiparty case. In *2019 IEEE Symposium on Security and Privacy*, pages 1051–1066. IEEE Computer Society Press, May 2019.

[DLR16] Sébastien Duval, Virginie Lallemand, and Yann Rotella. Cryptanalysis of the FLIP family of stream ciphers. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 457–475. Springer, Berlin, Heidelberg, August 2016.

[DMMS21] Sébastien Duval, Pierrick Méaux, Charles Momin, and François-Xavier Standaert. Exploring crypto-physical dark matter and learning with physical rounding. *IACR TCHES*, 2021(1):373–401, 2021.

[DMR23] Aurélien Dupin, Pierrick Méaux, and Mélissa Rossi. On the algebraic immunity - resiliency trade-off, implications for Goldreich's pseudorandom generator. *Des. Codes Cryptogr.*, 91(9):3035–3079, 2023.

[DPSZ12] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 643–662. Springer, Berlin, Heidelberg, August 2012.

[DS09] Yevgeniy Dodis and John P. Steinberger. Message authentication codes from unpredictable block ciphers. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 267–285. Springer, Berlin, Heidelberg, August 2009.

[DSZ15] Daniel Demmler, Thomas Schneider, and Michael Zohner. ABY - A framework for efficient mixed-protocol secure two-party computation. In *NDSS 2015*. The Internet Society, February 2015.

[Duc24] Clément Ducros. *Multiparty Computation from the Hardness of Coding Theory*. Ph.d. thesis, Université Paris Cité, 2024. https://hal.science/tel-04889558v1/document.

[FIPR05] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 303–324. Springer, Berlin, Heidelberg, February 2005.

[FKOS15] Tore Kasper Frederiksen, Marcel Keller, Emmanuela Orsini, and Peter Scholl. A unified approach to MPC with preprocessing using OT. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 711–735. Springer, Berlin, Heidelberg, November / December 2015.

[FLLL24] Ximing Fu, Mo Li, Shihan Lyu, and Chuanyi Liu. Bit-fixing correlation attacks on goldreich's pseudorandom generators. Cryptology ePrint Archive, Report 2024/1594, 2024.

[GBJR23] Henri Gilbert, Rachelle Heim Boissier, Jérémy Jean, and Jean-René Reinhard. Cryptanalysis of elisabeth-4. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part III*, volume 14440 of *LNCS*, pages 256–284. Springer, Singapore, December 2023.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.

[Gol00] Oded Goldreich. Candidate one-way functions based on expander graphs. *Electron. Colloquium Comput. Complex.*, TR00-090, 2000.

[Goond] Google. Google Wallet. https://wallet.google, n.d. Accessed: 2025-08-09.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Berlin, Heidelberg, May 2016.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Berlin, Heidelberg, April 2008.

[GYKW24] Radhika Garg, Kang Yang, Jonathan Katz, and Xiao Wang. Scalable mixed-mode MPC. In *2024 IEEE Symposium on Security and Privacy*, pages 523–541. IEEE Computer Society Press, May 2024.

[HL08] Carmit Hazay and Yehuda Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 155–175. Springer, Berlin, Heidelberg, March 2008.

[HMM+23] Clément Hoffmann, Pierrick Méaux, Charles Momin, Yann Rotella, François-Xavier Standaert, and Balazs Udvarhelyi. Learning with physical rounding for linear and quadratic leakage functions. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 410–439. Springer, Cham, August 2023.

[HMR20] Clément Hoffmann, Pierrick Méaux, and Thomas Ricosset. Transciphering, using FiLIP and TFHE for an efficient delegation of computation. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *INDOCRYPT 2020*, volume 12578 of *LNCS*, pages 39–61. Springer, Cham, December 2020.

[HMS23] Clément Hoffmann, Pierrick Méaux, and François-Xavier Standaert. The patching landscape of elisabeth-4 and the mixed filter permutator paradigm. In Anupam Chattopadhyay, Shivam Bhasin, Stjepan Picek, and Chester Rebeiro, editors, *INDOCRYPT 2023, Part I*, volume 14459 of *LNCS*, pages 134–156. Springer, Cham, December 2023.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007.

[JKK14] Stanislaw Jarecki, Aggelos Kiayias, and Hugo Krawczyk. Round-optimal password-protected secret sharing and T-PAKE in the password-only model. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 233–253. Springer, Berlin, Heidelberg, December 2014.

[JKR19] Stanislaw Jarecki, Hugo Krawczyk, and Jason K. Resch. Updatable oblivious key management for storage systems. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 379–393. ACM Press, November 2019.

[JMN23] Thomas Johansson, Willi Meier, and Vu Nguyen. Differential cryptanalysis of mod-2/mod-3 constructions of binary weak prfs. In *IEEE International Symposium on Information Theory, ISIT 2023, Taipei, Taiwan, June 25-30, 2023*, pages 477–482. IEEE, 2023.

[Joy21] Marc Joye. Guide to fully homomorphic encryption over the [discretized] torus. Cryptology ePrint Archive, Report 2021/1402, 2021.

[Kha93] Michael Kharitonov. Cryptographic hardness of distribution-specific learning. In *25th ACM STOC*, pages 372–381. ACM Press, May 1993.

[Kir11] Paul Kirchner. Improved generalized birthday attack. Cryptology ePrint Archive, Report 2011/377, 2011.

[KKRT16] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. Efficient batched oblivious PRF with applications to private set intersection. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 818–829. ACM Press, October 2016.

[KOS16] Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT: Faster malicious arithmetic secure computation with oblivious transfer. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 830–842. ACM Press, October 2016.

[KPR18] Marcel Keller, Valerio Pastro, and Dragos Rotaru. Overdrive: Making SPDZ great again. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 158–189. Springer, Cham, April / May 2018.

[KS08] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free XOR gates and applications. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 486–498. Springer, Berlin, Heidelberg, July 2008.

[KV94] Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

[LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2), 1988.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. 2005.

[Mat94] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 386–397. Springer, Berlin, Heidelberg, May 1994.

[MBKM19] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2111–2128. ACM Press, November 2019.

[MCJS19] Pierrick Méaux, Claude Carlet, Anthony Journault, and François-Xavier Standaert. Improved filter permutators for efficient FHE: Better instances and implementations. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, *INDOCRYPT 2019*, volume 11898 of *LNCS*, pages 68–91. Springer, Cham, December 2019.

[Méa21] Pierrick Méaux. On the fast algebraic immunity of threshold functions. *Cryptogr. Commun.*, 13(5):741–762, 2021.

[Méa22] Pierrick Méaux. On the algebraic immunity of direct sum constructions. *Discret. Appl. Math.*, 320:223–234, 2022.

[MJSC16] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 311–343. Springer, Berlin, Heidelberg, May 2016.

[MP20] Daniele Micciancio and Yuriy Polyakov. Bootstrapping in FHEW-like cryptosystems. Cryptology ePrint Archive, Report 2020/086, 2020.

[MV12] Eric Miles and Emanuele Viola. Substitution-permutation networks, pseudorandom functions, and natural proofs. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 68–85. Springer, Berlin, Heidelberg, August 2012.

[MW18] Daniele Micciancio and Michael Walter. On the bit security of cryptographic primitives. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 3–28. Springer, Cham, April / May 2018.

[Ngu25] Vu Nguyen. *Code-based Cryptography: Attacking and Constructing Cryptographic Systems*. Doctoral Thesis (compilation), Department of Electrical and Information Technology, 2025. https://lup.lub.lu.se/search/files/219155690/dissertation_VU_Final.pdf.

[NR95] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. In *36th FOCS*, pages 170–181. IEEE Computer Society Press, October 1995.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.

[NW88] Noam Nisan and Avi Wigderson. Hardness vs. randomness (extended abstract). In *29th FOCS*, pages 2–11. IEEE Computer Society Press, October 1988.

[OKMZ24] Michele Orrù, George Kadianakis, Mary Maller, and Greg Zaverucha. Beyond the circuit: How to minimize foreign arithmetic in ZKP circuits. Cryptology ePrint Archive, Report 2024/265, 2024.

[OST19] Igor Carboni Oliveira, Rahul Santhanam, and Roei Tell. Expander-based cryptography meets natural proofs. In Avrim Blum, editor, *ITCS 2019*, volume 124, pages 18:1–18:14. LIPIcs, January 2019.

[OW14] Ryan O'Donnell and David Witmer. Goldreich's PRG: evidence for near-optimal polynomial stretch. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 1–12. IEEE Computer Society, 2014.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, Berlin, Heidelberg, May 1999.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. 1962.

[pri] Primus Lab. https://docs.primuslabs.xyz/.

[Raz87] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

[RR94] Alexander A. Razborov and Steven Rudich. Natural proofs. In *26th ACM STOC*, pages 204–213. ACM Press, May 1994.

[RRT23] Srinivasan Raghuraman, Peter Rindal, and Titouan Tanguy. Expand-convolute codes for pseudorandom correlation generators from LPN. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 602–632. Springer, Cham, August 2023.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, New York, August 1990.

[Set20] Srinath Setty. Spartan: Efficient and general-purpose zkSNARKs without trusted setup. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 704–737. Springer, Cham, August 2020.

[Sie84] Thomas Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Trans. Inf. Theory*, 30(5):776–780, 1984.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In Alfred Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, pages 77–82. ACM Press, May 1987.

[SV12] Rocco A Servedio and Emanuele Viola. On a special case of rigidity. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 19, page 144. Citeseer, 2012.

[tls] TLSNotary. https://docs.tlsnotary.org/.

[Ünа23a] Akin Ünal. New baselines for local pseudorandom number generators by field extensions. Cryptology ePrint Archive, Report 2023/550, 2023.

[Ünа23b] Akin Ünal. Worst-case subexponential attacks on PRGs of constant degree or constant locality. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 25–54. Springer, Cham, April 2023.

[Val84] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vio13] Emanuele Viola. The communication complexity of addition. In Sanjeev Khanna, editor, *24th SODA*, pages 632–651. ACM-SIAM, January 2013.

[Wag02] David Wagner. A generalized birthday problem. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, Berlin, Heidelberg, August 2002.

[Wil13] Ryan Williams. Natural proofs versus derandomization. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 21–30. ACM Press, June 2013.

[WRK17a] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Authenticated garbling and efficient maliciously secure two-party computation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 21–37. ACM Press, October / November 2017.

[WRK17b] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Global-scale secure multiparty computation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 39–56. ACM Press, October / November 2017.

[WYKW21] Chenkai Weng, Kang Yang, Jonathan Katz, and Xiao Wang. Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. In *2021 IEEE Symposium on Security and Privacy*, pages 1074–1091. IEEE Computer Society Press, May 2021.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.

[YGJL22] Jing Yang, Qian Guo, Thomas Johansson, and Michael Lentmaier. Revisiting the concrete security of Goldreich's pseudorandom generator. *IEEE Trans. Inf. Theory*, 68(2):1329–1354, 2022.

[YSWW21] Kang Yang, Pratik Sarkar, Chenkai Weng, and Xiao Wang. QuickSilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 2986–3001. ACM Press, November 2021.

[Zic17] Lior Zichron. *Locally computable arithmetic pseudorandom generators*. PhD thesis, Master's thesis, School of Electrical Engineering, Tel Aviv University, 2017.

[zkp] zkPass. https://docsend.com/view/5wdg66beu7m95jf3.

[ZRE15] Samee Zahur, Mike Rosulek, and David Evans. Two halves make a whole - reducing data transfer in garbled circuits using half gates. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 220–250. Springer, Berlin, Heidelberg, April 2015.