# Pseudorandom Functions and Lattices

Abhishek Banerjee[1,⋆], Chris Peikert[1,⋆⋆], and Alon Rosen[2,⋆⋆⋆]

[1] Georgia Institute of Technology
[2] IDC Herzliya

**Abstract.** We give direct constructions of pseudorandom function (PRF) families based on conjectured hard lattice problems and learning problems. Our constructions are asymptotically efficient and highly parallelizable in a practical sense, i.e., they can be computed by simple, relatively *small* low-depth arithmetic or boolean circuits (e.g., in $\mathrm{NC}^1$ or even $\mathrm{TC}^0$). In addition, they are the first low-depth PRFs that have no known attack by efficient quantum algorithms. Central to our results is a new "derandomization" technique for the learning with errors (LWE) problem which, in effect, generates the error terms deterministically.

## 1 Introduction and Main Results

The past few years have seen significant progress in constructing public-key, identity-based, and homomorphic cryptographic schemes using lattices, e.g., [35, 33, 15, 14, 13, 1] and many more. Part of their appeal stems from provable worst-case hardness guarantees (starting with the seminal work of Ajtai [3]), good asymptotic efficiency and parallelism, and apparent resistance to quantum attacks (unlike the classical problems of factoring integers or computing discrete logarithms).

Perhaps surprisingly, there has been comparatively less progress in using lattices for *symmetric* cryptography, e.g., message authentication codes, block ciphers, and the like, which are widely used in practice. While in principle most symmetric objects of interest can be obtained generically from any one-way function, and hence from lattices, these generic constructions are usually very inefficient, which puts them at odds with the high performance demands of most applications. In addition, generic constructions often use their underlying primitives (e.g., one-way functions) in an inherently inefficient and *sequential*

⋆ Research supported in part by an ARC Fellowship and the second author's grants.
⋆⋆ This material is based upon work supported by the National Science Foundation under Grant CNS-0716786 and CAREER Award CCF-1054495, by the Alfred P. Sloan Foundation, by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under Contract No. FA8750-11-C-0098, and by BSF grant 2010296. The views expressed are those of the authors and do not necessarily reflect the official policy or position of the National Science Foundation, the Sloan Foundation, DARPA or the U.S. Government, or the BSF.
⋆⋆⋆ Research supported in part by BSF grant 2010296.

D. Pointcheval and T. Johansson (Eds.): EUROCRYPT 2012, LNCS 7237, pp. 719–737, 2012.

manner. While most lattice-based primitives are relatively efficient and highly parallelizable in a practical sense (i.e., they can be evaluated by small, low-depth circuits), those advantages are completely lost when plugging them into generic sequential constructions. This motivates the search for specialized constructions of symmetric objects that have comparable efficiency and parallelism to their lower-level counterparts.

Our focus in this work is on *pseudorandom function* (PRF) families, a central object in symmetric cryptography first rigorously defined and constructed by Goldreich, Goldwasser, and Micali ("GGM") [16]. Given a PRF family, most central goals of symmetric cryptography (e.g., encryption, authentication, identification) have simple solutions that make efficient use of the PRF. Informally, a family of deterministic functions is pseudorandom if no efficient adversary, given adaptive oracle access to a randomly chosen function from the family, can distinguish it from a uniformly random function. The seminal GGM construction is based generically on any length-doubling pseudorandom generator (and hence on any one-way function), but it requires $k$ *sequential* invocations of the generator when operating on $k$-bit inputs.

In contrast, by relying on a generic object called a "pseudorandom *synthesizer*," or directly on concrete number-theoretic problems (such as decision Diffie-Hellman, RSA, and factoring), Naor and Reingold [28, 29] and Naor, Reingold, and Rosen [30] (see also [23, 9]) constructed very elegant and more efficient PRFs, which can in principle be computed in parallel by low-depth circuits (e.g., in $\mathrm{NC}^2$ or $\mathrm{TC}^0$). However, achieving such low depth for their number-theoretic constructions requires extensive preprocessing and enormous circuits, so their results serve mainly as a proof of theoretical feasibility rather than practical utility.

In summary, thus far all parallelizable PRFs from commonly accepted cryptographic assumptions rely on exponentiation in large multiplicative groups, and the functions (or at least their underlying hard problems) can be broken by polynomial-time quantum algorithms. While lattices appear to be a natural candidate for avoiding these drawbacks, and there has been some partial progress in the form of *randomized* weak PRFs [4] and randomized MACs [34, 21], constructing an efficient, parallelizable (deterministic) PRF under lattice assumptions has, frustratingly, remained open for some time now.

### 1.1 Results and Techniques

In this work we give the first direct constructions of PRF families based on lattices, via the *learning with errors* (LWE) [35] and *ring*-LWE [25] problems, and some new variants. Our constructions are highly parallelizable in a *practical* sense, i.e., they can be computed by relatively *small* low-depth circuits, and the runtimes are also potentially practical. (However, their performance and key sizes are still far from those of heuristically designed functions like AES.) In addition, (at least) one of our constructions can be evaluated in the circuit class $\mathrm{TC}^0$ (i.e., constant-depth, poly-sized circuits with unbounded fan-in and threshold gates), which asymptotically matches the shallowest known PRF constructions based on the decision Diffie-Hellman and factoring problems [29, 30].

As a starting point, we recall that in their work introducing *synthesizers* as a foundation for PRFs [28], Naor and Reingold described a synthesizer based on a simple, conjectured hard-to-learn function. At first glance, this route seems very promising for obtaining PRFs from lattices, using LWE as the hard learning problem (which is known to be as hard as worst-case lattice problems [35, 31]). However, a crucial point is that Naor and Reingold's synthesizer uses a *deterministic* hard-to-learn function, whereas LWE's hardness depends essentially on adding *random, independent* errors to every output of a mod-$q$ "parity" function. (Indeed, without any error, parity functions are trivially easy to learn.) Probably the main obstacle so far in constructing efficient lattice/LWE-based PRFs has been in finding a way to introduce (sufficiently independent) error terms into each of the exponentially many function outputs, while still keeping the function deterministic and its key size a fixed polynomial. As evidence, consider that recent constructions of weaker primitives such as symmetric authentication protocols [18, 19, 20], randomized weak PRFs [4], and message-authentication codes [34, 21] from noisy-learning problems are all inherently *randomized* functions, where security relies on introducing fresh noise at every invocation. Unfortunately, this is not an option for deterministic primitives like PRFs.

*Derandomizing* LWE. To resolve the above-described issues, our first main insight is a way of partially "derandomizing" the LWE problem, i.e., generating the *errors* efficiently and deterministically, while preserving hardness. This technique immediately yields a deterministic synthesizer and hence a simple and parallelizable PRF, though with a few subtleties specific to our technique that we elaborate upon below.

Before we explain the derandomization idea, first recall the learning with errors problem $\mathsf{LWE}_{n,q,\alpha}$ in dimension $n$ (the main security parameter) with modulus $q$ and error rate $\alpha$. We are given many independent pairs $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where each $\mathbf{a}_i$ is uniformly random, and the $b_i$ are all either "noisy inner products" of the form $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q$ for a random secret $\mathbf{s} \in \mathbb{Z}_q^n$ and "small" random error terms $e_i \in \mathbb{Z}$ of magnitude $\approx \alpha q$, or are uniformly random and independent of the $\mathbf{a}_i$. The goal of the (decision) LWE problem is to distinguish between these two cases, with any non-negligible advantage. In the *ring*-LWE problem [25], we are instead given noisy ring products $b_i \approx a_i \cdot s$, where $s$ and the $a_i$ are random elements of a certain polynomial ring $R_q$ (the canonical example being $R_q = \mathbb{Z}_q[z]/(z^n + 1)$ for $n$ a power of 2), and the error terms are "small" in a certain basis of the ring; the goal again is to distinguish these from uniformly random pairs. While the dimension $n$ is the main hardness parameter, the error rate $\alpha$ also plays a very important role in both theory and practice: as long as the "absolute" error $\alpha q$ exceeds $\sqrt{n}$ or so, (ring-)LWE is provably as hard as approximating conjectured hard problems on (ideal) lattices to within $\tilde{O}(n/\alpha)$ factors in the worst case [35, 31, 25]. Moreover, known attacks using lattice basis reduction (e.g., [22, 37]) or combinatorial/algebraic methods [8, 5] require time $2^{\tilde{\Omega}(n/\log(1/\alpha))}$, where the $\tilde{\Omega}(\cdot)$ notation hides polylogarithmic factors in $n$. We emphasize that without the error terms, (ring-)LWE would become

trivially easy, and that all prior hardness results for LWE and its many variants (e.g., [35, 31, 17, 25, 34]) require random, independent errors.

Our derandomization technique for LWE is very simple: instead of adding a small random error term to each inner product $\langle \mathbf{a}_i, \mathbf{s} \rangle \in \mathbb{Z}_q$, we just deterministically *round* it to the nearest element of a sufficiently "coarse" public subset of $p \ll q$ well-separated values in $\mathbb{Z}_q$ (e.g., a subgroup). In other words, the "error term" comes solely from deterministically rounding $\langle \mathbf{a}_i, \mathbf{s} \rangle$ to a relatively nearby value. Since there are only $p$ possible rounded outputs in $\mathbb{Z}_q$, it is usually easier to view them as elements of $\mathbb{Z}_p$ and denote the rounded value by $\lfloor \langle \mathbf{a}_i, \mathbf{s} \rangle \rceil_p \in \mathbb{Z}_p$. We call the problem of distinguishing such rounded inner products from uniform samples the *learning with rounding* ($\mathsf{LWR}_{n,q,p}$) problem. Note that the problem can be hard only if $q > p$ (otherwise no error is introduced), that the "absolute" error is roughly $q/p$, and that the "error rate" relative to $q$ (i.e., the analogue of $\alpha$ in the LWE problem) is on the order of $1/p$.

We show that for appropriate parameters, $\mathsf{LWR}_{n,q,p}$ is at least as hard as $\mathsf{LWE}_{n,q,\alpha}$ for an error rate $\alpha$ proportional to $1/p$, giving us a worst-case hardness guarantee for LWR. In essence, the reduction relies on the fact that with high probability, we have $\lfloor \langle \mathbf{a}, \mathbf{s} \rangle + e \rceil_p = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_p$ when $e$ is small relative to $q/p$, while $\lfloor U(\mathbb{Z}_q) \rceil_p \approx U(\mathbb{Z}_p)$ where $U$ denotes the uniform distribution. Therefore, given samples $(\mathbf{a}_i, b_i)$ of an unknown type (either LWE or uniform), we can simply round the $b_i$ terms to generate samples of a corresponding type (LWR or uniform, respectively). (The formal proof is somewhat more involved, because it has to deal with the rare event that the error term changes the rounded value.) In the ring setting, the derandomization technique and hardness proof based on ring-LWE all go through without difficulty as well. While our proof needs both the ratio $q/p$ and the inverse LWE error rate $1/\alpha$ to be slightly super-polynomial in $n$, the state of the art in attack algorithms indicates that as long as $q/p$ is an integer (so that $\lfloor U(\mathbb{Z}_q) \rceil_p = U(\mathbb{Z}_p)$) and is at least $\Omega(\sqrt{n})$, LWR may be exponentially hard (even for quantum algorithms) for any $p = \mathrm{poly}(n)$, and superpolynomially hard when $p = 2^{n^\epsilon}$ for any $\epsilon < 1$.

We point out that in LWE-based cryptosystems, rounding to a fixed, coarse subset is a common method of removing noise and recovering the plaintext when decrypting a "noisy" ciphertext; here we instead use it to avoid having to introduce any random noise in the first place. We believe that this technique should be useful in many other settings, especially in symmetric cryptography. For example, the LWR problem immediately yields a simple and practical pseudorandom generator that does not require extracting biased (e.g., Gaussian) random values from its input seed, unlike the standard pseudorandom generators based on the LWE or LPN (learning parity with noise) problems. In addition, the rounding technique and its implications for PRFs are closely related to the "modulus reduction" technique from a concurrent and independent work of Brakerski and Vaikuntanathan [11] on fully homomorphic encryption from LWE, and a very recent follow-up work of Brakerski, Gentry, and Vaikuntanathan [10]; see Section 1.3 below for a discussion and comparison.

LWR-*based synthesizers and PRFs.* Recall from [28] that a pseudorandom *synthesizer* is a two-argument function $S(\cdot,\cdot)$ such that, for random and independent sequences $x_1,\ldots,x_m$ and $y_1,\ldots,y_m$ of inputs (for any $m = \text{poly}(n)$), the matrix of all $m^2$ values $z_{i,j} = S(x_i, y_j)$ is pseudorandom (i.e., computationally indistinguishable from uniform). A synthesizer can be seen as an (almost) length-*squaring* pseudorandom generator with good locality properties, in that it maps $2m$ random "seed" elements (the $x_i$ and $y_j$) to $m^2$ pseudorandom elements, and any component of its output depends on only two components of the input seed.

Using synthesizers in a recursive tree-like construction, Naor and Reingold gave PRFs on $k$-bit inputs, which can be computed using a total of about $k$ synthesizer evaluations, arranged nicely in only $\lg k$ levels (depth). Essentially, the main idea is that given a synthesizer $S(\cdot,\cdot)$ and two independent PRF instances $F_0$ and $F_1$ on $t$ input bits each, one gets a PRF on $2t$ input bits, defined as

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

The base case of a 1-bit PRF can trivially be implemented by returning one of two random strings in the function's secret key. Using particular $\text{NC}^1$ synthesizers based on a variety of both concrete and general assumptions, Naor and Reingold therefore obtain $k$-bit PRFs in $\text{NC}^2$, i.e., having circuit depth $O(\log^2 k)$.

We give a very simple and computationally efficient $\mathsf{LWR}_{n,q,p}$-based synthesizer $S_{n,q,p}\colon \mathbb{Z}_q^n \times \mathbb{Z}_q^n \to \mathbb{Z}_p$, defined as

$$S_{n,q,p}(\mathbf{a}, \mathbf{s}) = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_p. \tag{2}$$

(In this and what follows, products of vectors or matrices over $\mathbb{Z}_q$ are always performed modulo $q$.) Pseudorandomness of this synthesizer under LWR follows by a standard hybrid argument, using the fact that the $\mathbf{a}_i$ vectors given in the LWR problem are public. (In fact, the synthesizer outputs $S(\mathbf{a}_i, \mathbf{s}_j)$ are pseudorandom even given the $\mathbf{a}_i$.) To obtain a PRF using the tree construction of [28], we need the synthesizer output length to roughly match its input length, so we actually use the synthesizer $T_{n,q,p}(\mathbf{S}_1, \mathbf{S}_2) = \lfloor \mathbf{S}_1 \cdot \mathbf{S}_2 \rceil_p \in \mathbb{Z}_p^{n \times n}$ for $\mathbf{S}_i \in \mathbb{Z}_q^{n \times n}$. Note that the matrix multiplication can be done with a constant-depth, size-$O(n^2)$ arithmetic circuit over $\mathbb{Z}_q$. Or for better space and time complexity, we can instead use the ring-LWR synthesizer $S_{R,q,p}(s_1, s_2) = \lfloor s_1 \cdot s_2 \rceil_p$, since the ring product $s_1 \cdot s_2 \in R_q$ is the same size as $s_1, s_2 \in R_q$. The ring product can also be computed with a constant depth, size-$O(n^2)$ circuit over $\mathbb{Z}_q$, or in $O(\log n)$ depth and only $O(n \log n)$ scalar operations using Fast Fourier Transform-like techniques [24, 25].

Using the recursive input-doubling construction from Equation (1) above, we get the following concrete PRF with input length $k = 2^d$. Let $q_d > q_{d-1} > \cdots > q_0 \geq 2$ be a chain of moduli where each $q_j/q_{j-1}$ is a sufficiently large integer, e.g., $q_j = q^{j+1}$ for some $q \geq \sqrt{n}$. The secret key is a set of $2k$ matrices $\mathbf{S}_{i,b} \in \mathbb{Z}_{q_d}^{n \times n}$ for each $i \in \{1,\ldots,k\}$ and $b \in \{0,1\}$. Each pair $(\mathbf{S}_{i,0}, \mathbf{S}_{i,1})$ defines a 1-bit PRF $F_i(b) = \mathbf{S}_{i,b}$, and these are combined in a tree-like fashion according to Equation (1) using the appropriate synthesizers $T_{n,q_j,q_{j-1}}$ for $j = d,\ldots,1$. As a concrete example, when $k = 4$ (so $x = x_1 \cdots x_4$ and $d = 2$), we have

$$F_{\{\mathbf{S}_{i,b}\}}(x) \;=\; \Big\lfloor \lfloor \mathbf{S}_{1,x_1} \cdot \mathbf{S}_{2,x_2} \rceil_{q_1} \cdot \lfloor \mathbf{S}_{3,x_3} \cdot \mathbf{S}_{4,x_4} \rceil_{q_1} \Big\rceil_{q_0} . \tag{3}$$

(In the ring setting, we just use random elements $s_{i,b} \in R_{q_d}$ in place of the matrices $\mathbf{S}_{i,b}$.) Notice that the function involves $d = \lg k$ levels of matrix (or ring) products, each followed by a rounding operation. In the exemplary case where $q_j = q^{j+1}$, the rounding operations essentially drop the "least-significant" base-$q$ digit, so they can be implemented very easily in practice, especially if every $q_j$ is a power of 2. The function is also amenable to all of the nice time/space trade-offs, seed-compression techniques, and incremental computation ideas described in [28].

In the security proof, we rely on the conjectured hardness of $\mathsf{LWR}_{q_j,q_{j-1}}$ for $j = d, \ldots, 1$. The strongest of these assumptions appears to be for $j = d$, and this is certainly the case when relying on our reduction from $\mathsf{LWE}$ to $\mathsf{LWR}$. For the example parameters $q_j = q^{j+1}$ where $q \approx \sqrt{n}$, the dominating assumption is therefore the hardness of $\mathsf{LWR}_{q^{d+1},q^d}$, which involves a quasi-polynomial inverse error rate of $1/\alpha \approx q^d = n^{O(\lg k)}$. However, because the strongest assumptions are applied to the "innermost" layers of the function, it is unclear whether security actually *requires* such strong assumptions, or even whether the innermost layers need to be rounded at all. We discuss these issues further in Section 1.2 below.

*Degree-k synthesizers and shallower PRFs.* One moderate drawback of the above function is that it involves $\lg k$ levels of rounding operations, which appears to lower-bound the depth of any circuit computing the function by $\Omega(\lg k)$. Is it possible to do better?

Recall that in later works, Naor and Reingold [29] and Naor, Reingold, and Rosen [30] gave direct, more efficient number-theoretic PRF constructions which, while still requiring exponentiation in large multiplicative groups, can in principle be computed in very shallow circuit classes like $\mathrm{NC}^1$ or even $\mathrm{TC}^0$. Their functions can be interpreted as "degree-$k$" (or $k$-argument) synthesizers for arbitrary $k = \mathrm{poly}(n)$, which immediately yield $k$-bit PRFs without requiring any composition. With this in mind, a natural question is whether there are direct $\mathsf{LWE}$/$\mathsf{LWR}$-based synthesizers of degree $k > 2$.

We give a positive answer to this question. Much like the functions of [29, 30], ours have a subset-product structure. We have public moduli $q \gg p$, and the secret key is a set of $k$ matrices $\mathbf{S}_i \in \mathbb{Z}_q^{n \times n}$ (whose distributions may not necessarily be uniform; see below) for $i = 1, \ldots, k$, along with a uniformly random $\mathbf{a} \in \mathbb{Z}_q^n$.[1] The function $F = F_{\mathbf{a},\{\mathbf{S}_i\}} \colon \{0,1\}^k \to \mathbb{Z}_p^n$ is defined as the "rounded subset-product"

$$F_{\mathbf{a},\{\mathbf{S}_i\}}(x_1 \cdots x_k) = \left\lfloor \mathbf{a}^t \cdot \prod_{i=1}^{k} \mathbf{S}_i^{x_i} \right\rceil_p . \tag{4}$$

[1] To obtain longer function outputs, we can replace $\mathbf{a} \in \mathbb{Z}_q^n$ with a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for any $m = \mathrm{poly}(n)$.

The ring variant is analogous, replacing $\mathbf{a}$ with uniform $a \in R_q$ and each $\mathbf{S}_i$ by some $s_i \in R_q$ (or $R_q^*$, the set of invertible elements modulo $q$). This function is particularly efficient to evaluate using the discrete Fourier transform, as is standard with ring-based primitives (see, e.g., [24, 25]). In addition, similarly to [29, 30], one can optimize the subset-product operation via pre-processing, and evaluate the function in $\mathrm{TC}^0$. We elaborate on these optimizations in the full version of the paper [7].

For the security analysis of construction (4), we have meaningful security proofs under various conditions on the parameters and computational assumptions, including standard LWE. In our LWE-based proof, two important issues are the distribution of the secret key components $\mathbf{S}_i$, and the choice of moduli $q$ and $p$. For the former, it turns out that our proof needs the $\mathbf{S}_i$ matrices to be *short*, i.e., their entries should be drawn from the LWE error distribution. (LWE is no easier to solve for such short secrets [4].) This appears to be an artifact of our proof technique, which can be viewed as a variant of our LWE-to-LWR reduction, enhanced to handle adversarial queries. Summarizing the approach, define

$$G(x) = G_{\mathbf{a},\{\mathbf{S}_i\}}(x) := \mathbf{a}^t \cdot \prod_i \mathbf{S}_i^{x_i}$$

to be the subset-product function inside the rounding operation of (4). The fact that $F = \lfloor G \rceil_p$ lets us imagine adding *independent error terms* to each distinct output of $G$, but *only as part of a thought experiment* in the proof. More specifically, we consider a related *randomized* function $\tilde{G} = \tilde{G}_{\mathbf{a},\{\mathbf{S}_i\}} \colon \{0,1\}^k \to \mathbb{Z}_q^n$ that computes the subset-product by multiplying by each $\mathbf{S}_i^{x_i}$ in turn, but then also adds a fresh error term immediately following each multiplication. Using the LWE assumption and induction on $k$, we can show that the randomized function $\tilde{G}$ is itself pseudorandom (over $\mathbb{Z}_q$), hence so is $\lfloor \tilde{G} \rceil_p$ (over $\mathbb{Z}_p$). Moreover, we show that for every queried input, with high probability $\lfloor \tilde{G} \rceil_p$ coincides with $\lfloor G \rceil_p = F$, because $G$ and $\tilde{G}$ differ only by a cumulative error term that is small relative to $q$—this is where we need to assume that the entries of $\mathbf{S}_i$ are *small*. Finally, because $\lfloor \tilde{G} \rceil_p$ is a (randomized) pseudorandom function over $\mathbb{Z}_p$ that coincides with the deterministic function $F$ on all queries, we can conclude that $F$ is pseudorandom as well.

In the above-described proof strategy, the gap between $G$ and $\tilde{G}$ grows *exponentially* in $k$, because we add a separate noise term following each multiplication by an $\mathbf{S}_i$, which gets enlarged when multiplied by all the later $\mathbf{S}_i$. So in order to ensure that $\lfloor \tilde{G} \rceil_p = \lfloor G \rceil_p$ on all queries, our LWE-based proof needs both the modulus $q$ and inverse error rate $1/\alpha$ to exceed $n^{\Omega(k)}$. In terms of efficiency and security, this compares rather unfavorably with the quasipolynomial $n^{O(\lg k)}$ bound in the proof for our tree-based construction, though on the positive side, the direct degree-$k$ construction has better circuit depth. However, just as with construction (3) it is unclear whether such strong assumptions and large parameters are actually *necessary* for security, or whether the matrices $\mathbf{S}_i$ really need to be short.

In particular, it would be nice if the function in (4) were secure if the $\mathbf{S}_i$ matrices were *uniformly random* over $\mathbb{Z}_q^{n \times n}$, because we could then recursively compose the function in a $k$-ary tree to rapidly extend its input length.[2] It would be even better to have a security proof for a smaller modulus $q$ and inverse error rate $1/\alpha$, ideally both polynomial in $n$ even for large $k$. While we have been unable to find such a security proof under standard LWE, we do give a very tight proof under a new, interactive "*related samples*" LWE/LWR assumption. Roughly speaking, the assumption says that LWE/LWR remains hard even when the sampled $\mathbf{a}_i$ vectors are related by adversarially chosen subset-products of up to $k$ given random matrices (drawn from some known distribution). This provides some evidence that the function may indeed be secure for appropriately distributed $\mathbf{S}_i$, small modulus $q$, and large $k$. For further discussion, see Section 1.2, and for full details see the full version of the paper [7].

*PRFs via the GGM construction.* The above constructions aim to minimize the depth of the circuit evaluating the PRF. However, if parallel complexity is not a concern, and one wishes to minimize the total amount of work per PRF evaluation (or the seed length), then the original GGM construction with an LWR-based pseudorandom generator may turn out to be even more efficient in practice. We elaborate in the full version [7].

### 1.2 Discussion and Open Questions

The quasipolynomial $n^{O(\log k)}$ or exponential $n^{O(k)}$ moduli and inverse error rates used in our LWE-based security proofs are comparable to those used in recent fully homomorphic encryption (FHE) schemes (e.g., [14, 38, 12, 11, 10]), hierarchical identity-based encryption (HIBE) schemes (e.g., [13, 1, 2]), and other lattice-based constructions. However, there appears to be a major difference between our use of such strong assumptions, and that of schemes such as FHE/HIBE in the public-key setting. Constructions of the latter systems actually reveal LWE samples having very small error rates (which are needed to ensure correctness of decryption) to the attacker, and the attacker can break the cryptosystems by solving those instances. Therefore, the underlying assumptions and the true security of the schemes are essentially equivalent. In contrast, our PRF uses (small) errors *only as part of a thought experiment* in the security proof, not for any purpose in the operation of the function itself. This leaves open the possibility that our functions (or slight variants) remain secure even for much larger input lengths and smaller moduli than our proofs require. We conjecture that this is the case, even though we have not yet found security proofs (under standard assumptions) for these more efficient parameters. Certainly, determining whether there are effective cryptanalytic attacks is a very interesting and important research direction.

Note that in our construction (4), if we draw the secret key components from the uniform (or error) distribution and allow $k$ to be too large relative to $q$,

[2] Note that we can always compose the degree-$k$ function with our degree-2 synthesizers from above, but this would only yield a tree with 2-ary internal nodes.

then the function can become insecure via a simple attack (and our new "interactive" LWR assumption, which yields a tight security proof, becomes false). This is easiest to see for the ring-based function: representing each $s_i \in R_q$ by its vector of "Fourier coefficients" over $\mathbb{Z}_q^n$, each coefficient is 0 with probability about $1/q$ (depending on the precise distribution of $s_i$). Therefore, with noticeable probability the product of $k = O(q \log n)$ random $s_i$ will have all-0 Fourier coefficients, i.e., will be $0 \in R_q$. In this case our function will return zero on the all-1s input, in violation of the PRF requirement. (A similar but more complicated analysis can also be applied to the matrix-based function.) Of course, an obvious countermeasure is just to restrict the secret key components to be *invertible*; to our knowledge, this does not appear to have any drawback in terms of security. In fact, it is possible to show that the decision-(ring-)LWE problem remains hard when the secret is restricted to be invertible (and otherwise drawn from the uniform or error distribution), and this fact may be useful in further analysis of the function with more efficient parameters.

In summary, our work raises several interesting concrete questions, including:

- Is $\mathsf{LWR}_{n,q,p}$ really exponentially hard for $p = \text{poly}(n)$ and sufficiently large integer $q/p = \text{poly}(n)$? Are there stronger worst-case hardness guarantees than our current proof based on LWE?
- Is there a security proof for construction (4) (with $k = \omega(1)$) for poly$(n)$-bounded moduli and inverse error rates, under a non-interactive assumption?
- In construction (4), is there a security proof (under a non-interactive assumption) for uniformly random $\mathbf{S}_i$? Is there any provable security advantage to using *invertible* $\mathbf{S}_i$?
- Is there an efficient, low-depth PRF family based on the conjectured average-case hardness of the *subset-sum* problem?
- Our derandomization technique and LWR problem require working with moduli $q$ greater than 2. Is there an efficient, parallel PRF family based on the learning parity with noise (LPN) problem?

### 1.3 Other Related Work

In a companion paper [6], we have defined and implemented practically efficient variants of our functions, using rounding over the ring $\mathbb{Z}_N$ of integers modulo large powers-of-2 $N$. The functions have throughput and security levels that appear comparable with (or even exceed) those of ASE.

Most closely related to the techniques in this work are two very recent results of Brakerski and Vaikuntanathan [11] and a follow-up work of Brakerski, Gentry, and Vaikuntanathan [10] on fully homomorphic encryption from LWE. In particular, the former work includes a "modulus reduction" technique for LWE-based cryptosystems, which maps a large-modulus ciphertext to a small-modulus one; this induces a shallower decryption circuit and allows the system to be "bootstrapped" into a fully homomorphic scheme using the techniques of [14]. The modulus-reduction technique involves a rounding operation much like the one we use to derandomize LWE; while they use it on ciphertexts that are already

"noisy," we apply it to noise-free LWE samples. Our discovery of the rounding/derandomization technique in the PRF context was independent of [11]. In fact, the first PRF and security proof we found were for the direct degree-$k$ construction defined in (4), not the synthesizer-based construction in (3). As another point of comparison, the "somewhat homomorphic" cryptosystem from [11] that supports degree-$k$ operations (along with all prior ones, e.g., [14, 38]) involves an inverse error rate of $n^{O(k)}$, much like the LWE-based proof for our degree-$k$ synthesizer.

Building on the modulus reduction technique of [11], Brakerski *et al.* [10] showed that homomorphic cryptosystems can support certain degree-$k$ functions using a much smaller modulus and inverse error rate of $n^{O(\log k)}$. The essential idea is to interleave the homomorphic operations with several "small" modulus-reduction steps in a tree-like fashion, rather than performing all the homomorphic operations followed by one "huge" modulus reduction. This very closely parallels the difference between our direct degree-$k$ synthesizer and the Naor-Reingold-like [28] composed synthesizer defined in (3). Indeed, after we found construction (4), the result of [10] inspired our search for a PRF having similar tree-like structure and quasipolynomial error rates. Given our degree-2 synthesizer, the solution turned out to largely be laid out in the work of [28]. We find it very interesting that the same quantitative phenomena arise in two seemingly disparate settings (PRFs and FHE).

## 2 Preliminaries

For a probability distribution $X$ over a domain $D$, let $X^n$ denote its $n$-fold product distribution over $D^n$. The uniform distribution over a finite domain $D$ is denoted by $U(D)$. The discrete Gaussian probability distribution over $\mathbb{Z}$ with parameter $r > 0$, denoted $D_{\mathbb{Z},r}$, assigns probability proportional to $\exp(-\pi x^2/r^2)$ to each $x \in \mathbb{Z}$. It is possible to efficiently sample from this distribution (up to $\mathrm{negl}(n)$ statistical distance) via rejection [15].

For any integer modulus $q \geq 2$, $\mathbb{Z}_q$ denotes the quotient ring of integers modulo $q$. We define a 'rounding' function $\lfloor \cdot \rceil_p \colon \mathbb{Z}_q \to \mathbb{Z}_p$, where $q \geq p \geq 2$ will be apparent from the context, as

$$\lfloor x \rceil_p = \lfloor (p/q) \cdot \bar{x} \rceil \bmod p, \tag{5}$$

where $\bar{x} \in \mathbb{Z}$ is any integer congruent to $x \bmod q$. We extend $\lfloor \cdot \rceil_p$ componentwise to vectors and matrices over $\mathbb{Z}_q$, and coefficient-wise (with respect to the "power basis") to the quotient ring $R_q$ defined in the next subsection. Note that we can use any other common rounding method, like the floor $\lfloor \cdot \rfloor$, or ceiling $\lceil \cdot \rceil$ functions, in Equation 5 above, with only minor changes to our proofs. In implementations, it may be advantageous to use the floor function $\lfloor \cdot \rfloor$ when $q$ and $p$ are both powers of some common base $b$ (e.g., 2). In this setting, computing $\lfloor \cdot \rfloor_p$ is equivalent to dropping the least-significant digit(s) in base $b$.

*Learning With Errors.* We recall the learning with errors (LWE) problem due to Regev [35] and its ring analogue by Lyubashevsky, Peikert, and Regev [25]. For positive integer dimension $n$ (the security parameter) and modulus $q \geq 2$, a probability distribution $\chi$ over $\mathbb{Z}$, and a vector $\mathbf{s} \in \mathbb{Z}_q^n$, define the LWE distribution $A_{\mathbf{s},\chi}$ to be the distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing a vector $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ uniformly at random, an error term $e \leftarrow \chi$, and outputting $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q)$. We use the following "normal form" of the *decision*-LWE$_{n,q,\chi}$ problem, which is to distinguish (with advantage non-negligible in $n$) between any desired number $m = \text{poly}(n)$ of independent samples $(\mathbf{a}_i, b_i) \leftarrow A_{\mathbf{s},\chi}$ where $\mathbf{s} \leftarrow \chi^n \bmod q$ is chosen from the (folded) error distribution, and the same number of samples from the uniform distribution $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$. This form of the problem is as hard as the one where $\mathbf{s} \in \mathbb{Z}_q^n$ is chosen uniformly at random [4].

We extend the LWE distribution to $w \geq 1$ secrets, defining $A_{\mathbf{S},\chi}$ for $\mathbf{S} \in \mathbb{Z}_q^{n \times w}$ to be the distribution obtained by choosing $\mathbf{a} \leftarrow \mathbb{Z}_q^n$, an error vector $\mathbf{e}^t \leftarrow \chi^w$, and outputting $(\mathbf{a}, \mathbf{b}^t = \mathbf{a}^t\mathbf{S} + \mathbf{e}^t \bmod q)$. By a standard hybrid argument, distinguishing such samples (for $\mathbf{S} \leftarrow \chi^{n \times w}$) from uniformly random is as hard as decision-LWE$_{n,q,\chi}$, for any $w = \text{poly}(n)$. It is often convenient to group many (say, $m$) sample pairs together in matrices. This allows us to express the LWE problem as: distinguish any desired number of pairs $(\mathbf{A}^t, \mathbf{B}^t = \mathbf{A}^t\mathbf{S} + \mathbf{E} \bmod q) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times w}$, for the same $\mathbf{S}$, from uniformly random.

For certain moduli $q$ and (discrete) Gaussian error distributions $\chi$, the decision-LWE problem is as hard as the search problem, where the goal is to find $\mathbf{s}$ given samples from $A_{\mathbf{s},\chi}$ (see, e.g., [35, 31, 4, 26], and [27] for the mildest known requirements on $q$, which include the case where $q$ is a power of 2). In turn, for $\chi = D_{\mathbb{Z},r}$ with $r = \alpha q \geq 2\sqrt{n}$, the search problem is as hard as approximating worst-case lattice problems to within $\tilde{O}(n/\alpha)$ factors; see [35, 31] for precise statements.[3]

*Ring-LWE.* For simplicity of exposition, we use the following special case of the ring-LWE problem. (Our results can be extended to the more general form defined in [25].) Throughout the paper we let $R$ denote the cyclotomic polynomial ring $R = \mathbb{Z}[z]/(z^n + 1)$ for $n$ a power of 2. (Equivalently, $R$ is the ring of integers $\mathbb{Z}[\omega]$ for $\omega = \exp(\pi i/n)$.) For any integer modulus $q$, define the quotient ring $R_q = R/qR$. An element of $R$ can be represented as a polynomial (in $z$) of degree less than $n$ having integer coefficients; in other words, the "power basis" $\{1, z, \ldots, z^{n-1}\}$ is a $\mathbb{Z}$-basis for $R$. Similarly, it is a $\mathbb{Z}_q$-basis for $R_q$.

For a modulus $q$, a probability distribution $\chi$ over $R$, and an element $s \in R_q$, the ring-LWE (RLWE) distribution $A_{s,\chi}$ is the distribution over $R_q \times R_q$ obtained by choosing $a \in R_q$ uniformly at random, an error term $x \leftarrow \chi$, and outputting $(a, b = a \cdot s + x \bmod qR)$. The normal form of the decision-RLWE$_{R,q,\chi}$ problem is to distinguish (with non-negligible advantage) between any desired number $m = \text{poly}(n)$ of independent samples $(a_i, b_i) \leftarrow A_{s,\chi}$ where $s \leftarrow \chi \bmod q$, and

[3] It is important to note that the original hardness result of [35] for search-LWE is for a *continuous* Gaussian error distribution, which when rounded naïvely to the nearest integer does not produce a true discrete Gaussian $D_{\mathbb{Z},r}$. Fortunately, a suitable randomized rounding method does so [32].

the same number of samples drawn from the uniform distribution $U(R_q \times R_q)$. We will use the error distribution $\chi$ over $R$ where each coefficient (with respect to the power basis) is chosen independently from the discrete Gaussian $D_{\mathbb{Z},r}$ for some $r = \alpha q \geq \omega(\sqrt{n \log n})$.

For a prime modulus $q = 1 \bmod 2n$ and the error distribution $\chi$ described above, the decision-RLWE problem is as hard as the search problem, via a reduction that runs in time $q \cdot \mathrm{poly}(n)$ [25]. In turn, the search problem is as hard as quantumly approximating worst-case problems on ideal lattices.[4]

## 3 The Learning with Rounding Problem

We now define the "learning with rounding" (LWR) problem and its ring analogue, which are like "derandomized" versions of the usual (ring)-LWE problems, in that the error terms are chosen deterministically.

**Definition 1.** *Let $n \geq 1$ be the main security parameter and moduli $q \geq p \geq 2$ be integers.*

- *For a vector $\mathbf{s} \in \mathbb{Z}_q^n$, define the* LWR *distribution $L_{\mathbf{s}}$ to be the distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_p$ obtained by choosing a vector $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ uniformly at random, and outputting $(\mathbf{a}, b = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_p)$.*
- *For $s \in R_q$ (defined in Section 2), define the ring-*LWR *(*RLWR*) distribution $L_s$ to be the distribution over $R_q \times R_p$ obtained by choosing $a \leftarrow R_q$ uniformly at random and outputting $(a, b = \lfloor a \cdot s \rceil_p)$.*

*For a given distribution over $\mathbf{s} \in \mathbb{Z}_q^n$ (e.g., the uniform distribution), the decision-$\mathsf{LWR}_{n,q,p}$ problem is to distinguish (with advantage non-negligible in $n$) between any desired number of independent samples $(\mathbf{a}_i, b_i) \leftarrow L_{\mathbf{s}}$, and the same number of samples drawn uniformly and independently from $\mathbb{Z}_q^n \times \mathbb{Z}_p$. The decision-$\mathsf{RLWR}_{R,q,p}$ problem is defined analogously.*

Note that we have defined LWR exclusively as a decision problem, as this is the only form of the problem we will need. By a simple (and by now standard) hybrid argument, the (ring-)LWR problem is no easier, up to a $\mathrm{poly}(n)$ factor in advantage, if we reuse each public $\mathbf{a}_i$ across several independent secrets. That is, distinguishing samples $(\mathbf{a}_i, \lfloor \langle \mathbf{a}_i, \mathbf{s}_1 \rangle \rceil_p, \ldots, \lfloor \langle \mathbf{a}_i, \mathbf{s}_\ell \rangle \rceil_p) \in \mathbb{Z}_q^n \times \mathbb{Z}_p^\ell$ from uniform, where each $\mathbf{s}_j \in \mathbb{Z}_q^n$ is chosen independently for any $\ell = \mathrm{poly}(n)$, is at least as hard as decision-LWR for a single secret $\mathbf{s}$. An analogous statement also holds for ring-LWR.

[4] More accurately, to prove that the search problem is hard for an a priori *unbounded* number of RLWE samples, the worst-case connection from [25] requires the error distribution's parameters to themselves be chosen at random from a certain distribution. Our constructions are easily modified to account for this subtlety, but for simplicity, we ignore this issue and assume hardness for a fixed, public error distribution.

### 3.1 Reduction from LWE

We now show that for appropriate parameters, decision-LWR is at least as hard as decision-LWE. We say that a probability distribution $\chi$ over $\mathbb{R}$ (more precisely, a family of distributions $\chi_n$ indexed by the security parameter $n$) is *$B$-bounded* (where $B = B(n)$ is a function of $n$) if $\Pr_{x \leftarrow \chi}[|x| > B] \leq \mathrm{negl}(n)$. Similarly, a distribution over the ring $R$ is $B$-bounded if the marginal distribution of every coefficient (with respect to the power basis) of an $x \leftarrow \chi$ is $B$-bounded.

**Theorem 1.** *Let $\chi$ be any efficiently sampleable $B$-bounded distribution over $\mathbb{Z}$, and let $q \geq p \cdot B \cdot n^{\omega(1)}$. Then for any distribution over the secret $\mathbf{s} \in \mathbb{Z}_q^n$, solving decision-$\mathsf{LWR}_{n,q,p}$ is at least as hard as solving decision-$\mathsf{LWE}_{n,q,\chi}$ for the same distribution over $\mathbf{s}$. The same holds true for $\mathsf{RLWR}_{R,q,p}$ and $\mathsf{RLWE}_{R,q,\chi}$, for any $B$-bounded $\chi$ over $R$.*

We note that although our proof uses a super-polynomial $q = n^{\omega(1)}$, as long as $q/p \geq \sqrt{n}$ is an integer, the LWR problem appears to be exponentially hard (in $n$) for any $p = \mathrm{poly}(n)$, and super-polynomially hard for $p \leq 2^{n^\epsilon}$ for any $\epsilon < 1$, given the state of the art in noisy learning algorithms [8, 5] and lattice reduction algorithms [22, 37]. We also note that in our proof, we do not require the error terms drawn from $\chi$ in the LWE samples to be independent; we just need them all to have magnitude bounded by $B$ with overwhelming probability.

*Proof (Sketch, Theorem 1).* We give a rough proof sketch for the LWR case; the one for RLWR proceeds essentially identically. For the full and detailed proof, we refer the reader to the full version of the paper. The main idea behind the reduction is simple: given pairs $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ which are distributed either according to an LWE distribution $A_{\mathbf{s},\chi}$ or are uniformly random, we translate them into the pairs $(\mathbf{a}_i, \lfloor b_i \rceil_p) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$, which we show will be distributed according to the LWR distribution $L_{\mathbf{s}}$ (with overwhelming probability) or uniformly random, respectively.

## 4 Synthesizer-Based PRFs

We now describe the LWR-based synthesizer and our construction of a PRF from it. We first define a *pseudorandom synthesizer*, slightly modified from the definition proposed by Naor and Reingold [28].

Let $S : A \times A \to B$ be a function (where $A$ and $B$ are finite domains, which along with $S$ are implicitly indexed by the security parameter $n$) and let $X = (x_1, \ldots, x_k) \in A^k$ and $Y = (y_1, \ldots, y_\ell) \in A^\ell$ be two sequences of inputs. Then $\mathbf{C}_S(X, Y) \in B^{k \times \ell}$ is defined to be the matrix with $S(x_i, y_j)$ as its $(i, j)$th entry. (Here $\mathbf{C}$ stands for combinations.)

**Definition 2 (Pseudorandom Synthesizer).** *We say that a function $S : A \times A \to B$ is a* pseudorandom synthesizer *if it is polynomial-time computable, and if for every $\mathrm{poly}(n)$-bounded $k = k(n)$, $\ell = \ell(n)$,*

$$\mathbf{C}_S\big(U(A^k), U(A^\ell)\big) \stackrel{c}{\approx} U\big(B^{k \times \ell}\big).$$

*That is, the matrix* $\mathbf{C}_S(X, Y)$ *for uniform and independent* $X \leftarrow A^k$, $Y \leftarrow A^\ell$ *is computationally indistinguishable from a uniformly random* $k$*-by-*$\ell$ *matrix over* $B$.

### 4.1 Synthesizer Constructions

We now describe synthesizers whose security is based on the (ring-)LWR problem.

**Definition 3 ((Ring-)LWR Synthesizer).** *For moduli* $q > p \geq 2$, *the* LWR *synthesizer is the function* $S_{n,q,p} \colon \mathbb{Z}_q^n \times \mathbb{Z}_q^n \to \mathbb{Z}_p$ *defined as*

$$S_{n,q,p}(\mathbf{x}, \mathbf{y}) = \lfloor \langle \mathbf{x}, \mathbf{y} \rangle \rceil_p.$$

*The* RLWR *synthesizer is the function* $S_{R,q,p} \colon R_q \times R_q \to R_p$ *defined as*

$$S_{R,q,p}(x, y) = \lfloor x \cdot y \rceil_p.$$

**Theorem 2.** *Assuming the hardness of decision-*$\mathsf{LWR}_{n,q,p}$ *(respectively, decision-*$\mathsf{RLWR}_{R,q,p}$*) for a uniformly random secret, the function* $S_{n,q,p}$ *(respectively,* $S_{R,q,p}$*) given in Definition 3 above is a pseudorandom synthesizer.*

It follows generically from this theorem that the function $T_{n,q,p} \colon \mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^{n \times n} \to \mathbb{Z}_p^{n \times n}$, defined as $T_{n,q,p}(\mathbf{X}, \mathbf{Y}) = \lfloor \mathbf{X} \cdot \mathbf{Y} \rceil_p$, is also a pseudorandom synthesizer, since by the definition of matrix multiplication, we only incur a factor of $n$ increase in the length of the input sequences. This is the synthesizer that we use below in the construction of a PRF.

### 4.2 The PRF Construction

**Definition 4 ((Ring-)LWR PRF).** *For parameters* $n \in \mathbb{N}$, *input length* $k = 2^d \geq 1$, *and moduli* $q_d \geq q_{d-1} \geq \ldots \geq q_0 \geq 2$, *the* LWR *family* $\mathcal{F}^{(j)}$ *for* $0 \leq j \leq d$ *is defined inductively to consist of functions from* $\{0,1\}^{2^j}$ *to* $\mathbb{Z}_{q_{d-j}}^{n \times n}$. *We define* $\mathcal{F} = \mathcal{F}^{(d)}$.

- *For* $j = 0$, *a function* $F \in \mathcal{F}^{(0)}$ *is indexed by* $\mathbf{S}_b \in \mathbb{Z}_{q_d}^{n \times n}$ *for* $b \in \{0,1\}$, *and is defined simply as* $F_{\{\mathbf{S}_b\}}(x) = \mathbf{S}_x$. *We endow* $\mathcal{F}^{(0)}$ *with the distribution where the* $\mathbf{S}_b$ *are uniform and independent.*
- *For* $j \geq 1$, *a function* $F \in \mathcal{F}^{(j)}$ *is indexed by some* $F_0, F_1 \in \mathcal{F}^{(j-1)}$, *and is defined as*

  $$F_{F_0,F_1}(x_0, x_1) = T^{(j)}\big(\, F_0(x_0)\,,\, F_1(x_1)\,\big)$$

  *where* $|x_0| = |x_1| = 2^{j-1}$ *and* $T^{(j)} = T_{n,q_{d-j+1},q_{d-j}}$ *is the appropriate synthesizer. We endow* $\mathcal{F}^{(j)}$ *with the distribution where* $F_0$ *and* $F_1$ *are chosen independently from* $\mathcal{F}^{(j-1)}$.

*The ring-*LWR *family* $\mathcal{RF}^{(j)}$ *is defined similarly to consist of functions from* $\{0,1\}^{2^j}$ *to* $R_{q_{d-j}}$, *where in the base case* ($j = 0$) *we replace each* $\mathbf{S}_b$ *with a uniformly random* $s_b \in R_{q_d}$, *and in the inductive case* ($j \geq 1$) *we use the ring-*LWR *synthesizer* $S^{(j)} = S_{R,q_{d-j+1},q_{d-j}}$.

We remark that the recursive LWR-based construction above does not have to use *square* matrices; any legal dimensions would be acceptable with no essential change to the security proof. Square matrices appear to give the best combination of seed size, computational efficiency, and input/output lengths.

### 4.3 Security

The security proof for our PRF hinges on the fact that the functions $T^{(j)} = T_{n,q_{d-j+1},q_{d-j}}$ are synthesizers for appropriate choices of the moduli. In fact, the proof is essentially identical to Naor and Reingold's [28] for their PRF construction from pseudorandom synthesizers; the only reason we cannot use their theorem exactly as stated is because they assume that the synthesizer output is exactly the same size as its two inputs, which is not quite the case with our synthesizer due to the modulus reduction. This is a minor detail that does not change the proof in any material way; it only limits the number of times we may compose the synthesizer, and hence the input length of the PRF. We thus refer the reader to the full version for the proof.

**Theorem 3.** *Assuming that $T^{(j)} = T_{n,q_{d-j+1},q_{d-j}}$ is a pseudorandom synthesizer for every $j \in [d]$, the* LWR *family $\mathcal{F}$ from Definition 4 is a pseudorandom function family.*

*The same is also true for the ring-*LWR *family $\mathcal{RF}$, assuming that $S^{(j)} = S_{R,q_{d-j+1},q_{d-j}}$ is a pseudorandom synthesizer for every $j \in [d]$.*

## 5 Direct PRF Constructions

Here we present another, potentially more efficient construction of a pseudorandom function family whose security is based on the intractibility of the LWE problem.

**Definition 5 ((Ring-)LWE degree-$k$ PRF).** *For parameters $n \in \mathbb{N}$, moduli $q \geq p \geq 2$, positive integer $m = \mathrm{poly}(n)$, and input length $k \geq 1$, the family $\mathcal{F}$ consists of functions from $\{0,1\}^k$ to $\mathbb{Z}_p^{m \times n}$. A function $F \in \mathcal{F}$ is indexed by some $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{S}_i \in \mathbb{Z}^{n \times n}$ for each $i \in [k]$, and is defined as*

$$F(x) = F_{\mathbf{A},\{\mathbf{S}_i\}}(x_1 \cdots x_k) := \left\lfloor \mathbf{A}^t \cdot \prod_{i=1}^{k} \mathbf{S}_i^{x_i} \right\rceil_p . \tag{6}$$

*We endow $\mathcal{F}$ with the distribution where $\mathbf{A}$ is chosen uniformly at random, and below we consider a number of natural distributions for the $\mathbf{S}_i$.*

*The ring-based family $\mathcal{RF}$ is defined similarly to consist of functions from $\{0,1\}^k$ to $R_p$, where we replace $\mathbf{A}$ with uniformly random $a \in R_q$ and each $\mathbf{S}_i$ with some $s_i \in R$.*

### 5.1 Efficiency

Consider a function $F \in \mathcal{F}$ as in Definition 5. Using ideas from [36], we see that both binary matrix product and rounding can be implemented with simple depth-2 arithmetic circuits, and hence in $\mathrm{TC}^0$, so at worst $F$ can be computed in $\mathrm{TC}^1$ by computing the subset product in a tree-like fashion, followed by a final rounding step.

The ring variant of Construction 6 appears to be more efficient to evaluate, by storing the ring elements in the discrete Fourier transform or "Chinese remainder" representation modulo $q$ (see, e.g., [24, 25]), so that multiplication of two ring elements just corresponds to a coordinate-wise product of their vectors. Then to evaluate the function, one would just compute a subset-product of the appropriate vectors, then interpolate the result to the power-basis representation, using essentially an $n$-dimensional Fast Fourier Transform over $\mathbb{Z}_q$, in order to perform the rounding operation. In terms of theoretical depth, the multi-product of vectors can be performed in $\mathrm{TC}^0$, as can the Fast Fourier Transform and rounding steps [36]. This implies that the entire function can be computed in $\mathrm{TC}^0$, matching (asymptotically) the shallowest known PRFs based on the DDH and factoring problems [29, 30].

### 5.2 Security under LWE

**Theorem 4.** *Let $\chi = D_{\mathbb{Z},r}$ for some $r > 0$, and let $q \geq p \cdot k(Cr\sqrt{n})^k \cdot n^{\omega(1)}$ for a suitable universal constant $C$. Endow the family $\mathcal{F}$ from Definition 4 with the distribution where each $\mathbf{S}_i$ is drawn independently from $\chi^{n \times n}$. Then assuming the hardness of decision-$\mathsf{LWE}_{n,q,\chi}$, the family $\mathcal{F}$ is pseudorandom.*

An analogous theorem holds for the ring-based family $\mathcal{RF}$, under decision-RLWE.

**Theorem 5.** *Let $\chi$ be the distribution over the ring $R$ where each coefficient (with respect to the power basis) is chosen independently from $D_{\mathbb{Z},r}$ for some $r > 0$, and let $q \geq p \cdot k(r\sqrt{n} \cdot \omega(\sqrt{\log n}))^k \cdot n^{\omega(1)}$. Endow the family $\mathcal{RF}$ from Definition 4 with the distribution where each $s_i$ is drawn independently from $\chi$. Then assuming the hardness of decision-$\mathsf{RLWE}_{n,q,\chi}$, the family $\mathcal{RF}$ is pseudorandom.*

*Proof (Sketch, Theorem 4).* To aid the proof, it helps to define a family $\mathcal{G}$ of functions $G\colon \{0,1\}^k \to \mathbb{Z}_q^{n \times n}$, which are simply the unrounded counterparts of the functions in $\mathcal{F}$. That is, for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{S}_i \in \mathbb{Z}^{n \times n}$ for $i \in [k]$, we define $G_{\mathbf{A},\{\mathbf{S}_i\}}(x_1 \cdots x_k) := \mathbf{A}^t \cdot \prod_{i=1}^k \mathbf{S}_i^{x_i}$. We endow $\mathcal{G}$ with the same distribution over $\mathbf{A}$ and the $\mathbf{S}_i$ as $\mathcal{F}$ has.

We proceed via a sequence of games, much like in the proof of Theorem 1. First as a "thought experiment" we define a new family $\tilde{\mathcal{G}}$ of functions from $\{0,1\}^k$ to $\mathbb{Z}_q^{m \times n}$. This family is a counterpart to $\mathcal{G}$, but with two important differences: it is a PRF family *without* any rounding (and hence, with rounding as well), but each function in the family has an exponentially large key. Alternatively, one

may think of the functions in $\tilde{\mathcal{G}}$ as *randomized* functions with small keys. Then we show that with overwhelming probability, the rounding of $\tilde{G} \leftarrow \tilde{\mathcal{G}}$ agrees with the rounding of the corresponding $G \in \mathcal{G}$ on all the attacker's queries, because the outputs of the two functions are relatively close. It follows that the rounding of $G \leftarrow \mathcal{G}$ (i.e., $F \leftarrow \mathcal{F}$) cannot be distinguished from a uniformly random function, as desired. We again refer the reader to the full version of the paper for the formal proof.

## References


[1] Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)

[2] Agrawal, S., Boneh, D., Boyen, X.: Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)

[3] Ajtai, M.: Generating hard instances of lattice problems. Quaderni di Matematica 13, 1–32 (2004); Preliminary version in STOC 1996

[4] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)

[5] Arora, S., Ge, R.: New Algorithms for Learning in Presence of Errors. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 403–415. Springer, Heidelberg (2011)

[6] Banerjee, A., Ben-Zvi, N., Peikert, C., Rosen, A.: SPRINT: Efficient pseudorandomness via rounded integer products (2011) (manuscript)

[7] Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. Cryptology ePrint Archive, Report 2011/401 (2011), `http://eprint.iacr.org/`

[8] Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM 50(4), 506–519 (2003)

[9] Boneh, D., Montgomery, H.W., Raghunathan, A.: Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In: ACM Conference on Computer and Communications Security, pp. 131–140 (2010)

[10] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Report 2011/277 (2011), `http://eprint.iacr.org/`

[11] Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: FOCS, pp. 97–106 (2011)

[12] Brakerski, Z., Vaikuntanathan, V.: Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011)

[13] Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)

[14] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC, pp. 169–178 (2009)

[15] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)

[16] Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM 33(4), 792–807 (1986); Preliminary version in FOCS 1984
[17] Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: ICS, pp. 230–240 (2010)
[18] Hopper, N.J., Blum, M.: Secure Human Identification Protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001)
[19] Juels, A., Weis, S.A.: Authenticating Pervasive Devices with Human Protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
[20] Katz, J., Shin, J.S., Smith, A.: Parallel and concurrent security of the HB and $HB^+$ protocols. J. Cryptology 23(3), 402–421 (2010); Preliminary version in Eurocrypt 2006
[21] Kiltz, E., Pietrzak, K., Cash, D., Jain, A., Venturi, D.: Efficient Authentication from Hard Learning Problems. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 7–26. Springer, Heidelberg (2011)
[22] Lenstra, A.K., Lenstra Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen 261(4), 515–534 (1982)
[23] Lewko, A.B., Waters, B.: Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In: ACM Conference on Computer and Communications Security, pp. 112–120 (2009)
[24] Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFT: A Modest Proposal for FFT Hashing. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 54–72. Springer, Heidelberg (2008)
[25] Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)
[26] Micciancio, D., Mol, P.: Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011)
[27] Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
[28] Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of pseudo-random functions. J. Comput. Syst. Sci. 58(2), 336–375 (1999); Preliminary version in FOCS 1995
[29] Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. J. ACM 51(2), 231–262 (2004); Preliminary version in FOCS 1997
[30] Naor, M., Reingold, O., Rosen, A.: Pseudorandom functions and factoring. SIAM J. Comput. 31(5), 1383–1404 (2002); Preliminary version in STOC 2000
[31] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC, pp. 333–342 (2009)
[32] Peikert, C.: An Efficient and Parallel Gaussian Sampler for Lattices. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 80–97. Springer, Heidelberg (2010)
[33] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC, pp. 187–196 (2008)
[34] Pietrzak, K.: Subspace LWE (2010) (manuscript), `http://homepages.cwi.nl/~pietrzak/publications/SLWE.pdf` (Last retrieved from June 28, 2011)

[35] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6), 1–40 (2009); Preliminary version in STOC 2005
[36] Reif, J.H., Tate, S.R.: On threshold circuits and polynomial computation. SIAM J. Comput. 21(5), 896–908 (1992)
[37] Schnorr, C.-P.: A hierarchy of polynomial time lattice basis reduction algorithms. Theor. Comput. Sci. 53, 201–224 (1987)
[38] van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)