Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems

Benny Applebaum*, David Cash, Chris Peikert**, and Amit Sahai***

[1] Princeton University
[2] Georgia Institute of Technology
[3] SRI International
[4] UCLA

Abstract. The well-studied task of *learning a linear function with errors* is a seemingly hard problem and the basis for several cryptographic schemes. Here we demonstrate additional applications that enjoy strong security properties and a high level of efficiency. Namely, we construct:

1. Public-key and symmetric-key cryptosystems that provide security for *key-dependent messages* and enjoy *circular security.* Our schemes are highly efficient: in both cases the ciphertext is only a constant factor larger than the plaintext, and the cost of encryption and decryption is only $n \cdot \mathrm{polylog}(n)$ bit operations per message symbol in the public-key case, and $\mathrm{polylog}(n)$ bit operations in the symmetric-case.
2. Two efficient pseudorandom objects: a "weak randomized pseudorandom function" — a relaxation of standard PRF — that can be computed obliviously via a simple protocol, and a length-doubling pseudorandom generator that can be computed by a circuit of $n \cdot \mathrm{polylog}(n)$ size. The complexity of our pseudorandom generator almost matches the complexity of the fastest known construction (Applebaum *et al.*, RANDOM 2006), which runs in linear time at the expense of relying on a nonstandard intractability assumption.

Our constructions and security proofs are simple and natural, and involve new techniques that may be of independent interest. In addition, by combining our constructions with prior ones, we get fast implementations of several other primitives and protocols.

Keywords: Encryption, Key-dependent message security, Learning problems, Lattice-based cryptography

* This material is based upon work supported by the National Science Foundation under Grants CNS-0627526, CCF-0426582 and CCF-0832797. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

** This material is based upon work supported by the National Science Foundation under Grants CNS-0716786 and CNS-0749931.

*** Research supported in part from NSF grants 0716389, 0627781, 0830803, a subgrant from SRI as part of the Army Cyber-TA program, an equipment grant from Intel, an Alfred P. Sloan Foundation Fellowship, and an Okawa Foundation Research Grant.

1 Introduction

The problem of "learning a linear function with errors" (LWE) has found many interesting cryptographic and complexity-theoretic applications in the last few years (see [32, 36, 46, 37, 45, 25], to name a few). Informally, the LWE problem, for a dimension n and modulus q, is to recover a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ given arbitrarily many "noisy random inner products" $(\mathbf{a}_i, b_i \approx \langle \mathbf{a}_i, \mathbf{s} \rangle) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where the $\mathbf{a}_i \in \mathbb{Z}_q^n$ are uniform and independent. The "learning parity with noise" problem (LPN) is the special case where $q = 2$. These problems have been studied extensively in several works, and their known best algorithms require $2^{O(n \log q / \log n)}$ time and space [11].

Much evidence suggests that no efficient algorithm can solve LWE/LPN with better than negligible probability, even using quantum computation. In particular, the LPN problem can be formulated as the famous problem of decoding a random binary linear code, and therefore a successful attack would imply a major breakthrough in coding theory. The LPN problem also occupies a central position in learning theory: an efficient algorithm for it could be used to learn several important concept classes, including 2-DNF formulas, juntas, and *any* function with a sparse Fourier spectrum [21].

For the case of the LWE, hardness is supported by a remarkable connection to *worst-case* lattice problems. Regev [46] showed that solving LWE (for certain Gaussian-like error distributions) is as hard as *quantumly* solving some apparently intractable lattice problems, such as the approximate shortest vector problem GapSVP. Recently, Peikert [43] also gave a *classical* reduction from GapSVP (and variants) to LWE.

The LWE/LPN problems provide an interesting combination of two useful properties: *efficiency*, i.e., instances of the problem can be generated by "cheap" operations such as (modular) addition and multiplication (or even simple *bit operations* in the case of LPN), and *simple algebraic structure*, i.e., the noisy inner products are computed by an "almost linear" function. Indeed, previous works relied on these properties to obtain cryptography with low complexity [32, 4, 35, 6], and to derive desirable cryptographic features such as random self-reducibility (with respect to the choice of $\mathbf{a}$) and pseudorandomness [10]. Interestingly, other problems that provide the latter feature, such as those from number theory, typically require relatively expensive computational operations such as exponentiation over large groups.

1.1 Our Results

In this paper, we further exploit the properties of LWE/LPN to obtain new cryptographic constructions that are both efficient and enjoy desirable security properties.

Circular-secure encryption schemes. One of our main applications is the construction of efficient encryption schemes (in both the symmetric- and public-key settings) that achieve security against certain *key-dependent message* (KDM) attacks [9]; that is, they remain secure even when the adversary is allowed to obtain encryptions of messages that depend on the secret keys themselves, via any affine function of the adversary's choice. Moreover, our schemes are "circular secure" [15], that is, they remain secure even in the presence of key "cycles" or even "cliques," where any user's secret key may be encrypted under any user's public key. Such usage arises in key-management systems, in anonymous credential systems [15], and in the context of "axiomatic security" [1] (See [13] for a detailed discussion).

In the last few years, the notions of KDM and circular security have been studied extensively [29, 8, 14, 7, 31, 28]. Without resorting to the use of random oracles, constructing a circular-secure encryption scheme (either in the private-key or public-key setting) was a long-standing open problem. This question was recently resolved by Boneh *et al.* [13], who constructed such a *public-key* encryption scheme based on the DDH assumption. Their construction relies on a clever use of the homomorphic properties of the DDH problem. However, exploiting these properties incurs a large overhead in both computation and communication. In contrast, our approach yields very *natural* encryption schemes that have significant efficiency advantages over the prior scheme of [13].

The contrast is clearest when we compare the current "cost" of achieving security for key-dependent messages against the cost of achieving ordinary semantic security, for a given computational intractability assumption. Comparing the scheme of [13] to other semantically secure encryption schemes based on the DDH problem, the cost is dramatic: While standard encryption schemes like ElGamal can encrypt their key, which is about $k = \log |\mathbb{G}|$ bits (where $\mathbb{G}$ is the underlying group of the DDH problem), using a single exponentiation and one group element of overhead in the ciphertext, the scheme given in [13] requires about k exponentiations and group elements of ciphertext overheard *per bit of the secret key.* Encrypting key-independent messages is about k times more efficient, but it still incurs a factor k loss over standard ElGamal. In contrast, our constructions are essentially *as efficient* as prior semantically secure schemes based on essentially the same hardness assumptions.

Specifically, our *public-key* schemes are variants of Regev's LWE-based scheme and the more-efficient amortized version of Peikert, Vaikuntanathan, and Waters [44], with several non-trivial modifications to facilitate the proof of security for key-dependent messages. The most efficient version takes only $\tilde{O}(n)$ amortized time per message symbol for both encryption and decryption, and the ciphertext is only a constant factor larger than the plaintext.

Our *symmetric-key* cryptosystem is based on the LPN problem. Its ciphertexts are only a constant factor larger than the plaintexts, and both encryption and decryption can be performed by Boolean circuits of quasi-linear size (in the message length), which is almost optimal even for standard CPA-security. The scheme is a close variant of the LPN-based encryption scheme of Gilbert *et al.* [26], which was proved secure only in the standard sense (i.e., without key-dependent messages), and did not achieve quasi-linear time efficiency. The scheme was discovered independently by the first author and by Dodis *et al.* [19], who proved security in the presence of key-leakage under a stronger version of the LPN assumption. We stress that key-leakage security is incomparable to the notions studied here.

Fast pseudorandom objects.

Pseudorandom generator. Based on the hardness of LPN, we construct a pseudorandom generator (PRG) that doubles its input length and can be computed by a Boolean circuit of size $\widetilde{O}(n)$ (i.e., quasilinear size). This is considerably faster than previous constructions of linear-stretch PRGs (e.g., [18, 17, 24]), which suffer from *polynomial* overhead. (Similar limitations also hold for previous coding-based constructions [10, 22].) To the best of our knowledge, the only exception is the construction of [5] which is computable by linear-size ($\mathbf{NC^0}$) circuits. This construction is based on a plausible, yet non-standard, assumption of Alekhnovich [3]. Roughly speaking, that assumption says that a noisy random codeword of a code with *sparse* generating matrix is pseudorandom. This assumption is relatively new and, while seemingly reasonable, it has not been widely studied yet. Moreover, unlike our LPN-based assumption, Alekhnovich's assumption posits *pseudorandomness* rather than just *one-wayness*, which is in general a stronger notion.

Application. Typical cryptographic functions introduce a multiplicative computational overhead that grows with the desired level of security. Recently, Ishai *et al.* [34] showed that many cryptographic tasks can be implemented while incurring only a constant computational overhead compared to insecure implementations of the same tasks.[5] These results were based on the PRG construction of [5], and hence on non-standard intractability assumptions.

By plugging our PRG into the reductions of [34], we get implementations with polylogarithmic overhead for several primitives such as commitment schemes, symmetric encryption schemes, and public-key encryption schemes (under the assumption that the latter exist). This provides an interesting alternative to the original suggestion of [34], as it relies on a standard assumption and still gives a considerable improvement over typical (non-IKOS) schemes.[6] We view this result as an important support for the possibility of cryptography with low overhead.

Randomized weak pseudorandom function. We also obtain a simple construction of randomized weak pseudorandom function family (RWPRFs). This primitive

[5] We make the usual security requirement that the advantage of any polynomial-time attacker must be negligible in the input length.

[6] A trivial construction of primitives with polylogarithmic security can be achieved by starting from an exponentially strong primitive (e.g., PRG) and applying it separately on input blocks of polylogarithmic size. This construction results in primitives with weak (quasi-polynomial) security. In contrast, our construction starts from a weaker assumption (in particular, it does not require exponential hardness) and it results in a primitive whose security is essentially the same as the security of the assumption (up to standard polynomial loss).

relaxes the standard notion of pseudorandom function family [27] in two ways: it provides security only when the function is evaluated on randomly chosen points, and it uses secret internal randomness. To make this notion nontrivial we require an efficient "equality-tester" that verifies whether different invocations of the PRF (with independent internal randomness) correspond to the same preimage. While this primitive is considerably weaker than PRFs, we argue that in some scenarios RWPRFs can be used instead of standard PRFs. Moreover, the use of internal randomness provably admits more efficient constructions.[7]

Our construction has several interesting syntactic properties: it is injective and symmetric (one can replace the roles of the argument and the key without violating security). Moreover, we describe a simple constant-round protocol for obliviously evaluating the function. Such a protocol allows two parties, one holding a point x and another holding a key k, to evaluate the function $f_k(x)$ without learning each other's inputs. Pseudorandom functions that allow oblivious evaluation (OPRFs) were recently shown to be a useful cryptographic tool [23, 30]. An oblivious RWPRF can replace an OPRF in some settings (despite its weaker cryptographic properties), and hence our construction provides an alternative to the relatively small number of existing schemes (see [23] and references within).

Practical efficiency vs. asymptotic efficiency. In this paper, we treat efficiency in *asymptotic* terms. Still, we believe that some of our results may turn to be useful in practice as well. Indeed, our LPN-based constructions mainly rely on addition and multiplication of large binary matrices. These operations can be performed very fast in practice [12, 2] even if one does not employ the asymptotically-fast algorithms used in our analysis (e.g., for matrix multiplication), which might not be applicable in practice. In particular, as in the case of the HB protocol [32, 35], our schemes (or variants of them) might turn to be useful for hardware implementation by computationally-weak devices. We leave this direction for future study.

1.2 Techniques

Our LWE-based public key construction involves a few techniques that may be of independent interest and application.

In the LWE-based cryptosystems of [46, 44], the secret key is a vector $\mathbf{s} \in \mathbb{Z}_q^n$ chosen uniformly at random, while the message space is $\mathbb{Z}_p$ for some $p \ll q$. An important idea in the work of Boneh *et al.* [13] is the ability to generate, given only a public key, a ciphertext that decrypts to a message related to $\mathbf{s}$. Because decryption in the LWE-based schemes of [46, 44] is essentially a linear operation, it is easy to generate ciphertexts that are somehow related to $\mathbf{s}$. However, because

[7] For example, it can be shown that PRFs cannot be computed by constant-depth circuits with unbounded fan-in AND and XOR gates [38]. In contrast, our construction can be computed by such circuits of depth 2. Moreover, one can show that, under plausible assumptions, RWPRFs can be constructed even in weaker classes such as $\mathbf{NC^0}$.

the entries of $\mathbf{s}$ are taken modulo $q \gg p$, it is unclear how to "fit" the entries into the message space.

We address this issue by instead drawing the entries of the secret key $\mathbf{s}$ from the very same (Gaussian) *error distribution* as in the underlying LWE problem. For a sufficiently "narrow" error distribution, each entry of $\mathbf{s}$ can take on at most p different values (with overwhelming probability), allowing the entire entry to fit unambiguously into the message space. Moreover, this change *does not affect the hardness of the* LWE *problem*: we show a simple, tight reduction from the standard LWE problem to the variant just described. Abstractly, the reduction may be viewed as putting the LWE distribution into *Hermite normal form* (HNF); interestingly, the HNF was also used by Micciancio [39] and Micciancio and Regev [41] as a way to improve the *efficiency* of lattice-based cryptosystems.

The second important technique relates to the faithful simulation of key-dependent messages. We modify the encryption algorithms of [46, 44] to ensure that ciphertexts *themselves* have a "nice" distribution that supports the desired homomorphisms. Essentially, our encryption algorithms apply Regev's worst-case to average-case reduction (from lattices to LWE) to the (already random) public key itself; we also generalize Regev's analysis to deal with the amortized system of [44]. In addition, to support the homomorphisms we need to rely on LWE with a *prime power* modulus $q = p^e$, where p is the size of the message space. Fortunately, a hybrid-argument extension of the usual pseudorandomness proof [10, 46] for LWE also works for prime power moduli, as long as the error distribution is sufficiently "narrow."

A final interesting technique concerns a more general attack involving key cycles/cliques, where every user's secret key may be encrypted under every user's public key. Simulating such a scenario seems to require knowing a relation between every pair of (unknown and independent) secret keys. Conveniently, the above-described transformation for LWE (allowing the secret $\mathbf{s}$ to be drawn from the error distribution) can also be used to produce many independent keys, and happens to produce the desired linear relations among them as a side effect!

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 $U(D)$. We write U_n to denote the special case of the uniform distribution over $\{0,1\}^n$ and (by abuse of notation) the uniform distribution over $\mathbb{Z}_2^n$. Let Ber_ε denote the Bernoulli distribution over $\{0,1\}$ that is 1 with probability ε and 0 with probability $1-\varepsilon$.

We write $\mathrm{negl}(n)$ to denote an arbitrary *negligible* function, i.e., one that vanishes faster than the inverse of any polynomial. We say that that a probability is *overwhelming* if it $1 - \mathrm{negl}(n)$.

The *statistical distance* between two distributions X and Y over a countable domain D (or two random variables having those distributions) is defined as $\Delta(A, B) = \max_{A \subseteq D} |f_X(A) - f_Y(A)|$. We write $X \equiv Y$ if the two random variable are identically distributed. We say that two ensembles $\{X_n\}$ and $\{Y_n\}$ of distributions indexed by n are *statistically indistinguishable* if $\Delta(X_n, Y_n) = \mathrm{negl}(n)$. The ensembles are *computationally indistinguishable* if for every probabilistic polynomial-time adversary $\mathcal{A}$, the distinguishing advantage $|\Pr[\mathcal{A}(X_n) = 1] - \Pr[\mathcal{A}(Y_n) = 1]| = \mathrm{negl}(n)$. A distribution ensemble $\{X_n\}_{n\in\mathbb{N}}$ is *pseudorandom* if X_n is computationally indistinguishable from $U(D_n)$ where D_n is the domain of X_n (which is usually clear from the context).

2.1 Noisy Learning Problems

We recall the learning with error (LWE), due to Regev [46], for which learning parity with noise (LPN) is a special case.

For positive integers n and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution χ on $\mathbb{Z}_q$, define $A_{\mathbf{s},\chi}$ to be the distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, an error term $x \leftarrow \chi$, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + x)$.

Definition 1. *For an integer function $q = q(n)$ and an error distribution χ over $\mathbb{Z}_q$, the* learning with errors *problem* $\mathsf{LWE}_{q,\chi}$ *in n dimensions is defined as follows: given access to an oracle that produces independent samples from $A_{\mathbf{s},\chi}$ for some arbitrary $\mathbf{s} \in \mathbb{Z}_q^n$, output $\mathbf{s}$ with noticeable probability, e.g., $1/2$, over all the randomness of the oracle and the algorithm.*

The learning parity with noise *problem* LPN_ε *is the special case of* $\mathsf{LWE}_{q,\chi}$ *for $q = 2$ and $\chi = \mathsf{Ber}_\varepsilon$.*

We say that $\mathsf{LWE}_{q,\chi}$ is *hard* (or intractable) for a class of adversaries (by default, probabilistic poly(n)-time algorithms) if there does not exist an algorithm in the class that can solve it for infinitely many n.

Note that LWE as defined above is a "worst-case" style of problem in that the value of $\mathbf{s} \in \mathbb{Z}_q^n$ is arbitrary, not random as is typical in cryptography. This is not too important of a distinction, because LWE is amenable to randomized self-reduction and amplification techniques [10, 46]. In particular, here we give a reduction from the form of the problem in Definition 1 to an *average-case decision problem*, for *prime power moduli* and "narrow enough" error distributions. In other words, under the hypotheses of the lemma, the LWE distribution is pseudorandom if the search problem is hard.

Lemma 1. *Let $q = p^e$ be a prime power with $p = \mathrm{poly}(n)$, and let χ be a distribution over $\mathbb{Z}_q$ that produces an element in $\{-\frac{p-1}{2}, \ldots, \frac{p-1}{2}\} \subset \mathbb{Z}_q$ with overwhelming probability. There is a probabilistic polynomial-time reduction from solving* $\mathsf{LWE}_{q,\chi}$ *to distinguishing (with non-negligible advantage) between $A_{\mathbf{s},\chi}$ for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ and the uniform distribution $U = U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$.*

Proof (sketch). The proof is a simple extension of prior ones for prime moduli (see, e.g., [46, Lemma 4.2]), therefore we sketch only the new elements. The idea is to use a distinguisher to recover the *least significant* digit (in base p) of each entry of $\mathbf{s}$, after which the error distribution can be made narrow enough to solve for all the remaining digits of $\mathbf{s}$ via rounding and linear algebra. Due to space limitations, the entire proof is deferred to the full version.

For $i = 0, \dots, e$, define the hybrid distribution $A^i_{\mathbf{s},\chi}$ that is obtained by drawing a sample $(\mathbf{a}, b)$ from $A_{\mathbf{s},\chi}$ and outputting $(\mathbf{a}, b + p^i \cdot r) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ for a uniformly random $r \in \mathbb{Z}_q$ (freshly chosen for each sample). By a hybrid argument and standard amplification techniques, we can use an algorithm D that distinguishes between $A_{\mathbf{s},\chi}$ and U to solve for $\mathbf{s}' = \mathbf{s} \bmod p$. Having done so, we can then transform $A_{\mathbf{s},\chi}$ into $A_{p\cdot\mathbf{t},\chi}$, where $p \cdot \mathbf{t} = \mathbf{s} - \mathbf{s}' \in \mathbb{Z}_q^n$. A sample from the latter distribution is of the form $(\mathbf{a}, b = p \cdot \langle \mathbf{a}, \mathbf{t} \rangle + x)$ for $x \leftarrow \chi$; because $x \in \{-\frac{p-1}{2}, \dots, \frac{p-1}{2}\}$ with overwhelming probability, we may round b to the nearest multiple of p and learn the value of $\langle \mathbf{a}, \mathbf{t} \rangle \bmod p$ exactly. With enough samples of this form, we may then solve for $\mathbf{t}$ by linear algebra.

We are interested in error distributions χ over $\mathbb{Z}_q$ that are derived from Gaussians. For any $r > 0$, define the one-dimensional Gaussian probability distribution by its density function $D_r(x) = \exp(-\pi(x/r)^2)/r$. For $\alpha > 0$, define $\bar{\Psi}_\alpha$ to be the distribution on $\mathbb{Z}_q$ obtained by drawing $y \leftarrow D_\alpha$ and outputting $\lfloor q \cdot y \rceil \bmod q$. Regev [46] demonstrated strong evidence for the hardness of the LWE problem with such a Gaussian error distribution, by giving a *quantum* reduction from approximating well-studied lattice problems to within $\tilde{O}(n/\alpha)$ factors in the *worst case* to solving $\mathsf{LWE}_{q,\bar{\Psi}_\alpha}$, when (say) $\alpha \cdot q \geq n$. Recently, Peikert [43] also gave a related *classical* reduction for similar parameters.

For our public-key encryption algorithms, we also need the *discrete Gaussian* distribution $D_{\mathbb{Z}^m,r}$ over the integer lattice $\mathbb{Z}^m$, which assigns probability proportional to $\prod_{i\in[m]} D_r(x_i)$ to each $\mathbf{x} \in \mathbb{Z}^m$. It is possible to sample efficiently from $D_{\mathbb{Z}^m,r}$ for any $r > 0$ [25].

2.2 Key-Dependent Message Security

We now define key-dependent message security for encryption, following the presentation of Boneh *et al.* [13], which generalizes the definition of Black *et al.* [9]. In this definition, an adversary plays a game with a challenger that answers encryption queries for functions of the users' secret keys. The adversary is restricted to queries for functions from a certain family, which we will denote $\mathcal{F} \subset \{f \mid f : \mathcal{K}^\ell \to \mathcal{M}\}$, where $\mathcal{K}$ and $\mathcal{M}$ are the keyspace and message space of the encryption scheme. Strictly speaking, $\mathcal{F}$ is a family of sets of functions parameterized by the security parameter n and the number of users ℓ.

Let us fix a public-key encryption scheme, and let $\mathcal{A}$ be an adversary. We will write $\mathsf{Enc}(pk, m)$ to denote encrypting message m under public key pk. The game proceeds as follows:

1. The challenger chooses a bit $b \leftarrow \{0, 1\}$. It also chooses $(pk_1, sk_1), \dots, (pk_\ell, sk_\ell)$ by running the scheme's key generation algorithm ℓ times. It gives $pk_1, \dots, pk_\ell$ to the adversary.
2. $\mathcal{A}$ makes encryption queries of the form (i, f), where $1 \leq i \leq \ell$ and $f \in \mathcal{F}$. To process a query, if $b = 0$, the challenger computes $m \leftarrow f(sk_1, \dots, sk_\ell)$

and $c \leftarrow \mathsf{Enc}(pk_i, m)$. If $b = 1$ it instead sets $c \leftarrow \mathsf{Enc}(pk_i, 0^{|m|})$. It returns c to $\mathcal{A}$.

3. $\mathcal{A}$ attempts to guess b and outputs $\hat{b} \in \{0, 1\}$.

The scheme is *KDM-CPA secure with respect to* $\mathcal{F}$ if for every efficient adversary $\mathcal{A}$, the probability of guessing b is at most $\frac{1}{2} + \text{negl}(n)$ for some negligible function $\text{negl}(\cdot)$.

We can define KDM-CPA security for symmetric key encryption similarly: in phase one, the challenger generates secret keys and gives the adversary nothing, and in phase two it uses the secret keys to encrypt (and as input to f). Everything else is exactly the same. Finally, the definition of *CCA-KDM security* is similar except that the adversary has also an oracle access to the decryption function $\mathsf{Dec}(k, \cdot)$ (but cannot query this oracle on any output given to him by the encryption oracle).

If all constant functions (that is, functions f_m such that $f_m(k_1, \ldots, k_\ell) = m$ for some $m \in \mathcal{M}$) are contained in $\mathcal{F}$, then security with respect to $\mathcal{F}$ implies standard CPA security. If the projection functions (f_j such that $f_j(k_1, \ldots, k_\ell) = k_j$ for some j) are contained in $\mathcal{F}$, then security with respect to $\mathcal{F}$ implies (and is actually stronger than) circular security.

3 Public-Key Encryption

In this section we design a public-key cryptosystem based on the $\mathsf{LWE}_{q,\chi}$ problem, where as usual, the error distribution χ is the discretized Gaussian $\bar{\Psi}_\alpha$ for parameter $\alpha = \alpha(n) \in (0, 1)$, and the modulus q is chosen to satisfy various constraints.

3.1 A Generic Transformation

We start with a useful transformation that reduces the LWE problem to one in which the *secret itself* is chosen from the error distribution χ, essentially putting the LWE distribution into "Hermite normal form."

Lemma 2. *Let $q = p^e$ be a prime power. There is a deterministic polynomial-time transformation T that, for arbitrary $\mathbf{s} \in \mathbb{Z}_q^n$ and error distribution χ, maps $A_{\mathbf{s},\chi}$ to $A_{\bar{\mathbf{x}},\chi}$ where $\bar{\mathbf{x}} \leftarrow \chi^n$, and maps $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ to itself. The transformation also produces an invertible square matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times n}$ and $\bar{\mathbf{b}} \in \mathbb{Z}_q^n$ that, when mapping $A_{\mathbf{s},\chi}$ to $A_{\bar{\mathbf{x}},\chi}$, satisfy $\bar{\mathbf{x}} = -\bar{\mathbf{A}}^T\mathbf{s} + \bar{\mathbf{b}}$.*

Proof. The transformation T is given access to some distribution D over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ (where D may be either $A_{\mathbf{s},\chi}$ or $U = U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$), and proceeds in two stages.

In the first stage, T performs some initial processing to obtain $\bar{\mathbf{A}}, \bar{\mathbf{b}}$. It does this by drawing several pairs $(\mathbf{a}, b)$ from D, and keeping certain of them until it has accumulated a set of n pairs $\{(\bar{\mathbf{a}}_i, \bar{b}_i)\}$ that will make up $\bar{\mathbf{A}}, \bar{\mathbf{b}}$ in the natural way. With each new sample $(\mathbf{a}, b)$, T checks whether $\mathbf{a}$ is linearly independent modulo q of all those $\bar{\mathbf{a}}_i$ that have been kept so far; if so, $(\mathbf{a}, b)$ is kept, otherwise it is discarded. Note that the probability of keeping a particular sample is at least $\varphi(q)/q \geq 1/2$ (where φ denotes the Euler totient function), so with high probability, T accumulates the required n samples after drawing $O(n^2)$ samples from D. Now by construction, $\bar{\mathbf{A}}$ is invertible modulo q. Also observe that each sample is kept or discarded based only on its $\mathbf{a}$ component, so when $D = A_{\mathbf{s},\chi}$, we have $\bar{\mathbf{b}} = \bar{\mathbf{A}}^T\mathbf{s} + \bar{\mathbf{x}}$ where $\bar{\mathbf{x}}$ is drawn from χ^n.

The second stage actually transforms (fresh) samples from D into samples from a possibly different distribution. Given a draw $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ from D, T outputs $(\mathbf{a}', b') \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where

$$\mathbf{a}' = -\bar{\mathbf{A}}^{-1}\mathbf{a} \quad \text{and} \quad b' = b + \langle \mathbf{a}', \bar{\mathbf{b}} \rangle.$$

Observe that because $\bar{\mathbf{A}}$ is invertible modulo q and $\mathbf{a} \in \mathbb{Z}_q^n$ is uniform, $\mathbf{a}' \in \mathbb{Z}_q^n$ is uniform as well. We now consider the two cases for D. If $D = U$, then $(\mathbf{a}', b')$ is also distributed according to U, because $b \in \mathbb{Z}_q$ is uniform and independent of $\mathbf{a}$. If $D = A_{\mathbf{s},\chi}$, then $b = \langle \mathbf{a}, \mathbf{s} \rangle + x$ for some $x \leftarrow \chi$, so we have

$$b' = \langle \mathbf{a}, \mathbf{s} \rangle + x - \langle \bar{\mathbf{A}}^{-1}\mathbf{a}, \bar{\mathbf{A}}^T\mathbf{s} \rangle + \langle \mathbf{a}', \bar{\mathbf{x}} \rangle = \langle \mathbf{a}', \bar{\mathbf{x}} \rangle + x.$$

Therefore, $(\mathbf{a}', b')$ is distributed according to $A_{\bar{\mathbf{x}},\chi}$, as desired.

3.2 The Cryptosystem

We now define a KDM-secure cryptosystem based on the LWE problem. For technical reasons, our construction uses a *prime power* modulus $q = p^2$ of a certain size, with messages taken over $\mathbb{Z}_p$. (Other choices of $q = p^e$ are possible, but $q = p^2$ seems to correspond to the mildest underlying assumption.) Note that any element $v \in \mathbb{Z}_q$ may be written as $v = (v_1, v_0) \in \mathbb{Z}_p \times \mathbb{Z}_p$, where v_1 and v_0 are the most and least significant digits in the base-p representation of v, respectively, with the digits chosen from the set of residues $\{-\frac{p-1}{2}, \ldots, \frac{p-1}{2}\}$. Recall that by Lemma 1, the LWE distribution $A_{\mathbf{s},\chi}$ (for uniform $\mathbf{s} \in \mathbb{Z}_q^n$) is pseudorandom if the search problem $\mathsf{LWE}_{q,\chi}$ is hard, and if the error distribution χ is concentrated on $\{0\} \times \mathbb{Z}_p$ (which will be the case in our system, by design).

For simplicity, we start with a scheme that encrypts a single element of $\mathbb{Z}_p$ at a time, later extending it to an amortized version in Section 3.4. Our scheme is very similar to Regev's cryptosystem [46], with two main differences. First, the entries of the secret key $\mathbf{s} \in \mathbb{Z}_q^n$ are chosen from the (narrow) *error distribution* χ (rather than uniformly), so that they may be represented unambiguously as elements of the message space $\mathbb{Z}_p$ (see Lemma 3); this is secure due to Lemma 2. Second, we modify the encryption algorithm so that it induces a 'nice' distribution over ciphertexts (see Lemma 4). Specifically, the encryption algorithm chooses a random vector $\mathbf{r} \in \mathbb{Z}^m$ from a discrete Gaussian distribution (rather than from $\{0,1\}^m$), and adds a small extra term e to 'smooth out' the ciphertext distribution. These steps may be seen as applying Regev's main worst-case to average-case reduction [46] to the (already random) public key.

Construction 1 *The construction is parametrized by $q = p^2$ for some prime p, and an error parameter α; we instantiate these parameters below. Let $\chi = \bar{\Psi}_\alpha$, the discretized Gaussian over $\mathbb{Z}_q$.*

- Key generation: *The secret key is $\mathbf{s} \leftarrow \chi^n$. The public key is $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$, which is made up of $m \geq 2(n+1) \lg q$ draws $(\mathbf{a}_i, b_i)$ from $A_{\mathbf{s},\chi}$. That is, $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$ for independent $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{x} \leftarrow \chi^m$.*
- Encryption: *Before specifying the encryption algorithm, we define a distribution $E_{\mathbf{A},\mathbf{b}}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, which has parameters $r = \omega(\sqrt{\log m})$ and $r' = r \cdot \sqrt{m} \cdot (\alpha + \frac{1}{2q})$. The distribution is obtained by choosing $\mathbf{r} \leftarrow D_{\mathbb{Z}^m,r}$ and $e \leftarrow \bar{\Psi}_{r'}$ and outputting*

  $$(\mathbf{A}\mathbf{r}, \langle \mathbf{r}, \mathbf{b} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

  To encrypt a message $z \in \mathbb{Z}_p$ given the public key $(\mathbf{A}, \mathbf{b})$, draw a sample $(\mathbf{u}, v)$ from $E_{\mathbf{A},\mathbf{b}}$ and output the ciphertext $(\mathbf{u}, c = v + z \cdot p) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.
- Decryption: *To decrypt a ciphertext $(\mathbf{u}, c)$ given the secret key $\mathbf{s}$, output the $z \in \mathbb{Z}_p$ such that $z \cdot p$ is closest to $c - \langle \mathbf{u}, \mathbf{s} \rangle$ modulo q.*

The main constraints on the parameters are given by the correctness requirement (α cannot be too large) and the hardness requirement (α should be large enough to invoke the worst-case lattice connections of [46, 43]). These constraints are satisfied if the following inequalities hold true:

$$\frac{n}{q} = \frac{n}{p^2} \leq \alpha \leq \frac{1}{p \cdot \sqrt{m} \cdot \omega(\log n)} \qquad (1)$$

By routine calculations, it is possible to satisfy the above inequalities for $m = O(n \log n)$, $p = \tilde{O}(\sqrt{mn})$, and $\alpha = 1/\tilde{O}(m \cdot \sqrt{n})$. This yields an underlying worst-case approximation factor of $\tilde{O}(n/\alpha) = \tilde{O}(n^{2.5})$ for lattice problems such as GapSVP.

Theorem 2. *For parameters satisfying Equation (1), the above cryptosystem is KDM-secure with respect to the set of affine functions over $\mathbb{Z}_p$, assuming that $\mathsf{LWE}_{q,\chi}$ is hard.*

3.3 Proof of Security

Overview. The proof of Theorem 2 has the following structure. First we show completeness, including correct decryption of key-dependent messages. Next we prove KDM security in two main steps.

The first step is to show that the view of the adversary in the real attack game may be generated faithfully, up to negligible statistical distance, via an alternate game: starting from the distribution $A_{\mathbf{s},\chi}$ (for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$), the game invokes the transformation from Lemma 2 several times to produce independent distributions $A_{\mathbf{s}_1,\chi}, A_{\mathbf{s}_2,\chi}, \ldots$ for each user (where each $\mathbf{s}_i \leftarrow \chi^n$), and generates the users' public keys from these distributions in the natural way.

The transformation additionally outputs an invertible linear relation modulo q (hence modulo p as well) between each $\mathbf{s}_i$ and $\mathbf{s}$, thus linking every pair $\mathbf{s}_i, \mathbf{s}_j$ in a known way. The game answers the adversary's (key-dependent) message queries using these relations and the linear homomorphisms of the cryptosystem; this is where we use the fact that the system has a 'nice' ciphertext distribution. The crucial property of this game is that, aside from oracle access to $A_{\mathbf{s},\chi}$, the game works without needing to know any of the secret vectors $\mathbf{s}, \mathbf{s}_1, \mathbf{s}_2, \ldots$.

The second (and final) step is to consider a game that proceeds in exactly the same way as above, except that the original distribution $A_{\mathbf{s},\chi}$ is replaced by the *uniform* distribution $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$. Because the game uses only oracle access to its given distribution, the two games are computationally indistinguishable under the assumption that $\mathsf{LWE}_{q,\chi}$ is hard (and by Lemma 1). Moreover, all the public keys in this game are uniform and independent, which implies that all the simulated ciphertexts are as well (up to negligible statistical distance). It follows that the adversary has negligible advantage in this game, and the scheme is KDM-secure.

Abstract Properties. Here we state a few technical facts about the cryptosystem. The proof of security relies only on these abstract properties, which can be shown via routine application of Gaussians over lattices from prior works (e.g., [40, 46, 25]). Due to space limitations, we defer the proofs to the full version.

The first fact is that the entries of the secret key may be represented unambiguously in the message space $\mathbb{Z}_p$. For convenience in dealing with key-dependent messages, from now on we view the secret key $\mathbf{s}$ as an element of $\mathbb{Z}_p^n \subset \mathbb{Z}_q^n$.

Lemma 3. *An $s \leftarrow \chi$ is of the form $s = (0, s_0) \in \mathbb{Z}_p \times \mathbb{Z}_p$ with overwhelming probability.*

Proof. This follows directly from the upper bound on α from Equation (1) and the exponential tail bound on the Gaussian distribution.

The following lemmas characterize the ciphertext distribution, which is needed for showing correctness, and (more importantly) for producing proper key-dependent ciphertexts using the scheme's homomorphisms.

Lemma 4. *With overwhelming probability over the choice of the public key $(\mathbf{A}, \mathbf{b})$ for secret key $\mathbf{s}$, the distribution $E_{\mathbf{A},\mathbf{b}}$ is within negligible statistical distance of $A_{\mathbf{s},\bar{\Psi}_\beta}$ for some $\beta \leq \sqrt{2}r'$.*

Lemma 5. *Let $\mathbf{t} \in \mathbb{Z}_p^n$ and $y \in \mathbb{Z}_p$ be arbitrary. With overwhelming probability over the choice of the public key $(\mathbf{A}, \mathbf{b})$ for arbitrary secret key $\mathbf{s} \in \mathbb{Z}_p^n$, the following holds: for $(\mathbf{u}, v) \leftarrow E_{\mathbf{A},\mathbf{b}}$, the distribution of*

$$(\mathbf{u} - \mathbf{t} \cdot p, v + w \cdot p) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

is within negligible statistical distance of a (properly generated) encryption of the message $\langle \mathbf{t}, \mathbf{s} \rangle + w \in \mathbb{Z}_p$.

Finally, the next lemma is used for showing statistical security in the final hybrid game.

Lemma 6. *With overwhelming probability over the choice of a 'malformed' public key* $(\mathbf{A}, \mathbf{b})$ *from the* uniform *distribution* $U(\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m)$*, the distribution* $E_{\mathbf{A},\mathbf{b}}$ *is within negligible statistical distance of the uniform distribution* $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$*.*

Proof Details.

Correctness. By Lemma 4, the noise in the c component of a ciphertext is distributed according to $\bar{\Psi}_\beta$ for some

$$\beta \leq \sqrt{2}r' \leq 4\alpha\sqrt{m} \cdot \omega(\sqrt{\log n}) \leq \frac{1}{p \cdot \omega(\sqrt{\log n})},$$

by Equation (1). By the exponential tail inequality for Gaussians and the definition of $\bar{\Psi}_\beta$, the noise term does not exceed $q/2p = p/2$, except with negligible probability. We remark that the scheme can be made correct with probability 1 by modifying the key generation and encryption schemes to reject and re-sample values of $\mathbf{x}, \mathbf{r}, e$ that are 'too long;' however, this comes at the cost of an extra $\tilde{O}(\sqrt{n})$ factor in the noise parameter α and the underlying approximation factor for lattice problems.

The first hybrid game. We now describe an alternate game that faithfully simulates the true KDM attack game, up to negligible statistical distance. The game starts with access to the distribution $A_{\mathbf{s},\chi}$ for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$. For each user i, it applies the transformation described in Lemma 2 (using fresh draws from $A_{\mathbf{s},\chi}$) to produce the distribution $A_{\mathbf{s}_i,\chi}$, where $\mathbf{s}_i$ is distributed according to χ^n. As a side-effect, the transformation also outputs invertible square matrices $\bar{\mathbf{A}}_i \in \mathbb{Z}_q^{n \times n}$ and vectors $\bar{\mathbf{b}}_i \in \mathbb{Z}_q^n$ such that for all i,

$$\mathbf{s} = \bar{\mathbf{A}}_i^{-T}(\bar{\mathbf{b}}_i - \mathbf{s}_i) \bmod q.$$

Note that by setting the right-hand sides equal for any i, j and reducing modulo p, we have

$$\bar{\mathbf{A}}_i^{-T}(\mathbf{s}_i - \bar{\mathbf{b}}_i) = \bar{\mathbf{A}}_j^{-T}(\mathbf{s}_j - \bar{\mathbf{b}}_j) \bmod p \iff \mathbf{s}_i = \bar{\mathbf{A}}_{i,j}^T \cdot \mathbf{s}_j + \bar{\mathbf{b}}_{i,j} \bmod p, \quad (2)$$

where $\bar{\mathbf{A}}_{i,j} = \bar{\mathbf{A}}_j^{-1}\bar{\mathbf{A}}_i$ and $\bar{\mathbf{b}}_{i,j} = \bar{\mathbf{b}}_i - \bar{\mathbf{A}}_{i,j}^T \cdot \bar{\mathbf{b}}_j$. The game then generates a public key $(\mathbf{A}_i, \mathbf{b}_i)$ for each user i in the usual way by drawing m samples from $A_{\mathbf{s}_i,\chi}$.

We now describe how the game answers (key-dependent) message queries. Suppose the adversary requests an encryption, under the jth user's public key $(\mathbf{A}_j, \mathbf{b}_j)$, of the function $f_{\mathbf{t},w}(\mathbf{s}_i) = \langle \mathbf{t}, \mathbf{s}_i \rangle + w \in \mathbb{Z}_p$ (for some $\mathbf{t} \in \mathbb{Z}_p^n, w \in \mathbb{Z}_p$) applied to the ith user's secret key $\mathbf{s}_i$. Observe that

$$f_{\mathbf{t},w}(\mathbf{s}_i) = \langle \mathbf{t}, \mathbf{s}_i \rangle + w = (\underbrace{\bar{\mathbf{A}}_{i,j} \cdot \mathbf{t}}_{\mathbf{t}' \in \mathbb{Z}_p^n})^T \cdot \mathbf{s}_i + \underbrace{\langle \mathbf{t}, \bar{\mathbf{b}}_{i,j} \rangle + w}_{w' \in \mathbb{Z}_p}.$$

The game therefore draws a sample $(\mathbf{u}, v) \leftarrow E_{\mathbf{A}_j,\mathbf{b}_j}$ and outputs

$$(\mathbf{u} - \mathbf{t}' \cdot p, v + w' \cdot p) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

This completes the description of the game.

By the above description and Lemmas 2 and 5, the following claim is apparent.

Claim. The views of the adversary in the real attack game and in the hybrid game are within negligible statistical distance.

The final hybrid game. The last hybrid game proceeds exactly as the one above, except that the initial distribution $A_{\mathbf{s},\chi}$ is replaced with $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$. Note that the game above only treats $A_{\mathbf{s},\chi}$ as an oracle (it never uses $\mathbf{s}$ directly), so $A_{\mathbf{s},\chi}$ may be replaced in this way.

Now by Lemma 2, all the public keys $(\mathbf{A}_i, \mathbf{b}_i)$ generated by the game are uniform and independent. Moreover, by Lemma 6, all the (key-dependent) message queries are answered by ciphertexts that are uniform and independent of the message. The next claim follows, and the proof of Theorem 2 is complete.

Claim. Assuming that $\mathsf{LWE}_{q,\chi}$ is hard, the two hybrid games are computationally indistinguishable. Moreover, the adversary's advantage in the final hybrid game is negligible.

3.4 Amortized Extension

The system described in Section 3.2 encrypts only a single element $z \in \mathbb{Z}_p$ per syndrome $\mathbf{u} \in \mathbb{Z}_q^n$, so the ciphertext is a factor at least n larger than the message, and the encryption algorithm performs at least $n \cdot m$ operations per message element. Peikert, Vaikuntanathan, and Waters [44] proposed a significantly more efficient amortized version of the cryptosystem, which can encrypt $\ell = O(n)$ symbols using only about twice the time and space as the basic scheme. We can show that a variant of that system is also KDM-secure.

Construction 3 *Just as in Construction 1, the scheme is parametrized by $q = p^2$ and $\alpha \in (0, 1)$.*

- Key generation: *The secret key is $\mathbf{S} \leftarrow \chi^{n\times\ell}$. The public key is $(\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{m\times\ell}$ for $m \geq 2(n + \ell)\lg q$, where $\mathbf{B} = \mathbf{A}^T\mathbf{S} + \mathbf{X}$ for independent $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{X} \leftarrow \chi^{m\times\ell}$.*
- Encryption: *first define the distribution $E_{\mathbf{A},\mathbf{B}}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q^\ell$, obtained as follows: choose $\mathbf{r} \leftarrow D_{\mathbb{Z}^m,r}$ where $r = \omega(\sqrt{\log m})$, choose $\mathbf{e} \leftarrow \bar{\Psi}_{r'}^\ell$ where $r' = r \cdot \sqrt{\ell \cdot m} \cdot (\alpha + \frac{1}{2q})$, and output $(\mathbf{u}, \mathbf{v}) = (\mathbf{A}\mathbf{r}, \mathbf{B}^T\mathbf{r} + \mathbf{e})$.* Note that the parameter r' is a $\sqrt{\ell}$ factor larger than in Construction 1. *To encrypt a message $\mathbf{z} \in \mathbb{Z}_p^\ell$, draw $(\mathbf{u}, \mathbf{v}) \leftarrow E_{\mathbf{A},\mathbf{B}}$ and output $(\mathbf{u}, \mathbf{c} = \mathbf{v} + \mathbf{z} \cdot p)$.*
- Decryption: *output the $\mathbf{z} \in \mathbb{Z}_p^\ell$ such that $\mathbf{z} \cdot p$ is closest to $\mathbf{c} - \mathbf{S}^T\mathbf{u}$ modulo q.*

The proof of security extends to this construction in a straightforward way, with the exception of Lemma 4, which characterizes the ciphertext distribution and allows the simulator to answer key-dependent message queries faithfully. By generalizing the techniques from [46, Corollary 3.10] to higher dimensions, we can prove the following fact about $E_{\mathbf{A},\mathbf{B}}$, which suffices for proving KDM security.

Lemma 7. *With overwhelming probability over the choice of the public key* $(\mathbf{A}, \mathbf{B})$ *for secret key* $\mathbf{S}$*, the distribution* $E_{\mathbf{A},\mathbf{B}}$ *is within negligible statistical distance of* $(\mathbf{u}, \mathbf{S}^T\mathbf{u} + \mathbf{v})$*, where* $\mathbf{u} \in \mathbb{Z}_q^n$ *is uniform and* $\mathbf{v} \in \mathbb{Z}_q^\ell$ *is drawn from some distribution that depends only on* $\mathbf{B}$ *(and not on* $\mathbf{u}$*).*

Proof (Proof sketch). We need to show that the distribution of $\mathbf{X}^T\mathbf{r} + \mathbf{e} \in \mathbb{Z}_q^\ell$ conditioned on $\mathbf{A}\mathbf{r} = \mathbf{u}$ is essentially the same for every fixed $\mathbf{u}$. We can show that the distribution is a (discretized) *non-spherical* Gaussian whose covariance matrix depends only on r' and the positive semidefinite Gram matrix $\mathbf{X}^T\mathbf{X}$. The proof relies on the fact that a (continuous) Gaussian can be decomposed into the sum of two Gaussians whose covariance matrices sum to that of the original, and also uses the partial ordering of positive semidefinite matrices to establish a sufficient lower bound for r' (this is where the extra $\sqrt{\ell}$ term arises).

4 Linear-Stretch PRG in Quasi-Linear Time

4.1 Overview

Our starting point is a simple pseudorandom generator which was originally suggested in [10]. Let $G(\mathbf{A}, \mathbf{s}, r) = (\mathbf{A}, \mathbf{A}\mathbf{s} + e(r))$, where $\mathbf{A} \in \mathbb{Z}_2^{m\times n}, \mathbf{s} \in \mathbb{Z}_2^n$ and $e(\cdot)$ is a noise sampling procedure that uses a random input r to sample a random error vector from $\mathsf{Ber}_\varepsilon^m$. It was shown in [10] that, assuming the hardness of LPN_ε, the output distribution of G is pseudorandom. (See also [10, 22, 46, 36, 6]). In order to get expansion the noise-sampling algorithm should use a seed r of length shorter than m. Indeed, the noise vector can be sampled by using a seed r whose length is roughly $H_2(\varepsilon) \cdot m$, where H_2 is the binary entropy function. This gives an additive expansion of $m(1 - H_2(\varepsilon)) - n$ which is positive when the rate n/m is smaller than $1 - H_2(\varepsilon)$.

The resulting PRG is quite efficient as it mainly uses bit-operations rather than costly arithmetic operations over large fields. However, it still does not bring us to our goal (quasilinear time PRG). The main problem is that the matrix-vector product requires $\Omega(mn)$ operations, and so the time complexity of the generator is (at least) proportional to the product of the output length m and the security parameter n.

To solve this problem, we exploit the fact that the matrix $\mathbf{A}$ is public and hence can be reused with many different information words $\mathbf{s}_1, \ldots, \mathbf{s}_\ell$. Hence, the modified generator will compute the product of an $m \times n$ matrix $\mathbf{A}$ with an $n \times \ell$ matrix $\mathbf{S}$, and will add a noisy bit to each of the entries of the matrix $\mathbf{AS}$. By choosing ℓ carefully, we can use algorithms for fast rectangular matrix multiplication to speed up the computation.

We should also show how to sample the noise vector in quasilinear time *without using too many random bits*. At first glance, this seems to be hard, and indeed, we are not aware of any such sampling procedure[8]. However, we can bypass this problem by using a fast sampling procedure suggested in [5]. This procedure Sam samples an m-length noise vector e by using more than m random bits. To compensate this loss Sam also outputs a "leftover" vector – a vector v which is almost-random even when e is given. This allows us to concatenate v to the output of the PRG.

4.2 The Construction

The following lemma shows that for a random matrix $\mathbf{A}$, the mapping $(\mathbf{s}, e) \mapsto \mathbf{A}\mathbf{s} + e$ is pseudorandom even when it is applied to polynomially-many random strings $\mathbf{s}_1, \ldots, \mathbf{s}_\ell$. The proof combines the ideas of [10] with a standard hybrid argument and is therefore omitted from this version.

Lemma 8. *Let $0 < \varepsilon < \frac{1}{2}$ be a noise parameter and let $m(n), \ell(n)$ be arbitrary polynomials. If LPN_ε is hard, then the distribution $(\mathbf{A}, \mathbf{A}\cdot\mathbf{S}+\mathbf{E})$ is pseudorandom, where $\mathbf{A} \leftarrow \mathsf{U}_{m(n)\times n}$, $\mathbf{S} \leftarrow \mathsf{U}_{n\times\ell(n)}$, and $\mathbf{E} \leftarrow \mathsf{Ber}_\varepsilon^{m(n)\times\ell(n)}$.*

The following fact is based on [16].

Fact 4 *For every $r \leq 0.172$ the product of a matrix in $\mathbb{Z}_2^{m\times m^r}$ and a matrix in $\mathbb{Z}_2^{m^r\times m}$ can be computed by a circuit of size $\tilde{O}(m^2)$.*

We will use a sampling procedure due to [5].

Lemma 9 (implicit in [5]). *There exist positive integers $k > 1$ and $c > 2k$, and a sampling algorithm* Sam *that uses $(k + k/c)N$ random bits and outputs a pair of strings $(\mathbf{e}, v)$ whose joint distribution is $2^{-\Omega(N)}$ statistically-close to $(\mathsf{Ber}_{2^{-k}}^N, \mathsf{U}_{kn})$. Moreover,* Sam *can be implemented in $\mathbf{NC^0}$ and therefore by a circuit family of size $O(N)$.*

We can now present our construction.

Construction 5 *Let $N = n^{12}$. Let k, c and* $\mathrm{Sam} : \{0,1\}^{(k+k/c)N} \to \{0,1\}^N \times \{0,1\}^{kN}$ *be the constants and sampling algorithm promised by Lemma 9. Let $e(r)$ and $v(r)$ denote the first and second entries of* $\mathrm{Sam}(r)$*. Define the function*

$$G(\mathbf{A}, \mathbf{S}, r) \overset{def}{=} (\mathbf{A}, \mathbf{A}\cdot\mathbf{S} + e(r), v(r))$$

where, $\mathbf{A} \in \mathbb{Z}_2^{n^6\times n}$, $\mathbf{S} \in \mathbb{Z}_2^{n\times n^6}$, $r \in \{0,1\}^{(k+k/c)N}$, $e(r)$ is parsed as a matrix in $\mathbb{Z}_2^{n^6\times n^6}$, and matrix addition is computed entry-wise.

Theorem 6. *Assuming that $\mathsf{LPN}_{2^{-k}}$ is hard, the function G defined in Construction 5 is a PRG with linear-stretch that can be computed by a circuit family of size quasilinear in the output length.*

[8] For example, the time complexity of the noise-sampling procedure of [22] is quadratic in the length of the error vector (for a constant error rate).

Proof. It can be easily verified that G takes less than $(k+0.6)N$ input bits and outputs more than $(k+1)N$ bits, and therefore, the stretch is linear in the input length. Pseudorandomness follows by Lemmas 9 and 8 as the tuple $(\mathbf{A}, \mathbf{A}\cdot\mathbf{S} + e(r), v(r))$ is statistically-indistinguishable from the tuple $(\mathbf{A}, \mathbf{A}\cdot\mathbf{S}+e(r), \mathsf{U}_{kn^7})$, which, in turn, is computationally-indistinguishable from $\mathsf{U}_{n^{4.5}+n^7+kn^7}$. Finally, by Fact 4, Lemma 9, and since entry-wise addition of two matrices is computable by linear-size circuits, the generator G can be computed by a circuit-family of size $\tilde{O}(N)$. □

5 Weak Randomized PRF

An efficiently computable *randomized* function family $F : \{0,1\}^n \times \{0,1\}^{m(n)} \rightarrow \{0,1\}^{s(n)}$ is called a *randomized weak pseudorandom function* (RWPRF) if it satisfies the following:

- (weak pseudorandomness) For every polynomial $p(\cdot)$ the sequence

$$\big(A_1, F_S(A_1), \ldots, A_{p(n)}, F_S(A_{p(n)})\big) \text{ is pseudorandom,}$$

  where $S \leftarrow \mathsf{U}_n$ and $(A_1, \ldots, A_{p(n)}) \leftarrow (\mathsf{U}_m)^{p(n)}$ and fresh internal randomness is used in each evaluation of F_S.
- (verifiability) There exists an efficient equality-tester algorithm V such that

$$\Pr[V(Y_1, Y_2) = \mathsf{equal}] > 1 - \text{negl}(n)$$
$$\Pr[V(Y_1, Y_2') = \mathsf{not\text{-}equal}] > 1 - \text{negl}(n),$$

  where $S \leftarrow \mathsf{U}_n, A \leftarrow \mathsf{U}_m, A' \leftarrow \mathsf{U}_m, Y_1 \leftarrow F_S(A), Y_2 \leftarrow F_S(A)$, and $Y_2' \leftarrow F_S(A')$.

The PRG construction from the previous section, suggests a simple implementation of RWPRF. We let $\mathbf{S} \in \mathbb{Z}_2^{n\times\ell(n)}$ be the secret key of the function family, and let $\mathbf{A} \in \mathbb{Z}_2^{m(n)\times n}$ be the argument on which the function is being evaluated. The randomized function is defined as $f_{\mathbf{S}}(\mathbf{A}) = \mathbf{AS} + \mathbf{E}$ where $\mathbf{E} \in \mathsf{Ber}_\varepsilon^{m(n)\times\ell(n)}$ is a secret error vector which is randomly chosen in each invocation. By Lemma 8, the resulting function family is pseudorandom when it is evaluated on randomly chosen inputs $\mathbf{A}_1, \ldots, \mathbf{A}_q$. Also, given $y = f_{\mathbf{S}}(\mathbf{A})$ and $y = f_{\mathbf{S}}(\mathbf{B})$, one can easily check, with overwhelming probability, whether $\mathbf{A}$ and $\mathbf{B}$ are equal, even without knowing the key $\mathbf{S}$.

Note that now we have no limitation on the amount of randomness used to generate the error matrix $\mathbf{E}$. Hence, we can rely on the hardness of, say $\mathsf{LPN}_{1/4}$, and generate the error matrix $\mathbf{E}$ by taking the entry-wise product of 2 random matrices. The resulting function is quite efficient, and can be computed by a depth two Boolean circuit of size $O(n\ell m)$, or, by a circuit of size $\tilde{O}(m\ell)$ for a proper choice of the parameters. (The first option uses the trivial circuit for matrix multiplication, and the latter relies on Fact 4.)

When $\ell(n) = m(n)$ the function is symmetric, that is, one can replace the role of the argument and the key without violating the pseudorandomness property.

We also note that when $\ell(n)$ is sufficiently large (e.g., $\ell(n) > n/(1 - H_2(\varepsilon))$), then, except with negligible probability, $\mathbf{S}$ forms an error correcting code whose distance is larger than ε. In this case, the function $f_{\mathbf{S}}$ is injective and the equality-tester works well with respect to *every* input (as long as the collection-key and the internal randomness are random). By symmetry this is also true when the argument $\mathbf{A}$ is viewed as the key of the function. Hence, a random pair $(\mathbf{A}, f_{\mathbf{S}}(\mathbf{A}))$ forms a commitment to the collection key $\mathbf{S}$, which might be useful in some contexts.

Oblivious evaluation protocol. In an oblivious evaluation protocol for a collection of functions $f_{\mathbf{S}}$, one party (Alice) holds a key $\mathbf{S}$ and another party (Bob) holds a point $\mathbf{A}$. At the end of the protocol, Bob learns the value $f_{\mathbf{S}}(\mathbf{A})$, while Alice learns nothing. One can also consider the symmetric variant of the problem in which Alice learns $f_{\mathbf{S}}(\mathbf{A})$ and Bob learns nothing. In our setting, we also assume that the party who does not get the output selects the internal randomness of the function. That is, we consider the task of securely computing the following functionalities $g((\mathbf{S}, \mathbf{E}), \mathbf{A})) = (\lambda, \mathbf{AS}+\mathbf{E})$ and $h(\mathbf{S}, (\mathbf{A}, \mathbf{E})) = (\mathbf{AS}+\mathbf{E}, \lambda)$ where λ denotes the empty string. We give an efficient and secure protocol for evaluating both g and h Our protocol employs one-out-of-two oblivious transfer (OT) [20] for strings of length m. Such a protocol allows a receiver to receive one of two m-bit strings held by the sender in an oblivious way, that is, without revealing which string is selected.

Lemma 10. *There exists a constant-round protocol for securely evaluating f which uses circuits of size $O(m\ell n)$ with ℓn oracle gates to oblivious transfer which supports strings of length m.*

Proof. The protocol is similar to the protocol suggested in [23] for obliviously evaluating the Naor-Reingold PRF [42].

We begin with the version in which Alice receives the value of $f_{\mathbf{S}}(\mathbf{A})$. Let $\mathbf{S}$ be Alice's input and $\mathbf{A}, \mathbf{E}$ be Bob's input. For each $i \in [\ell]$ invoke in-parallel the following sub-protocol where $\mathbf{s}$ (resp. $\mathbf{e}$) is the i-th column of $\mathbf{S}$ (resp. $\mathbf{E}$):

- Bob chooses a random matrix $\mathbf{R} \leftarrow \mathbb{Z}_2^{m(n)\times n}$.
- For each $j \in [n]$ Alice and Bob call the string-OT oracle with Alice as the receiver and Bob as sender in the following way. Alice's input is $\mathbf{s}_j$, the j-th bit of $\mathbf{s}$, and Bob's input is the pair $(\mathbf{R}_j, \mathbf{R}_j + \mathbf{A}_j)$, where $\mathbf{R}_j$ and $\mathbf{A}_j$ are the j-th columns of $\mathbf{R}$ and $\mathbf{A}$. In addition, Bob sends the sum $\mathbf{t} = \mathbf{e} + \sum_j \mathbf{R}_j$.
- Alice sums up (over $\mathbb{Z}_2^{m(n)}$) the $n + 1$ vectors she received and outputs the result which is equal to $\sum_{\mathbf{s}_j=1} \mathbf{A}_j + \mathbf{e}$.

It is not hard to see that the protocol securely evaluates the functionality h. Indeed, the view of Alice which consists of the values learned by the OT and the vector $\mathbf{t}$ can be easily sampled given $f_{\mathbf{S}}(\mathbf{A}; \mathbf{E})$. A protocol in which Bob receives the output can be derived by slightly changing the previous protocol. Details omitted. □

Comparison to the OPRF of [23]. Let us briefly compare the efficiency of our scheme to the standard instantiation of OPRF [23] which is based on the Naor-Reingold function [42]. Our scheme uses large number of OT calls – if we set ℓ to be 1, which does not affect the security of the construction, this number is linear in the security parameter n. In contrast, the FIPR scheme uses only $O(m)$ calls where m is the length of the *input*. On the other hand, the additional overhead of FIPR is m modular multiplications and a single exponentiation, where our protocol performs only m vector additions ($O(mn)$ bit-wise XORs). This tradeoff is interesting as, by using the batch-OT protocol of [33], OT operations cost almost as little as symmetric operations. Furthermore, by using standard techniques one can compute all the OT operations in a preprocessing stage. In such case, it seems that the current scheme has the potential to obtain better performance, at least in some usage scenarios. (This possibility deserves further study.)

Application. Oblivious evaluation of pseudorandom function was recently used by Hazay and Lindell [30] to obtain an efficient two-party protocol for secure set-intersection (an explicit version for the semi-honest model appears in [23]). Our construction can be used in their protocol whenever the inputs of the parties are randomly distributed. This restriction is natural in some scenarios (e.g., when the inputs are names of entities or keys that were randomly selected by some authority) and can be always obtained at the expense of using a random oracle. We also note that RWPRF can be used to derive an identification scheme: we let parties share a key for the RWPRF and verify the identity of a party by querying the value of the function on a random point. When this protocol is instantiated with our function we get the well known HB protocol [32]. (This view is implicit in [36].)

6 Fast Circular-Secure Symmetric Encryption

6.1 The Construction

We now construct a symmetric encryption scheme. Our construction can be viewed as using the previous weak, randomized PRF in an analogous way to the standard construction of symmetric encryption from PRF, except that to deal with the error introduced by the PRF randomization we need to make the message redundant. This is done by employing an additional efficiently decodable error correcting code. As mentioned before, a similar construction was suggested in [26].

Let $\ell = \ell(n)$ be a message-length parameter which is set to be an arbitrary polynomial in the security parameter n. (Shorter messages are padded with zeroes.) Let $\varepsilon = 2^{-k}$ and $0 < \delta < 1$ be constants. We will use a family of good binary linear codes with information words of length $\ell(n)$ and block length $m = m(n)$, that has an efficient decoding algorithm D that can correct up to $(\varepsilon + \delta) \cdot m$ errors. We let $\mathbf{G} = \mathbf{G}_\ell$ be the $m \times \ell$ binary generator matrix of this family and we assume that it can be efficiently constructed (given 1^n).

Construction 7 *Let $N = N(n)$ be an arbitrary polynomial (which controls the tradeoff between the key-length and the time complexity of the scheme). The private key of the scheme is a matrix $\mathbf{S}$ which is chosen uniformly at random from $\mathbb{Z}_2^{n\times N}$.*

- Encryption*: To encrypt a message $\mathbf{M} \in \mathbb{Z}_2^{\ell\times N}$, choose a random $\mathbf{A} \leftarrow \mathbb{Z}_2^{m\times n}$ and a random noise matrix $\mathbf{E} \leftarrow \mathsf{Ber}_\varepsilon^{m\times N}$. Output the ciphertext*
$$(\mathbf{A}, \mathbf{A}\cdot\mathbf{S} + \mathbf{E} + \mathbf{G}\cdot\mathbf{M}).$$
- Decryption*: Given a ciphertext $(\mathbf{A}, \mathbf{Z})$ apply the decoding algorithm D to each of the columns of the matrix $\mathbf{Z} - \mathbf{AS}$ and output the result.*

Observe that the decryption algorithm errs only when there exists a column in $\mathbf{E}$ whose Hamming weight is larger than $(\varepsilon + \delta)m$, which, by Chernoff Bound, happens with negligible probability.

Quasilinear-time implementation. To get a quasilinear time implementation (for sufficiently long messages), we instantiate the above scheme with the error-correcting codes of Spielman [47, Thm. 19] which maps ℓ bits to $m = \Theta(\ell)$ bits with constant relative-distance and with the property that the encoding can be computed via a circuit of size $O(\ell)$ and the decoding can be decoded by a circuit of size $O(\ell \log \ell)$. Hence, the complexity of encryption (and decryption) is dominated by the complexity of the product $\mathbf{A}\cdot\mathbf{S}$. (The error matrix $\mathbf{E}$ can be generated in linear time by taking the entry-wise product of k random matrices $\mathbf{R}^{(1)}, \ldots, \mathbf{R}^{(k)} \leftarrow \mathbb{Z}_2^{m\times N}$.) To compute this product in quasilinear time we set $N = n^6$ and assume that $m = \Omega(n^6)$, *i.e.,* assume that the message length $N\cdot\ell$ is at least $\Omega(n^{12})$. In this case, by Fact 4, the encryption and decryption can be computed by a circuit of size $\tilde{O}(N\ell)$.

Useful properties. The scheme enjoys several useful "homomorphic properties" which follow from its linear structure. In particular, given an encryption $(\mathbf{A}, \mathbf{Y})$ of an unknown message $\mathbf{M}$ under an unknown key $\mathbf{S}$, one can transform it to an encryption $(\mathbf{A}', \mathbf{Y}')$ of $\mathbf{M}+\mathbf{M}'$ under the key $\mathbf{S}+\mathbf{S}'$, for any given $\mathbf{M}', \mathbf{S}'$. This is done by letting $\mathbf{A}' = \mathbf{A}$ and $\mathbf{Y}' = \mathbf{Y} + \mathbf{AS}' + \mathbf{GM}'$. Furthermore, if the message $\mathbf{M}$ is the all zeroes string, then it is possible to convert the ciphertext $(\mathbf{A}, \mathbf{Y})$ to be an encryption $(\mathbf{A}', \mathbf{Y}')$ of the key $\mathbf{S}$ itself or, more generally, to be an encryption of $\mathbf{T}\cdot\mathbf{S}$ for an arbitrary linear transformation $\mathbf{T} \in \mathbb{Z}_2^{\ell\times n}$. This is done by letting $\mathbf{Y}' = \mathbf{Y}$ and $\mathbf{A}' = \mathbf{A} + \mathbf{G}\cdot\mathbf{T}$. Indeed, in this case $\mathbf{Y}' = \mathbf{A}'\mathbf{S} + \mathbf{E} + \mathbf{G}(\mathbf{TS})$. By choosing $\mathbf{T}$ to be the $\begin{pmatrix}\mathbf{I}_n \\ \mathbf{0}_{\ell-n\times n}\end{pmatrix}$, we can get an encryption of the key itself (padded with zeroes). We summarize these properties in the following lemma.

Lemma 11. *There exist efficiently computable transformations f, g, h such that for every* unknown $\mathbf{S} \in \mathbb{Z}_2^{n\times N}$ *and* $\mathbf{M} \in \mathbb{Z}_2^{\ell\times N}$ *and known* $\mathbf{S}' \in \mathbb{Z}_2^{n\times N}, \mathbf{M}' \in \mathbb{Z}_2^{\ell\times N}$ *and* $\mathbf{T} \in \mathbb{Z}_2^{\ell\times n}$*: $f(\mathbf{M}', \mathsf{Enc}_{\mathbf{S}}(\mathbf{M})) \equiv \mathsf{Enc}_{\mathbf{S}}(\mathbf{M} + \mathbf{M}')$, $g(\mathbf{S}', \mathsf{Enc}_{\mathbf{S}}(\mathbf{M})) \equiv \mathsf{Enc}_{\mathbf{S}+\mathbf{S}'}(\mathbf{M})$, and $h(\mathbf{T}, \mathsf{Enc}_{\mathbf{S}}(0^{\ell\times N})) \equiv \mathsf{Enc}_{\mathbf{S}}(\mathbf{TS})$, where $\mathsf{Enc}_{\mathbf{K}}(\mathbf{A})$ denotes a random encryption of the message $\mathbf{A}$ under the key $\mathbf{K}$.*

6.2 KDM Security

From now on, we fix the parameters $N(\cdot)$, $\ell(\cdot)$, $m(\cdot)$ and ε of our scheme. We consider the class of affine transformations that map the i-th column of the key $\mathbf{S}$ to the i-th column of the message $\mathbf{M}$. Let $t = t(n)$ be some arbitrary polynomial and let $N = N(n)$ and $\ell = \ell(n)$. For a matrix $\mathbf{T} \in \mathbb{Z}_2^{\ell \times n}$, a matrix $\mathbf{B} \in \mathbb{Z}_2^{\ell \times N}$ and an integer $i \in [t]$ we define the function $f_{T,B,i}$ which maps a tuple of t keys $(\mathbf{S}_1, \ldots, \mathbf{S}_t) \in (\mathbb{Z}_2^{n \times N})^t$ to a message $\mathbf{M} \in \mathbb{Z}_2^{\ell \times N}$ by letting $\mathbf{M} = \mathbf{T} \cdot \mathbf{S}_i + \mathbf{B}$. We let $\mathcal{F}_{\ell,N,t} = \{f_{\mathbf{T},\mathbf{B},i} | \mathbf{T} \in \mathbb{Z}_2^{\ell \times n}, \mathbf{B} \in \mathbb{Z}_2^{\ell \times N}, i \in [t]\}$. We will prove KDM-CPA-security with respect to the class $\mathcal{F}_{\ell,N,t}$. Formally,

Theorem 8. *Suppose that the* LPN_ε *is hard. Then Construction 7 is CPA-KDM secure with respect to* $\mathcal{F}_{\ell,N,t}$ *for every polynomial* $t(\cdot)$*.*

The proof uses the properties described in Lemma 11 in a straightforward way. A similar proof outline is used in [13].

Proof (Sketch). CPA security follows easily from Lemma 8. To prove KDM security, we show how to transform an adversary that wins the KDM game (with respect to $\mathcal{F}_{\ell,N,t}$) into an adversary that wins the standard CPA-game. Let $\mathbf{S}$ be the key of the scheme that was chosen by the challenger in the CPA game. The idea is to choose t random offsets $\mathbf{S}'_i \leftarrow U_{n \times N}$ and emulate the KDM game where the i-th key is $\mathbf{S}_i = \mathbf{S}'_i + \mathbf{S}$. Now, by using the properties of Lemma 11, we can transform a ciphertext $\mathsf{Enc}_{\mathbf{S}}(0^{\ell \times N})$ into a ciphertext $\mathsf{Enc}_{\mathbf{S}_j}(\mathbf{T} \cdot \mathbf{S}_i + \mathbf{B})$ for any given $i, j, \mathbf{T}$ and $\mathbf{B}$. Hence, we can perfectly emulate answers to the queries asked by the KDM adversary. □

As shown in [8], we can use the standard encrypt-then-MAC transformation to upgrade the security to KDM-CCA security (with respect to $\mathcal{F}_{\ell,N,t}$). In [34], it is shown that the existence of a linear-time computable MAC scheme follows from the existence of any one-way function. Hence, the intractability of LPN_ε allows us to construct a KDM-CCA-secure symmetric cryptosystem in which encryption and decryption are performed in quasilinear time, and the length of the ciphertext is linear in the message length.

7 Acknowledgement.

The first author would like to thank Boaz Barak and Shai Halevi for helpful discussions, and Yuval Ishai for many valuable comments on an earlier draft of this paper.

References

1. P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In *ESORICS*, pages 374–396, 2005.
2. M. Albrecht, G. Bard, and W. Hart. Efficient multiplication of dense matrices over gf(2). *CoRR*, abs/0811.1714, 2008.

3. M. Alekhnovich. More on average case vs approximation complexity. In *Proc. 44th FOCS*, pages 298–307, 2003.
4. B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM J. Comput.*, 36(4):845–888, 2006. Preliminary version in Proc. 45th FOCS, 2004.
5. B. Applebaum, Y. Ishai, and E. Kushilevitz. On pseudorandom generators with linear stretch in NC^0. In *Proc. 10th Random.*, 2006.
6. B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography with constant input locality. In *Advances in Cryptology: Proc. of CRYPTO '07*, 2007. full version in http://www.cs.princeton.edu/ bappelba/pubs/input-locality-full.pdf.
7. M. Backes, M. Dürmuth, and D. Unruh. OAEP is secure under key-dependent messages. In *ASIACRYPT*, pages 506–523, 2008.
8. M. Backes, B. Pfitzmann, and A. Scedrov. Key-dependent message security under active attacks - BRSIM/UC-soundness of symbolic encryption with key cycles. In *CSF*, pages 112–124, 2007.
9. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography*, pages 62–75, 2002.
10. A. Blum, M. L. Furst, M. J. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *CRYPTO*, pages 278–291, 1993.
11. A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.
12. A. Bogdanov and M. C. Mertens. A parallel hardware architecture for fast gaussian elimination over gf(2). In *FCCM '06: Proceedings of the 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 237–248, Washington, DC, USA, 2006. IEEE Computer Society.
13. D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In *CRYPTO*, pages 108–125, 2008.
14. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. Cryptology ePrint Archive, Report 2008/375, 2008.
15. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT*, pages 93–118, 2001.
16. D. Coppersmith. Rapid multiplication of rectangular matrices. *SICOMP: SIAM Journal on Computing*, 11, 1982.
17. I. B. Damgård and J. B. Nielsen. An efficient pseudo-random generator with applications to public-key encryption and constant-round multiparty computation. Unpublished, 2002.
18. N. Dedic, L. Reyzin, and S. P. Vadhan. An improved pseudorandom generator based on hardness of factoring. In *Proc. 3rd SCN*, pages 88–101, 2002.
19. Y. Dodis, Y. T. Kalai, and S. Lovett. Cryptography with auxiliary inputs. In *Proc. 41st STOC*, 2009.
20. Even, Goldreich, and Lempel. A randomized protocol for signing contracts. *CACM: Communications of the ACM*, 28, 1985.
21. V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. New results for learning noisy parities and halfspaces. In *FOCS*, pages 563–574, 2006.
22. J.-B. Fischer and J. Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *Proc. EuroCrypt '96*, volume 1070, pages 245–255, 1996.
23. Freedman, Ishai, Pinkas, and Reingold. Keyword search and oblivious pseudorandom functions. In *Theory of Cryptography Conference (TCC), LNCS*, volume 2, 2005.
24. R. Gennaro. An improved pseudo-random generator based on the discrete logarithm problem. *J. Cryptology*, 18(2):91–110, 2005.

25. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.
26. H. Gilbert, M. J. B. Robshaw, and Y. Seurin. How to encrypt with the LPN problem. In *ICALP (2)*, pages 679–690, 2008.
27. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. of the ACM*, 33:792–807, 1986.
28. I. Haitner and T. Holenstein. On the (im)possibility of key dependent encryption. In *TCC*, pages 202–219, 2009.
29. S. Halevi and H. Krawczyk. Security under key-dependent inputs. In *CCS '07*, pages 466–475, 2007.
30. C. Hazay and Y. Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In *TCC '08*, pages 155–175, 2008.
31. D. Hofheinz and D. Unruh. Towards key-dependent message security in the standard model. In *EUROCRYPT*, pages 108–126, 2008.
32. N. J. Hopper and M. Blum. Secure human identification protocols. In *ASIACRYPT*, pages 52–66, 2001.
33. Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious transfers efficiently. In *Advances in Cryptology: Proc. of CRYPTO '03*, pages 145–161, 2003.
34. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography with constant computational overhead. In *Proc. 40th STOC*, 2008.
35. A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology: Proc. of CRYPTO '05*, pages 293–308, 2005.
36. J. Katz and J. S. Shin. Parallel and concurrent security of the HB and HB^+ protocols. In *EUROCRYPT*, pages 73–87, 2006.
37. A. R. Klivans and A. A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *FOCS*, pages 553–562, 2006.
38. M. Krause and S. Lucks. On the minimal hardware complexity of pseudorandom function generators (extended abstract). In *Proc. 18th STACS*, volume 2010 of *LNCS*, pages 419–430, 2001.
39. D. Micciancio. Improving lattice based cryptosystems using the Hermite normal form. In *CaLC*, pages 126–145, 2001.
40. D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.
41. D. Micciancio and O. Regev. Lattice-based cryptography. In *Post Quantum Cryptography*, pages 147–191. Springer, February 2009.
42. M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudorandom functions. *J. ACM*, 51(2):231–262, 2004. Preliminary version in Proc. 38th FOCS, 1997.
43. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, 2009.
44. C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571, 2008.
45. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, 2008.
46. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.
47. D. A. Spielman. Linear-time encodable and decodable error-correcting codes. In *Proc. 27th STOC*, pages 388–397, 1995.