Fast Cryptographic Primitives Based on the Hardness of Decoding Random Linear Code

PRELIMINARY TECHNICAL REPORT

Benny Applebaum*

Abstract

Current cryptographic constructions typically involve a large multiplicative computational overhead that grows with the desired level of security. Recently, at STOC 2008, Ishai, Kushilevitz, Ostrovsky, and Sahai (IKOS) suggested the possibility of implementing cryptographic primitives, while incurring only a constant computational overhead compared to insecure implementations of the same tasks. Surprisingly, Ishai et al showed that such highly efficient cryptographic constructions can be realized, under plausible, yet nonstandard, intractability assumptions.

In this paper, we show that if one is willing to accept polylogarithmic computational overhead, many constructions can be achieved under *standard* assumptions. Specifically, assuming the hardness of *decoding random linear code* (or equivalently, hardness of *learning parity with noise*), we get the following results.

1. A pseudorandom generator $G : \{0,1\}^n \to \{0,1\}^{2n}$ which doubles its input length and can be computed in quasilinear time $\tilde{O}(n) = n \cdot \mathrm{polylog}(n)$.
2. A construction of weak randomized pseudorandom function – a relaxation of standard PRF – which can be obliviously computed in quasilinear time. This is far more efficient than previously known constructions, such as the oblivious evaluation of the Naor-Reingold PRF (FOCS 1997).
3. A symmetric encryption scheme whose encryption and decryption algorithms are computable by circuits of quasilinear size (in the message length). Our scheme provides security against key-dependent messages and achieves circular security. This provides a highly efficient alternative in the private-key setting to the circular-secure *public-key* encryption scheme of Boneh, Halevi, Hamburg, and Ostrovsky (CRYPTO 2008).

By combining our results with previous ones, we get fast implementations of various other primitives and protocols.

*Department of Computer Science, Princeton University, benny.applebaum@gmail.com . Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797.

1 Introduction

Improving the efficiency of cryptographic construction is an important goal motivated by both practical and theoretical interests. Typical cryptographic functions introduce a multiplicative computational overhead that grows with the desired level of security. Recently, Ishai et. al. [IKOS08] showed that many cryptographic tasks can be implemented while incurring only a constant computational overhead compared to insecure implementations of the same tasks. These results were based on plausible, yet nonstandard, intractability assumptions. The purpose of this paper is to derive similar results under a standard assumption. We show that this is possible at the expense of slightly relaxing the efficiency requirement. Namely, we show that if one strives for *polylogarithmic* computational overhead, it is possible to derive some of the results of [IKOS08] as well as other new results, under the well-known assumption that random binary linear codes cannot be decoded in polynomial-time.

1.1 Our Assumption: Hardness of Decoding Random Linear Code

Our results are based on the intractability of decoding a random linear code. In the following we introduce and formalize this assumption.

For code length parameter $m = m(n)$, and noise parameter $0 < \mu < \frac{1}{2}$, we will consider the following "decoding game" $\text{DECODE}(m, \mu)$. Pick a random binary matrix $C \in \mathbb{F}_2^{m \times n}$ representing a linear code, and a random information word $x \in \mathbb{F}_2^n$. Encode x with C and transmit the resulting codeword $y = Cx$ over a binary symmetric channel in which every bit is flipped with probability μ. Output the noisy codeword $\hat{y}$ along with the code's description C. The adversary's task is to find the information word x. We say that $\text{DECODE}(m, \mu)$ is intractable if every polynomial-time adversary wins in the above game with no more than negligible probability in n.

Assumption 1.1 (Hardness of Decoding Random Linear Code (DRLC))**.** *For every constant* $0 < \mu < \frac{1}{2}$ *and polynomial* $m(\cdot)$*, the* $\text{DECODE}(m, \mu)$ *game is intractable.*

The hardness of $\text{DECODE}(m, \mu)$ is well studied [BFKL94, Kea98, HB01, BKW03, Lyu05, JW05, FGKP06]. It can be also formulated as the problem of learning parity with noise, and it is known to be NP-complete in the worst-case [BMvT78]. Assumptions similar to the DRLC assumption were put forward in [GKL93, BFKL94, Gol01, HB01, JW05, KS06, AIK07]. The plausibility of such an assumption is supported by the fact that a successful attack would imply a major breakthrough in coding theory. We mention that the best known algorithm for $\text{DECODE}(m, \mu)$, due to Blum et al. [BKW03], runs in time $2^{O(n/\log n)}$ and requires m to be $2^{O(n/\log n)}$. Lyubashevsky [Lyu05] showed how to reduce m to be only super-linear, i.e., $n^{1+\alpha}$, at the cost of increasing the running time to $2^{O(n/\log\log n)}$.

All the results of this paper follow from the DRLC assumption. In fact, we can rely on a relaxed version of DRLC in which the noise rate is set to some universal constant (*e.g.* , $1/8$). Moreover, for some of our applications it suffices to assume DRLC with respect to some fixed polynomial code length (*e.g.* , $m(n) = n^6$). Finally, we emphasize that DRLC does *not* assume exponential hardness.

1.2 Our Results

Assuming the DRLC assumption, we get the following results.

1.2.1 Quasilinear-time pseudorandom generator

We construct a pseudorandom generator (PRG) which doubles its input length and can be computed by a Boolean circuit of size $\tilde{O}(n)$. This is considerably faster than previous constructions of linear-stretch PRGs (*e.g.* ,[FS96, DRV02, DN02, Gen05]) which suffer from a *polynomial* overhead. To the best of our knowledge, the only exception is the construction of [AIK06] which is computable by linear-size ($\mathbf{NC^0}$) circuit. This construction is based on a plausible, yet non standard, assumption of Alekhnovich [Ale03]. Roughly speaking, this assumption suggests that a noisy random codeword of a code with *sparse* generating matrix is pseudorandom. This assumption is relatively new and, while seems reasonable, it has not been widely studied yet. Moreover, unlike our DRLC assumption, Alekhnovich's assumption assumes *pseudorandomness* rather than *one-wayness* – which seems to be less appealing.

In [IKOS08] it is shown how to construct several fast cryptographic primitives based on a linear-stretch PRG with low overhead. (These reductions are originally instantiated with the PRG construction of [AIK06].) By plugging our PRG to these reductions 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 exists).[1]

1.2.2 Randomized weak pseudorandom function

An efficiently computable function collection $\{f_k\}$ is a pseudorandom function family [GGM86] if a random member f_k of the family cannot be distinguished from a truly random function even when the distinguisher gets an evaluation oracle. Randomized weak pseudorandom functions (RWPRFs) relax this notion 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.

We construct an RWPRF which can be computed by a circuit of size quasilinear in the input length. Moreover, we describe a secure protocol that allows to obliviously evaluate the function in quasilinear time in the semi-honest model.[2] This is considerably more efficient than previous protocols for oblivious evaluation of PRFs (*e.g.* , the protocol of [FIPR05] for the Naor-Reingold PRF [NR04]). In some settings, our construction can be used to obtain a fast protocol for secure set intersection via the protocol of [HL08].

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

[2] Such a protocol allows two parties, one holding a point x and another holding a key k, to evaluate the function f_k on x without learning each other's inputs.

It should be mentioned that [IKOS08] constructs *linear-time* computable PRF under the minimal assumption that one-way functions exists. Moreover, they give a general linear-time computable protocol for securely evaluating *every* two-party functionality. However, their protocol relies on the existence of a *polynomial* stretch PRG which can be computed in $\mathbf{NC^0}$. This is a relatively strong assumption whose validity still deserves more study.[3]

1.2.3 Circular-secure symmetric encryption

We construct symmetric encryption scheme whose encryption and decryption algorithms are computable by circuits of quasilinear size (in the message length). Our scheme provides security against key-dependent messages attacks (KDM) [BRS02]; that is, it remains secure even when the adversary is allowed to see messages that depend on the secret keys in use via any fixed linear function. Moreover, our scheme is "circular secure" [CL01]: it remains secure under a "key cycle" (or even a "key clique") usage, where we have n users with n different keys and each key is encrypted under all the other ones. Such usage scenarios arise in key-management systems, in the context of anonymous credential systems, and in the context of "axiomatic security" (See [BHHO08]).

The notions of KDM and circular security were extensively studied for both symmetric and public-key encryption schemes [CL01, BRS02, HK07, BPS07, BHHO08, CCS08, BDU08, HU08, HH08]. Without resorting to the use of random oracles, constructing an encryption scheme (either in the private-key or public-key setting) that is circular secure was a long-standing open problem. This question was recently solved by Boneh et al. [BHHO08] who constructed such a *public-key* encryption scheme. Our scheme is much faster than the scheme of [BHHO08] and hence it provides a highly efficient alternative in the private-key setting.

We also note that our scheme (as well as the scheme of [BHHO08]) resists a restricted family of key-related attacks, where the adversary is allowed to ask for encryptions under several different unknown keys whose pairwise sums are fixed by the adversary. Such an encryption scheme was needed by Ishai et al. [IKNP03] in order to obtain a protocol for reducing the amortized complexity of oblivious transfer (OT). The original protocol used a random oracle to construct such a scheme. Our new scheme can be used to remove the random oracle and obtain an efficient amortization of OTs in the standard model. Finally, we mention that our scheme was used in [AIK08] to obtain encryption scheme with constant latency.

A related scheme. We recently learned that a scheme similar to ours was independently discovered by Dodis et al. [DKL]. Their variant is shown to be secure when the adversary is allowed to see arbitrary function $f(k)$ of the secret key k as long as the function is exponentially hard to invert. We stress that this notion of security is incomparable to the notions studied here (*i.e.,* circular security, security under key-dependent messages, and security under key related attacks). Moreover, the results of [DKL] rely ona non-standard assumption which seems stronger than the DLRC assumption. Finally, their scheme inherently adds a polynomial overhead in both computation and communication.

[3] Some recent evident which weakly support this assumption can be found in [ABW08].

2 Linear-Stretch PRG in Quasi-Linear Time

2.1 Overview

Our starting point is a simple pseudorandom generator which was originally suggested in [BFKL94]. Let $G(C, x, r) = (C, Cx + e(r))$, where $C \in \mathbb{F}_2^{m \times n}, x \in \mathbb{F}_2^n$ and $e(\cdot)$ is a noise sampling procedure that uses a random input r to sample an m-length vector of noise rate μ. It was shown in [BFKL94] that, under the DRLC assumption, the output distribution of G is pseudorandom. (See also [BFKL94, FS96, Reg05, KS06, AIK07]). 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 $H_2(\mu) \cdot m$; this gives an expansion of $m(1 - H_2(\mu)) - n$ which is positive when the rate n/m is smaller than $1 - H_2(\mu)$.

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 C is public and hence can be reused with many different information words $x_1, \ldots, x_\ell$. Hence, the modified generator will compute the product of an $m \times n$ matrix C with an $n \times \ell$ matrix X, and will add a noisy bit to each of the entries of the matrix CX. 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.[4] However, we can bypass this problem by using a fast sampling procedure suggested in [AIK06]. This procedure S samples an m-length noise vector e by using more than m random bits. To compensate this loss S also outputs a "change"– a vector v which is almost-random even when e is given. This allows us to concatenate v to the output of the PRG.

2.2 The Construction

The following lemma shows that for a random code C, the mapping $(x, e) \mapsto Cx + e$ is pseudorandom even when it is applied to polynomially-many random information words $x_1, \ldots, x_\ell$.

Lemma 2.1. *Let $m(n)$ be a code length parameter, $0 < \mu < \frac{1}{2}$ be a noise parameter and let $\ell(n)$ be arbitrary polynomial. If Assumption DRLC holds then the distribution $\mathcal{D}_{m,\mu,\ell} \stackrel{def}{=} (C, C \cdot X + E)$ is pseudorandom, where $C \stackrel{R}{\leftarrow} U_{m(n)\times n}$, $X \stackrel{R}{\leftarrow} U_{n\times \ell(n)}$, and $E \in \{0,1\}^{m(n)\times \ell(n)}$ is a random error matrix of noise rate μ.*

Proof. Assume, towards a contradiction, that there exists an efficient distinguisher $\{A_n\}$ which distinguishes $(C, CX + E)$ from $(C, U_{m(n)\times \ell(n)})$ with advantage $\varepsilon(n)$. We use $\{A_n\}$ to break the

[4] For example, the time complexity of the noise-sampling procedure of [FS96] is $\Theta(m^2 \log m)$, where m is the length of the error vector and the error rate is constant.

pseudorandomness of $\mathcal{D}_{m,\mu,1}$ and this way, by [BFKL94], derive a contradiction. Fix n and let $m = m(n), \ell = \ell(n)$ and $\mu = \mu(n)$. For a matrix $C \in \{0,1\}^{m\times n}$ let $\mathcal{D}_{m,\mu,\ell}(C)$ denote the distribution $C \cdot X + E$ where X, E are distributed as in the statement of the lemma.

Given a pair $C \in \{0,1\}^{m\times n}$ and $y \in \{0,1\}^n$, our algorithm B_n samples an $m \times \ell$ matrix T, invokes A_n on the pair (C, T) and outputs the result. The matrix $T = (T_L, y, T_R)$ is constructed as follows: B_n chooses a random index $i \stackrel{R}{\leftarrow} [\ell]$, and samples a matrix $T_L \stackrel{R}{\leftarrow} \mathcal{D}_{m,\mu,i}(C)$, and a matrix $T_R \stackrel{R}{\leftarrow} U_{m\times(\ell-i-1)}$.

By a standard hybrid argument, B_n distinguishes $\mathcal{D}_{m,\mu,1}$ from the uniform distribution with advantage ε/ℓ. Specifically, define $\ell + 1$ hybrids $\mathcal{H}_0, \ldots, \mathcal{H}_\ell$ where $\mathcal{H}_i$ consists of a pair $C \stackrel{R}{\leftarrow} U_{m(n)\times n}$ and an $m \times \ell$ matrix $M = (M_L, M_R) \stackrel{R}{\leftarrow} (\mathcal{D}_{m,\mu,i}(C), U_{m\times(\ell-i)})$. Clearly, $\mathcal{H}_0 \equiv U_{mn+m\ell}$ and $\mathcal{H}_\ell \equiv \mathcal{D}_{m,\mu,\ell}$. Hence, for $i \stackrel{R}{\leftarrow} [\ell]$, the adversary A distinguishes $\mathcal{H}_{i-1}$ from $\mathcal{H}_i$ with (expected) advantage ε/ℓ. The proof follows by noting that the mapping $(C, y) \mapsto (C, T)$ computed by B takes the uniform distribution U_{nm+m} to $\mathcal{H}_i$ and the pseudorandom distribution $\mathcal{D}_{m,\mu,1}$ to $\mathcal{H}_{i+1}$, for randomly chosen i. □

The following fact is based on [Cop82].

Fact 2.2. *For every $r \leq 0.172$ the product of a matrix in $\mathbb{F}_2^{m\times m^r}$ and a matrix in $\mathbb{F}_2^{m^r\times m}$ can be computed by a circuit of size $\tilde{O}(m^2)$.*

We will use a sampling procedure due to [AIK06].

Lemma 2.3 (implicit in [AIK06])**.** *There exist positive integers k and $c > 2k$, and a sampling algorithm S that uses $(k+k/c)N$ random bits and outputs a pair of strings (e, v) where e is an N-bit error-vector of noise rate 2^{-k}, the vector v is of length kN, and the statistical distance between (e, v) and (e, U_{kn}) is at most $2^{-\Omega(N)}$. Moreover, S can be implemented in $\mathbf{NC^0}$ and therefore by a circuit family of size $O(N)$.*

We can now present our construction.

Construction 2.4. *Let $N = n^{12}$. Let k, c and $S : \{0,1\}^{(k+k/c)N} \to \{0,1\}^N \times \{0,1\}^{kN}$ be the constants and sampling algorithm promised by Lemma 2.3. Let $e(r)$ and $v(r)$ denote the first and second entries of $S(r)$. Define the function*

$$G(C, X, r) \stackrel{def}{=} (C, C \cdot X + e(r), v(r))$$

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

Theorem 2.5. *Let G be the function defined in Construction 2.4. Then, assuming the DRLC assumption, G is a PRG with linear-stretch. Furthermore, the generator G can be computed by a circuit family of size quasilinear in the output length.*

Proof. We begin by verifying that G has linear stretch. Indeed, G takes $2n^{4.5} + (k + k/c)n^7 < (k+0.6)N$ input bits and outputs more than $(k+1)N$ bits. Hence, the stretch is linear (in the input length). Pseudorandomness follows by Lemmas 2.3 and 2.1 as

$$(C, C \cdot X + e(r), v(r)) \stackrel{s}{\equiv} (C, C \cdot X + e(r), U_{kn^7}) \stackrel{c}{\equiv} U_{n^{4.5}+n^7+kn^7}.$$

Finally, by Fact 2.2, Lemma 2.3, 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 $\tilde{O}(N)$. □

3 Weak Randomized PRF and Oblivious Evaluation Protocol

The PRG construction suggests a simple implementation of a weak form of pseudorandom function family [GGM86] (PRF). We let $X \in \mathbb{F}_2^{n \times \ell(n)}$ be the secret key of the function family, and let $C \in \mathbb{F}_2^{m(n) \times n}$ be the argument on which the function is being evaluated. The randomized function is defined as $f_X(C) = CX + E$ where $E \in \mathbb{F}_2^{m(n) \times \ell(n)}$ is a secret error vector of rate μ which is randomly chosen in each invocation. The resulting function family is pseudorandom when it is evaluated on randomly chosen inputs $C_1, \ldots, C_q$. Although the function is randomized one can easily verify whether y and y' are images of the same input C under the same (possibly unknown) key X. This can be done by checking that the two strings y and y' are close in Hamming distance. We formalize the above properties by defining the notion of weak randomized PRFs.

Definition 3.1. *Let $F : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^{s(n)}$ be an efficiently computable randomized function family. We say that F is a* randomized weak pseudorandom function *(RWPRF) if*

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

$$\big(C_1, F_X(C_1), \ldots, C_{p(n)}, F(C_{p(n)})\big) \text{ is pseudorandom},$$

  where $X \stackrel{R}{\leftarrow} U_n$ and $(C_1, \ldots, C_{p(n)}) \stackrel{R}{\leftarrow} (U_m)^{p(n)}$ and fresh internal randomness is used in each evaluation of F_X.

- *(verifiability) There exists an efficient algorithm V such that*

$$\Pr[V(Y_1, Y_2) = \textit{equal}] > 1 - \mathrm{neg}(n) \tag{1}$$
$$\Pr[V(Y_1, Y_2') = \textit{not-equal}] > 1 - \mathrm{neg}(n), \tag{2}$$

  where $X \stackrel{R}{\leftarrow} U_n, C \stackrel{R}{\leftarrow} U_m, C' \stackrel{R}{\leftarrow} U_m, Y_1 \stackrel{R}{\leftarrow} F_X(C), Y_2 \stackrel{R}{\leftarrow} F_X(C)$, and $Y_2' \stackrel{R}{\leftarrow} F_X(C')$.

The following lemma easily follows from Lemma 2.1.

Lemma 3.2. *Suppose that the DRLC assumption holds. Then, for every polynomials $m(\cdot), \ell(\cdot)$ and constant $\mu < \frac{1}{2}$ the function family $f_X(C) = C \cdot X + E$ defined above is a RWPRF.*

By choosing $\ell(n) = m(n) = n^{3.5}$ one can obtain quasilinear efficient evaluation of f. Moreover, in this case the function is symmetric, that is, one can replace the role of the argument and the key without violating the pseudorandomness property.

Oblivious evaluation protocol. In an oblivious evaluation protocol for a collection of functions f_x, one party (Alice) holds a key x and another party (Bob) holds a point C. At the end of the protocol, Bob learns the value $f_x(C)$, while Alice learns nothing. One can also consider the symmetric variant of the problem in which Alice learns $f_x(A)$ 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 $g((X,E),C)) = (\lambda, XC+E)$ and $h(X,(C,E)) = (XC+E,\lambda)$ where λ denotes the empty string. We give an efficient and secure protocol for evaluating both g and h in the semi-honest model.[5] Our protocol employs one-out-of-two oblivious transfer (OT) [EGL85] 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 3.3. *There exists a 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 [FIPR05] for obliviously evaluating the Naor-Reingold PRF [NR04].

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

- Bob chooses a random matrix $R \stackrel{R}{\leftarrow} \mathbb{F}_2^{m(n)\times n}$.
- For each $j \in [n]$ Alice and Bob run string-OT protocol in which Alice is the receiver with input x_j and Bob's input is the pair $(R_j, R_j + C_j)$ where x_j is the j-th bit of x and R_j is the j-th column of R. In addition, Bob sends the sum $t = e + \sum_j R_j$.
- Alice sums up (over $\mathbb{F}_2^{m(n)}$) the $n+1$ vectors she received and outputs the result which is equal to $\sum_{x_j=1} C_j + e$.

It is not hard to see that the protocol securely evaluates the functionality h. Indeed, if the OT is fully simulatable then one can easily sample the view of Alice given $f_X(C;E)$. To derive a protocol in which Bob receives the output we slightly change the previous protocol. In each iteration Bob selects an additional one time pad $s \stackrel{R}{\leftarrow} \mathbb{F}_2^{m(n)}$ and instead of sending $e + \sum_j R_j$ it sends $s + \sum_j R_j$. Then, we add a final round in which Alice sums up all the vectors she received together with e (the i-th column of the noise matrix which is now held by Alice) and sends the result w back to Bob. The output of Bob is computed by adding the pad s to the received vector w. Assuming the security of the OT, Alice's view consists of randomly chosen vectors, while Bob's view can be easily generated from its input and $f_X(C;E)$. □

[5] In fact, the protocol is also secure in the malicious setting (assuming that the OT provides such security). However, this setting does not make sense for weak PRFs as even in the ideal world an active adversary (either Alice or Bob) can learn the other player's secret input by carefully selecting its own (non-random) inputs. (E.g., by letting X or C be the identity matrix.)

By using the ideas of [IKOS08] we can use our quasilinear PRG and any standard OT to obtain a string OT that supports strings of length m with time complexity $\tilde{O}(m)$ for sufficiently large m. (Alternatively, one can use the reduction described in Section 4.3 to implement t calls to m-bit OT in time $\tilde{O}(mt)$.) Hence, if we set $\ell = 1$ (which does not affect the security of the construction) we get the following corollary:

Corollary 3.4. *Assuming the DRLC assumption and the existence of an Oblivious Transfer protocol, there exists a weak randomized pseudorandom function $f_x(C; e) = Cx + e$ which can be obliviously evaluated by constant-round protocol computed by circuits of size $\tilde{O}(|C|)$ (i.e., quasi-linear in the input length).*

Application. Oblivious evaluation of pseudorandom function was recently used by Hazay and Lindell [HL08] to obtain an efficient two-party protocol for secure set-intersection. 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. The use of our weak PRF achieves significant speed up in comparison to the existing instantiation which is based on the Naor-Reingold PRF [NR04].

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 [HB01]. (This view is implicit in [KS06].)

4 Fast Circular-Secure Encryption

4.1 The Construction

Syntactic definition of symmetric encryption. Symmetric encryption scheme consists of three probabilistic-polynomial time algorithms (G, E, D), where G is a key generation algorithm which given a security parameter 1^n outputs a secret-key k (without loss of generality, the secret-key is a randomly-chosen string k of length $\mathrm{poly}(n)$); E is a randomized encryption algorithm that takes a message m and a secret key k and outputs a ciphertext c; and D is a randomized decryption algorithm that takes a ciphertext c and a secret key k and outputs a plaintext m'. Correctness requires that decryption succeeds except with negligible probability in n.

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 $\mu = 2^{-k}$ 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 $(\mu + \delta) \cdot m$ errors. We let $G = G_\ell$ be the $m \times \ell$ binary generator matrix of this family and we assume that it can be efficiently constructed (given 1^n).

Construction 4.1. *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 X which is chosen uniformly at random from $\mathbb{F}_2^{n\times N}$.*

- ***Encryption**: To encrypt a message $M \in \mathbb{F}_2^{\ell\times N}$, choose a random $C \stackrel{R}{\leftarrow} \mathbb{F}_2^{m\times n}$ and k random matrices $R^{(1)}, \ldots, R^{(k)} \stackrel{R}{\leftarrow} \mathbb{F}_2^{m\times N}$. Let $E \in \mathbb{F}_2^{m\times N}$ be the entry-wise product of $R^{(1)}, \ldots, R^{(k)}$,* i.e., *, $E_{i,j} = \prod_t R^{(t)}_{i,j}$. Output the ciphertext*

$$(C, C \cdot X + E + G \cdot M).$$

- ***Decryption**: Given a ciphertext (C, Z) apply the decoding algorithm D to each of the columns of the matrix $Z - CX$ and output the result.*

Observe that the decryption algorithm errs only when there exists a column in E whose Hamming weight is larger than $(\mu + \delta)m$, 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 [Spi95, 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 $C \cdot X$. 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 2.2, 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 follows from its linear structure. In particular, given an encryption (C, Y) of an unknown message M under an unknown key X, one can transform it to an encryption (C', Y') of $M + M'$ under the key $X + X'$, for any given M', X'. This is done by letting $C' = C$ and $Y' = Y + CX' + GM'$. Furthermore, if the message M is the all zeroes string, then it is possible to convert the ciphertext (C, Y) to be an encryption (C', Y') of the key X itself or, more generally, to be an encryption of $T \cdot X$ for an arbitrary linear transformation $T \in \mathbb{F}_2^{\ell\times n}$. This is done by letting $Y' = Y$ and $C' = C + G \cdot T$. Indeed, in this case $Y' = C'X + E + G(TX)$. By choosing 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). Note that by using the first transformation we can apply the second transformation even if M is an arbitrary message as long as it is *known*. We summarize these properties in the following lemma.

Lemma 4.2. *There exist efficiently computable transformations f, g, h such that for every* unknown *$X \in \mathbb{F}_2^{n\times N}$ and $M \in \mathbb{F}_2^{\ell\times N}$ and known $X' \in \mathbb{F}_2^{n\times N}, M' \in \mathbb{F}_2^{\ell\times N}$ and $T \in \mathbb{F}_2^{\ell\times n}$:*

1. *$f(M', E_X(M)) \equiv E_X(M + M')$,*
2. *$g(X', E_X(M)) \equiv E_{X+X'}(M)$,*

3. $h(T, E_X(0^{\ell \times N})) \equiv E_X(TX)$,

where $E_K(A)$ denotes a random encryption of the message A under the key K (i.e., $E_K(A)$ is a random variable induced by the internal randomness of the encryption).

4.2 KDM-Security

The standard definition of CPA-security asserts that no efficient adversary can distinguish between the real encryption function $E(k, \cdot)$ and a fake encryption function which always returns an encryption of some fixed dummy message, *e.g.* $E(k, 0^\ell)$. The standard definition of CCA-security is similar except that the adversary has also an oracle access to the decryption algorithm (but is not allowed to query it on an output of the encryption oracle). KDM security extends these definitions by allowing the adversary to ask for encryptions of key-dependent messages $m = f(k)$. Following [HK07, BHHO08], we define key-dependence relative to a fixed set of functions $\mathcal{C}$. The following definition is taken from [BHHO08], and is based on the definitions of [BRS02, HK07].

Let $t > 0$ be an integer and let $\mathcal{C}$ be a set of functions $\mathcal{C} = \{f : K^t \to M\}$, where K is the key-space and M is the message space. (Formally, the sets $\mathcal{C}, K, M$ are all indexed by the security parameter.) We assume that $K \subset M$ and that for every $f \in \mathcal{C}$ the output length of f is independent of its input. KDM security is defined with respect to $\mathcal{C}$ via the following game that takes place between a challenger and an adversary A. For an integer $t > 0$ and a security parameter n the game proceeds as follows:

- **Initialization.** The challenger randomly chooses a bit $b \xleftarrow{R} \{0, 1\}$ and t secret keys $k_1, \ldots, k_t$ by invoking $G(1^n)$ for t times.
- **Queries.** The adversary asks for polynomially-many queries where each query is of the form (i, f) where $i \in [t]$ and $f \in \mathcal{C}$. The challenger computes $y = f(k_1, \ldots, k_t)$ and outputs

$$c \xleftarrow{R} \begin{cases} E(k_i, y) & \text{if } b = 0, \\ E(k_i, 0^{|y|}) & \text{if } b = 1. \end{cases}$$

- **Final phase.** The adversary attempt to guess b and outputs a bit $b' \in \{0, 1\}$.

Definition 4.3. ***(KDM-secure encryption)*** *Let $\mathcal{C}$ be a class of functions and t be an integer. A symmetric encryption scheme (G, E, D) is t-way* CPA-KDM-secure *with respect to $\mathcal{C}$ if no polynomial-time attacker A has non-negligible advantage in guessing the value of the bit b in the above game (where the running time and the advantage are measured as functions of the security parameter n). The definition of t-way* CCA-KDM-secure *with respect to $\mathcal{C}$ is similar except that the adversary has also an oracle access to the decryption function $D(k, \cdot)$ (but cannot query this oracle on any output given to him by the encryption oracle).*

We say that a function class $\mathcal{C} = \{f : K^t \to M\}$ is *non-trivial*, if is contains: (1) all $|M|$ constant functions $f_m(k) = m$ for all $m \in M$; and (2) all t selector functions $f_i(k_1, \ldots, k_n) = k_i$

for $i \in [t]$. It was observed in [BHHO08] that CPA-KDM-security (respectively, CCA-KDM-security) with respect to such non-trivial function class implies standard CPA-security (respectively CCA-security), since the constant functions let the adversary obtain the encryption of any message of its choice. The selector functions imply circular security (or even "clique"-security) since they let the adversary obtain $E(k_i, k_j)$ for all $i, j \in [t]$.

From now on, we fix the parameters of our scheme, that is we let $N(\cdot)$, $\ell(\cdot)$ and $m(\cdot)$ be some (arbitrary) polynomials, let μ be a constant noise rate and assume that we use some appropriate family of error correcting code.

Affine-transformation. We consider the class of affine transformations that map the i-th column of the key X to the i-th column of the message M. Let $t = t(n)$ be some arbitrary polynomial and let $N = N(n)$ and $\ell = \ell(n)$. For a matrix $T \in \mathbb{F}_2^{\ell \times n}$, a matrix $B \in \mathbb{F}_2^{\ell \times N}$ and an integer $i \in [t]$ we define the function $f_{T,B,i}$ which maps a tuple of t keys $(X_1, \ldots, X_t) \in (\mathbb{F}_2^{n \times N})^t$ to a message $M \in \mathbb{F}_2^{\ell \times N}$ by letting $M = T \cdot X_i + B$. We let $\mathcal{C}_{\ell,N,t} = \left\{ f_{T,B,i} | T \in \mathbb{F}_2^{\ell \times n}, B \in \mathbb{F}_2^{\ell \times N}, i \in [t] \right\}$.

We will prove KDM-CPA-security with respect to the class $\mathcal{C}_{\ell,N,t}$. Formally,

Theorem 4.4. *Suppose that the DRLC assumption holds. Then Construction 4.1 is $t(n)$-way CPA-KDM-secure with respect to $\mathcal{C}_{\ell,N,t}$ for every polynomial $t(\cdot)$.*

The proof uses the properties described in Lemma 4.2 in a straightforward way. A similar (yet more complicated) proof is used in [BHHO08].

Proof. Let $t = t(n)$ be some arbitrary polynomial. Consider a t-way CPA-KDM attack in which an attacker A makes $q(n)$ queries and breaks the scheme with probability $1/2 + \delta(n)$. We use A to break the pseudorandomness of the distribution $\mathcal{D}_{qm,\mu,N}$ (defined in Lemma 2.1) as follows:

- **Input**: a challenge $(C, Y) \in \mathbb{F}_2^{qm \times n} \times \mathbb{F}_2^{qm \times N}$ supposedly chosen from $\mathcal{D}_{qm,\mu,N}$ or from the uniform distribution.
- **Preprocessing**: Parse the challenge to q pairs $(C_1, Y_1), \ldots, (C_q, Y_q)$ where $(C_i, Y_i) \in \mathbb{F}_2^{m \times n} \times \mathbb{F}_2^{m \times N}$. Toss a coin $b \xleftarrow{R} \{0, 1\}$ and randomly choose $X'_1, \ldots, X'_t \xleftarrow{R} \mathbb{F}_2^{n \times N}$. (In our emulation we think of the i-th key as $X_i = X + X'_i$ where X is the random seed used to produce the challenge.)
- **Invoke the adversary** A**:** Let the e-th query of A be $(j, f_{T,B,i})$. To answer the query we think of (C_e, Y_e) as an encryption of the all-zero matrix under X. If $b = 1$ use the transformation g of Lemma 4.2 together with X'_j to convert (C_e, Y_e) into an encryption of the all-zeroes matrix under the key $X_j = X'_j + X$, and output the result. Otherwise, if $b = 0$ compute $h(T, (C_e, Y_e))$ and get (C'_e, Y'_e) which, in our mental experiment, is an encryption of $E_X(TX)$. Then, use the transformations f and g together to add X'_j to the key and $TX'_i + B$ to the message and output the result. (In our mental experiment the result should be $E_{X+X'_j}(TX + TX'_i + B) = E_{X_j}(TX_i + B)$).

- **Termination**: Let b' be the output of A. Check if it is equal to b and if so announce "pseudorandom"; otherwise announce "random".

It is not hard to verify the following claims.

Claim 4.5. *Suppose that $(C, Y) \stackrel{R}{\leftarrow} \mathcal{D}_{qm,\mu,N}$. Then, the joint distribution of the bit b together with the view of the adversary in the above experiment is identical to the joint distribution of the bit b together with the view of the adversary in a real attack on the scheme.*

Proof. If $(C, Y) \stackrel{R}{\leftarrow} \mathcal{D}_{qm,\mu,N}$ then (C_e, Y_e) is indeed an encryption of the all-zeroes string under a single random key X. Hence, by Lemma 4.2, we answer all the adversary's queries properly (exactly as we do in the real attack). □

Claim 4.6. *Suppose that $(C, Y) \stackrel{R}{\leftarrow} \mathbb{F}_2^{qm\times n} \times \mathbb{F}_2^{qm\times N}$. Then the view of the adversary in the above experiment conditioned on $b = 0$ is distributed identically to the view of the adversary conditioned on $b = 1$.*

Proof. In this case, each (C_e, Y_e) is randomly and independently chosen. Fix the queries of the adversary. The claim follows from two observations: (1) we answer the e-th query by $(C_e + C'_b, Y_e + Y'_b)$ where C'_b and Y'_b depend on b; and (2) (C_e, Y_e) is used only in the e-th query. Hence, the adversary sees uniformly chosen answers regardless of its queries or the choice of b. □

It follows that we distinguish $\mathcal{D}_{qm,\mu,N}$ from the uniform distribution with advantage δ. This, by Lemma 2.1, implies that δ should be negligible. □

We can now prove the following corollary:

Corollary 4.7. *Suppose that the DRLC assumption holds. Then, there exists polynomials ℓ, N and a symmetric-key encryption scheme in which encryption and decryption are performed in quasilinear time (in the message length), and the length of the ciphertext is linear in the message length. Furthermore, for every polynomial $t(\cdot)$ the scheme is $t(n)$-way CCA-KDM-secure with respect to $\mathcal{C}_{\ell,N,t}$.*

Proof. By Theorem 4.4 we have a scheme which satisfies the above efficiency conditions and supplies $t(n)$-way CPA-KDM-security for every polynomial $t(\cdot)$. As shown in [BPS07], we can use the standard encrypt-then-MAC transformation to upgrade the security to be $t(n)$-way CCA-KDM-security. By [IKOS08], the DRLC assumption (or more generally, the existence of any one-way function) implies the existence of a linear-time computable MAC scheme. Hence, this transformation adds a constant computational overhead. □

4.3 Security against Key-Related Attacks and Applications

Construction 4.1 is also secure against some key-related attacks. In such attack, the adversary is allowed to ask for encryptions under several different keys whose values are initially unknown, but where some mathematical relationship connecting the keys is known to the attacker.

Formally, we define such an attack with respect to a t-ary relation $\mathcal{R} \subseteq K^t$ where K is the key-space. (Again, $\mathcal{R}$ and K are both indexed by the security parameter.) Related-Key (RK) security is defined with respect to $\mathcal{R}$ via the following game that takes place between a challenger and an adversary A. For an integer $t > 0$ and a security parameter n the game proceeds as follows:

- **Initialization.** The challenger randomly chooses a bit $b \xleftarrow{R} \{0,1\}$ and t random secret keys secret keys $(k_1, \ldots, k_t)$ which satisfies $\mathcal{R}$. (I.e., the keys are sampled from the conditional distribution $[(G(1^n))^t | (G(1^n))^t \in \mathcal{R}]$.)
- **Queries.** The adversary asks for polynomially-many queries where each query is of the form (i, m) where $i \in [t]$ and m is in the message space. The challenger outputs

$$c \xleftarrow{R} \begin{cases} E(k_i, m) & \text{if } b = 0, \\ E(k_i, 0^{|m|}) & \text{if } b = 1. \end{cases}$$

- **Final phase.** The adversary attempt to guess b and outputs a bit $b' \in \{0,1\}$.

Definition 4.8. *(**RK-secure encryption**) Let $\mathcal{R}$ be a t-ary relation. A symmetric encryption scheme (G, E, D) is* RK-secure *with respect to $\mathcal{R}$ if no polynomial-time attacker A has non-negligible advantage in guessing the value of the bit b in the above game (where the running time and the advantage are measured as functions of the security parameter n).*

For a sequence of t matrices $Y = (Y_1, \ldots, Y_t) \in (\mathbb{F}_2^{n \times N})^t$ we let $\mathcal{R}_Y$ denote the linear relation $\left\{(X + Y_1, \ldots, X + Y_t) | X \in \mathbb{F}_2^{n \times N}\right\}$. By using the second property of Lemma 4.2 we can prove that our construction is RK-secure with respect to the class $\mathcal{R}_t$ of all linear relations.

Theorem 4.9. *Suppose that the DRLC assumption holds. Then, Construction 4.1 (instantiated with any arbitrary parameters $m(\cdot), N(\cdot), \ell(\cdot), \mu$) is RK-secure with respect to the relation $\mathcal{R}_Y$ for every polynomial $t = t(n)$ and $Y = (Y_1, \ldots, Y_t) \in (\mathbb{F}_2^{n \times N})^t$.*

sketch. Lemma 4.2 allows us to reduce an RK attack with a linear relation to a standard chosen-plaintext attack. By Theorem 4.4 our scheme is secure against such an attack. □

Application. Oblivious-Transfer (OT) is a cryptographic primitive which typically requires computationally expensive public-key operations. In [IKNP03] it was shown how to efficiently extend a small number of OTs to many OTs. The construction uses a random oracle which can be instantiated by any encryption scheme that provides RK-security with respect to linear functions.[6] Since our construction is highly efficient this instantiation does not add much overhead to the reduction.

[6]The authors of [IKNP03] originally refer to *correlation robust* function which seems stronger than encryption scheme with RK-security for linear functions. However, it is not hard to see that the latter primitive suffice for their application.

References

[ABW08] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. Submitted for publication., 2008.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in NC^0. In *Proc. 10th Random.*, 2006.

[AIK07] Benny Applebaum, Yuval Ishai, and Eyal 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.

[AIK08] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant latency. Submitted for publication., 2008.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *Proc. 44th FOCS*, pages 298–307, 2003.

[BDU08] Michael Backes, Markus Dürmuth, and Dominique Unruh. Oaep is secure under keydependent messages. In *ASIACRYPT '08*, 2008. To appear.

[BFKL94] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology: Proc. of CRYPTO '93*, volume 773 of *LNCS*, pages 278–291, 1994.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *CRYPTO '08*, pages 108–125, 2008.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003. Preliminary version in Proc. 32nd STOC, 2000.

[BMvT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[BPS07] Michael Backes, Birgit Pfitzmann, and Andre Scedrov. Key-dependent message security under active attacks - brsim/uc-soundness of symbolic encryption with key cycles. In *Proceedings of 20th IEEE Computer Security Foundation Symposium (CSF)*, June 2007. Preprint on IACR ePrint 2005/421.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *SAC '02*, pages 62–75, 2002.

[CCS08] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. Cryptology ePrint Archive, Report 2008/375, 2008.

[CL01] Camenisch and Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT '01*, 2001.

[Cop82] Don Coppersmith. Rapid multiplication of rectangular matrices. *SICOMP: SIAM Journal on Computing*, 11, 1982.

[DKL] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. Cryptography with auxiliary inputs. Submitted for publication.

[DN02] Ivan B. Damgård and Jesper Buus Nielsen. An efficient pseudo-random generator with applications to public-key encryption and constant-round multiparty computation. Unpublished, 2002.

[DRV02] Nenad Dedic, Leonid Reyzin, and Salil P. Vadhan. An improved pseudorandom generator based on hardness of factoring. In *Proc. 3rd SCN*, pages 88–101, 2002.

[EGL85] Even, Goldreich, and Lempel. A randomized protocol for signing contracts. *CACM: Communications of the ACM*, 28, 1985.

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *Proc. 47th FOCS*, pages 563–574, 2006.

[FIPR05] Freedman, Ishai, Pinkas, and Reingold. Keyword search and oblivious pseudorandom functions. In *Theory of Cryptography Conference (TCC), LNCS*, volume 2, 2005.

[FS96] Jean-Bernard Fischer and Jacques Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *Proc. EuroCrypt '96*, volume 1070, pages 245–255, 1996.

[Gen05] Rosario Gennaro. An improved pseudo-random generator based on the discrete logarithm problem. *J. Cryptology*, 18(2):91–110, 2005.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. of the ACM*, 33:792–807, 1986.

[GKL93] Oded Goldreich, Hugo Krawczyk, and Michael Luby. On the existence of pseudorandom generators. *SIAM J. Comput.*, 22(6):1163–1175, 1993. Preliminary version in Proc. 29th FOCS, 1988.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[HB01] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *Advances in Cryptology: Proc. of ASIACRYPT '01*, volume 2248 of *LNCS*, pages 52–66, 2001.

[HH08] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. Cryptology ePrint Archive, Report 2008/164, 2008.

[HK07] Shai Halevi and Hugo Krawczyk. Security under key-dependent inputs. In *CCS '07*, pages 466–475, 2007.

[HL08] Carmit Hazay and Yehuda Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In *TCC '08*, pages 155–175, 2008.

[HU08] Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In *EUROCRYPT '08*, pages 108–126, 2008.

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *Advances in Cryptology: Proc. of CRYPTO '03*, volume 2729, pages 145–161, 2003.

[IKOS08] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *Proc. 40th STOC*, 2008.

[JW05] A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology: Proc. of CRYPTO '05*, volume 3621 of *LNCS*, pages 293–308, 2005.

[Kea98] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. of the ACM*, 45(6):983–1006, 1998.

[KS06] J. Katz and J.-S. Shin. Parallel and concurrent security of the hb and hb+ protocols. In *Advances in Cryptology: Proc. of Eurocrypt 06'*, volume 4004 of *LNCS*, pages 73–87, 2006.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *Proc. 9th Random*, 2005.

[NR04] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004. Preliminary version in Proc. 38th FOCS, 1997.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. 37th STOC*, pages 84–93, 2005.

[Spi95] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. In *Proc. 27th STOC*, pages 388–397, 1995.