Cryptography from Learning Parity with Noise[*]

Krzysztof Pietrzak

Institute of Science and Technology (IST) Austria

Abstract. The Learning Parity with Noise (LPN) problem has recently found many applications in cryptography as the hardness assumption underlying the constructions of "provably secure" cryptographic schemes like encryption or authentication protocols. Being provably secure means that the scheme comes with a proof showing that the existence of an efficient adversary against the scheme implies that the underlying hardness assumption is wrong.

LPN based schemes are appealing for theoretical and practical reasons. On the theoretical side, LPN based schemes offer a very strong security guarantee. The LPN problem is equivalent to the problem of decoding random linear codes, a problem that has been extensively studied in the last half century. The fastest known algorithms run in exponential time and unlike most number-theoretic problems used in cryptography, the LPN problem does not succumb to known quantum algorithms. On the practical side, LPN based schemes are often extremely simple and efficient in terms of code-size as well as time and space requirements. This makes them prime candidates for light-weight devices like RFID tags, which are too weak to implement standard cryptographic primitives like the AES block-cipher.

This talk will be a gentle introduction to provable security using simple LPN based schemes as examples. Starting from pseudorandom generators and symmetric key encryption, over secret-key authentication protocols, and, if time admits, touching on recent constructions of public-key identification, commitments and zero-knowledge proofs.

1 Learning Parity with Noise and Related Problems

The *search* version of the learning parity with noise problem with parameters $\ell \in \mathbb{N}$ (the length of the secret), $\tau \in \mathbb{R}$ where $0 < \tau < 0.5$ (the noise rate) and $q \in \mathbb{N}$ (the numbers of samples) asks to find a fixed random ℓ bit secret $\mathbf{s} \in \mathbb{Z}_2^\ell$ from q samples of the form $\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle \oplus e$ where $\mathbf{a} \in \mathbb{Z}_2^\ell$ is random and $e \in \mathbb{Z}_2$ has Bernoulli distribution with parameter τ (we denote this distribution with Ber_τ), i.e. $\Pr[e = 1] = \tau$. The *decisional* LPN problem is defined similarly, except that we require that one cannot even distinguish noisy inner products from random. The distinction between the search and the decisional version of a problem is often made for problems used in cryptography. Typically, assuming the decisional version of a problem allows for much simpler and more efficient constructions

[*] This survey paper accompanies an invited talk at SOFSEM 2012.

M. Bieliková et al. (Eds.): SOFSEM 2012, LNCS 7147, pp. 99–114, 2012.

of cryptosystems, whereas the search version is a weaker assumption and thus constructions based on it require less "faith" in the presumed hardness of the assumption.[1] Interestingly, for the LPN problem one can show that the distinction between the search and the decisional version is irrelevant, more on this below. Before we formally define the LPN problem, let us set the notational conventions for the rest of this paper.

Notation. $\mathbb{Z}_q$ denotes the set $\{0,1,\dots,q-1\}$, and addition is always modulo q. In particular, $\mathbb{Z}_2=\{0,1\}$ are bits and $\oplus$ denotes bitwise XOR. We use bold small and capital letters like $\mathbf{x},\mathbf{X}$ to denote vectors and matrices, respectively. Calligraphic letters like $\mathcal{X}$ denote sets. For a set $\mathcal{X}$, $x \overset{\$}{\leftarrow} \mathcal{X}$ denotes that x is assigned a value sampled uniformly at random from $\mathcal{X}$. For a distribution D, $x \leftarrow D$ denotes that x is sampled according to D. With Ber_τ we denote the Bernoulli distribution with parameter τ, i.e. $\Pr[x=1\ ;\ x\leftarrow \mathsf{Ber}_\tau]=\tau$. For $m\in\mathbb{N}$, U_m denotes the uniform distribution over $\mathbb{Z}_2^m$. $X\sim D$ denotes that X is a random variable with distribution D. $\langle \mathbf{a},\mathbf{b}\rangle=\sum_{i=1}^n \mathbf{a}[i]\cdot\mathbf{b}[i] \bmod p$ denotes the inner product of $\mathbf{a},\mathbf{b}\in\mathbb{Z}_p^n$.

The Basic LPN Problem

Definition 1 (search/decisional LPN Problem). *For $\tau\in]0,1/2[$, $\ell\in\mathbb{N}$, the* decisional $\mathsf{LPN}_{\tau,\ell}$ *problem is (q,t,ϵ)-hard if for every distinguisher* D *running in time t*

$$\left|\Pr_{\mathbf{s},\mathbf{A},\mathbf{e}}[\mathsf{D}(\mathbf{A},\mathbf{A}.\mathbf{s}\oplus\mathbf{e})=1]-\Pr_{\mathbf{r},\mathbf{A}}[\mathsf{D}(\mathbf{A},\mathbf{r})=1]\right|\le\epsilon \tag{1}$$

Where $\mathbf{s}\overset{\$}{\leftarrow}\mathbb{Z}_2^\ell$, $\mathbf{A}\overset{\$}{\leftarrow}\mathbb{Z}_2^{q\times\ell}$, $\mathbf{e}\leftarrow\mathsf{Ber}_\tau^q$ and $\mathbf{r}\overset{\$}{\leftarrow}\mathbb{Z}_2^q$. The search $\mathsf{LPN}_{\tau,\ell}$ *problem is (q,t,ϵ)-hard if for every* D *running in time t*

$$\Pr_{\mathbf{s},\mathbf{A},\mathbf{e}}[\mathsf{D}(\mathbf{A},\mathbf{A}.\mathbf{s}\oplus\mathbf{e})=\mathbf{s}]\le\epsilon \tag{2}$$

The Learning with Errors (LWE) Problem. A problem closely related to LPN is the learning with errors (LWE) problem introduced by Regev [43]. LWE is a generalization of LPN to larger moduli. For some prime p,[2] we have a secret $\mathbf{s}\in\mathbb{Z}_p^\ell$, and the adversary is asked to find $\mathbf{s}$ given samples $\langle\mathbf{a},\mathbf{s}\rangle+e$. Here $\mathbf{a}$ is uniform in $\mathbb{Z}_p^\ell$ and the noise $e\in\mathbb{Z}_p$ is sampled according to some distribution χ, typically this distribution is a "discrete Gaussian". A good survey paper on LWE and its applications is [44].[3] LWE seems much more versatile than LPN. Besides all the cryptographic primitives we can construct from LPN, there are

[1] A typical example is public-key encryption based on the Diffie-Hellman problem, which is quite straight forward and efficient using the decisional version of the problem [14], but much more tricky and less practical using the search version [11].

[2] The case where the moduli is a the power of a prime has also been used [2].

[3] A bibliography of LWE (and more generally, lattice) based cryptosystems is maintained on `http://xagawa.net/bib-lattice/`

contsructions of much more sophisticated objects like public-key encryption [43] (even fully homomorphic [16] or identity-based [21,10]) and collision resistant hash functions [33], which we do not know how to construct from LPN.[4] LWE is also interesting for theoretical reasons, as it has the remarkable property that its hardness follows form *worst case* hardness of lattice assumptions [43,40]. LWE lacks the simplicity of LPN,[5] and thus LWE based schemes are less suited for weak devices like RFID tags.

Decision vs. Search. In contrast to most cryptographic assumptions which come in a search and decisional variant, it turns out that for LPN the two versions are "polynomially equivalent" [5,31] as stated in the lemma below.[6] This means that any attacker of size t against decisional LPN implies an attacker of size $poly(t)$ against the search version. Thus, cryptosystems proven secure under the decisional LPN assumption are already secure if search LPN is hard. Although this search to decision reduction is not tight, in practice we have no faster algorithms for decision than for search.

Lemma 1 (Lemma 1 from [31]). *If decisional* $\mathsf{LPN}_{\tau,\ell}$ *is not* (q,t,ε) *secure, then search* $\mathsf{LPN}_{\tau,\ell}$ *is not* $O(q',t',\varepsilon')$ *secure where*

$$q' = O(q \cdot \log \ell / \varepsilon^2) \qquad t' = O(t \cdot \ell \cdot \log \ell / \varepsilon^2) \qquad \varepsilon' = \varepsilon/4$$

Relations of LPN to Other Problems. With the current state in complexity theory, we cannot expect to prove that there exists no efficient adversary who breaks the LPN problem, as this would imply $\mathcal{P} \neq \mathcal{NP}$. The search LPN problem can be stated as the $\mathcal{NP}$ complete problem of decoding random linear codes [7].[7] Think of $\mathbf{A}$ as the generator matrix and $\mathbf{s}$ as the message. The decoding problem

[4] Alekhnovich [4] and Applebaum, Barak and Wigderson [1] construct public-key encryption from variants of LPN (which seem like much stronger assumptions than LPN) where either the noise rate τ is not constant but depends on the length ℓ of the secret as $\tau = O(1/\sqrt{\ell})$ [4], or the vectors $\mathbf{a}$ are not uniform, but have Hamming weight exactly 3 [1]. Another approach, pioneered by McEliece [37], replaces the random $\mathbf{A}$ with a "disguised" generator matrix of a code which allows for efficient error-correction. The security of the public-key encryption scheme follows from LPN and the assumption that the disguised matrix is indistinguishable from uniformly random.

[5] It requires many multiplications modulo some prime p (Typically p is polynomial in a security parameter, and thus much smaller than the moduli used in discrete logarithm (or factoring based) based schemes, where $\log(p)$ must be at least as as large as the security parameter), as opposed to inner products of bit-vectors as for LPN.

[6] Such an equivalence also holds for LWE with prime modulus [43] or if the modulus is the power of a prime ([2], Lemma 1).

[7] This does not imply that LPN is hard assuming $\mathcal{P} \neq \mathcal{NP}$ as search LPN is an average case problem (we require that no efficient adversary succeeds with non-negligible probability), whereas $\mathcal{NP}$ hardness is just a worst case guarantee (no efficient adversary succeeds on all inputs), see [5] for a more in-depth discussion.

then asks to recover the message $\mathbf{s}$ from the noisy codeword $\mathbf{A}.\mathbf{s} \oplus \mathbf{e}$, which is exactly search LPN. The LPN problem has been extensively studied in learning theory, as an efficient algorithm for LPN would allow to learn several important concept classes like 2-DNF formulas, juntas, and any function with a sparse Fourier spectrum [15].

The best known algorithms to recover an ℓ bit secreet need $2^{\Theta(\ell/\log \ell)}$ time and samples [6,32]. If given only polynomially many $q = \mathsf{poly}(\ell)$ samples, the running time of the best algorithm goes up to $2^{\Theta(\ell/\log\log \ell)}$ [36], and given only linearly many samples $q = \Theta(\ell)$, the best algorithms run in exponential $2^{\Theta(\ell)}$ time [47,38]. Unlike most number-theoretic problems used in cryptography, no quantum algorithms for LPN are known which are significantly faster than the classical ones. See [32] for more exact estimates and suggestions of parameters ℓ, τ for cryptographic applications. In the next paragraphs we discuss the hardness of LPN when either the secret $\mathbf{s} \sim U_\ell$, the randomness $\mathbf{A} \sim U_{q\times\ell}$ or the error $\mathbf{e} \sim \mathsf{Ber}_\tau^q$ does not have the right distribution.

Hardness of LPN for Non-Uniform Secrets. The secret $\mathbf{s} \in \mathbb{Z}_2^\ell$ in the LPN problem is usually assumed to be uniformly random. It is not hard to see that this is the hardest distribution, in the sense that given an adversary D who finds a uniform $\mathbf{s}$ given $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e})$ with some probability δ, we can recover a $\hat{\mathbf{s}}$ with any distribution over $\mathbb{Z}_2^\ell$ from $(\mathbf{A}, \mathbf{A}.\hat{\mathbf{s}} \oplus \mathbf{e})$ with the same probability δ as follows: given $(\mathbf{A}, \mathbf{A}.\hat{\mathbf{s}} \oplus \mathbf{e})$, sample a uniform $\mathbf{s}'$ and invoke D on $(\mathbf{A}, \mathbf{A}.\hat{\mathbf{s}} \oplus \mathbf{e} \oplus \mathbf{A}.\mathbf{s}') = (\mathbf{A}, \mathbf{A}.(\hat{\mathbf{s}} \oplus \mathbf{s}') \oplus \mathbf{e})$. Note that $\hat{\mathbf{s}} \oplus \mathbf{s}'$ is uniform as required, and if D finds $\hat{\mathbf{s}} \oplus \mathbf{s}'$ (which happens with probability δ) we can recover $\hat{\mathbf{s}} = (\hat{\mathbf{s}} \oplus \mathbf{s}') \oplus \mathbf{s}'$.

Surprisingly, the uniform distribution is not the unique hardest distribution. Applebaum et al. ([2] , Lemma 2) show that the LWE problem (where the modulus p is prime or a power of a prime, and the noise distribution χ is arbitrary) is basically as hard if the secret $\mathbf{s} \leftarrow \chi^\ell$ is chosen according to the noise distribution and not uniform $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^\ell$. In particular, the LPN problem where $\mathbf{s} \leftarrow \mathsf{Ber}_\tau^\ell$ is as hard as for uniform $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^\ell$. In [2] this result is used to construct a key-dependent message secure public-key encryption scheme from LWE, and key-dependent secret-key encryption from LPN, we'll revisit the latter result later.

Motivated by applications to leakage-resilient cryptography, Goldwasser et al. [18] investigate the hardness of the LWE problem when the secret $\mathbf{s} \in \mathbb{Z}_p^\ell$ is not uniform, but is only known to have some min-entropy.[8] The best one can hope for is that LWE with secrets of min-entropy k is as hard as standard LWE with secrets of length $\ell' = k/\log(p)$.[9] In [18] it is shown that this is almost the case for LWE where the noise has a Gaussian distribution (which is the most

[8] X has min entropy k if $\Pr[X = x] \leq 2^{-k}$ for every x in the support of X.

[9] The reason is that the particular distribution where the first ℓ' elements of $\mathbf{s}$ are uniform and the remaining ones are all zero has min-entropy $k = \ell' \log(p)$. LWE with such a secret is easily seen to be equivalent to LWE with a uniform secret in $\mathbb{Z}_p^{\ell'}$.

interesting case due to its equivalence to worst-case lattice problems.) They prove that if the standard LWE problem with uniform secrets over $\mathbb{Z}_p^{\ell'}$ (and noise distribution Gaussian with standard deviation α) is hard, then the (non-standard) LWE with secrets in $\mathbb{Z}_p^{\ell}$ having min-entropy $k = \ell' \log(p) + \omega(\log \ell)$ (and noise distribution Gaussian with standard deviation β) must also be hard. The reduction from [18] is not tight, but the main caveat is that it also blows up the noise distribution: the fraction of the deviations α/β must be negligible. For this reason, their result requires the modulus p to be at least superpolynomial, and in particular it implies nothing for the LPN problem where $p = 2$.

Nothing is known about the hardness of LPN for general distributions of high min-entropy. In the interesting special case where the secret $\mathbf{s} \in \mathbb{Z}_2^{\ell}$ is uniformly sampled from any $\ell' \leq \ell$ dimensional linear subspace (and thus has min-entropy ℓ'), the problem can be show to be exactly as hard as the standard LPN problem with a uniform ℓ' bit secret. This follows from the equivalence of LPN and the subspace LPN problem that we'll discuss below.

Hardness of LPN for Non-Uniform Noise. The LPN problem seems to remain hard, even if $\mathbf{s}$ and/or the rows of $\mathbf{A}$ are not uniform, but have sufficiently high min-entropy (if they are sampled from a linear subspace, this can even be proven.) Fiddling with the distribution of the error vector $\mathbf{e}$ is more delicate. If e.g. ℓ positions of $\mathbf{e}$ are fixed (or otherwise known to the adversary), she learns ℓ noiseless linear equations $\langle \mathbf{a}, \mathbf{s} \rangle = y$, and can compute $\mathbf{s}$ from this linear equations using normal Gaussian elimination. Arora and Ge [3] show an attack for the case where the bits of $\mathbf{e}$ are not i.i.d., but sampled as follows. For some n, the noise vector is sampled n bits at a time, where each such block is sampled independently at random, conditioned on having a 1 in exactly (or at most) $w = n.\tau$ positions. Although here each individual noise bit has distribution Ber_τ as required, the noise bits are not independent any more. Using a technique called linearization, [3] show that with this noise distribution one can recover the secret $\mathbf{s}$ in time roughly n^w.

Saving Public Randomness. The most expensive part in generating an LPN instance $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e})$ is the sampling of the random matrix $\mathbf{A} \in \mathbb{Z}_2^{q \times \ell}$. For some of the cryptosystems we'll discuss in Section 2 (namely for pseudorandom generators and commitment schemes) the fact that $\mathbf{A}$ is rather large and must be uniform will not be much of a problem: as $\mathbf{A}$ can be *public*, we can fix a public random $\mathbf{A}$ once and for all, and then use it to generate arbitrary many LPN instances. For other schemes (namely encryption and secret-key identification), the size of $\mathbf{A}$ is more of an issue, as here the secret $\mathbf{s}$ will play the role of a shared secret key, and thus is fixed. In order to generate new LPN instances (which is required to compute ciphertexts and during execution of the identification protocol), one must sample a fresh $\mathbf{A}$'s every time.

One way to leverage this problem is to use $n > 1$ independent random secrets, this will typically increase the size of the secret key by a factor of n, but decrease the cost due to the sampling, storing and/or sending $\mathbf{A}$ by the same factor as we can reuse each $\mathbf{A}$ n times.

It has also been suggested to not sample $\mathbf{A} \in \mathbb{Z}_2^{q \times \ell}$ uniformly at random, but from a distribution which allows a more succinct description of the samples. For example [23] suggest to use a random Toeplitz matrix, which requires only $q + \ell$ (as opposed $q \cdot \ell$) random bits. Such a matrix is sampled by choosing a random $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_2^{q+\ell}$, then the i'th row of $\mathbf{A}$ is $\mathbf{a}[i \ldots i + \ell]$.

Another variant, called Ring-LPN, has been suggested in [27]. Ring-LPN not only has a succinct description of $\mathbf{A}$, but also allows for extremely efficient evaluation of the matrix multiplication $\mathbf{A}.\mathbf{s}$ as it corresponds to a single multiplication of two polynomials. Ring-LPN is inspired by the Ring-LWE problem, strong evidence for the hardness of Ring-LWE is given in [34] who show it to be equivalent to hard problems on *ideal* lattices.

Subspace LPN. The subspace LPN problem [41] is a variant of the LPN problem where the adversary not only gets random samples $\langle \mathbf{a}, \mathbf{s} \rangle \oplus e$, but it is an interactive assumption where she can adaptively choose affine functions ϕ_a, ϕ_s and then gets samples $\langle \phi_a(\mathbf{a}), \phi_s(\mathbf{s}) \rangle \oplus e$. That is, $\mathbf{a}$ and $\mathbf{s}$ are first mapped to the linear subspaces defined by ϕ_a and ϕ_s, respectively, before the noisy inner product is computed.[10] If the adversary is restricted to choose mappings ϕ_a, ϕ_s that overlap in at least an ℓ' dimensional subspace,[11] then this problem is at most as hard as the LPN problem with secrets of length ℓ' (as one can map to a string which is all zero except for the first ℓ' bits.) In [41] it is shown that the other direction does also (almost) hold (this equivalence not only holds for LPN, but more generally for LWE using any prime modulus and any error distribution.) This equivalence has immediate consequences for several existing LPN and LWE based cryptosystems, as it implies much stronger security guarantees as anticipated by the designers of the schemes. For example security against related key attacks or security against "weak" randomness, cf. [41] for the details. The fact that subspace LPN is an interactive assumption, gives a powerful handle for constructing provably secure LPN based cryptosystems. In Section 2 we'll mention constructions of identification schemes and message authentication codes [30] whose proof heavily relies on this handle.

Exact LPN. Recall that the error vector $\mathbf{e} \in \mathbb{Z}_2^q$ of an $\mathsf{LPN}_{\tau,\ell}$ sample $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e})$ consists of q i.i.d. bits with distribution Ber_τ, and thus its expected Hamming weight (i.e. number of 1's) is $q\tau$. The Exact LPN (XLPN for short) problem is a minor variation of LPN where we require $\mathbf{e}$ to have Hamming weight *exactly* $\lceil q\tau \rfloor$. In the next section we'll mention some cryptosystems which rely on the search XLPN (a public-key identification scheme) and the decisional XLPN (efficient zero-knowledge proofs for linear functions of committed values.)

[10] The affine function $\phi_a : \mathbb{Z}_2^\ell \to \mathbb{Z}_2^\ell$ can be defined as $\mathbf{a} \to \mathbf{X}_a.\mathbf{a} \oplus \mathbf{x}_a$ for some matrix $\mathbf{X}_a \in \mathbb{Z}_2^{\ell \times \ell}$ (the linear part) and some vector $\mathbf{x}_a \in \mathbb{Z}_2^\ell$ (the affine part). Equivalently, ϕ_s can be defined by $\mathbf{X}_s, \mathbf{x}_s$.

[11] This means that $\mathbf{X}_a^T.\mathbf{X}_s$ has rank at least ℓ'.

Hardness of *search* XLPN trivially follows from standard LPN.[12] Unfortunately the proof of equivalence of search and decision for LPN does not work for the XLPN problem, and it is open if *decisional* XLPN is equivalent to LPN.[13]

A similar version of LPN where the Hamming weight is *at most* $\lceil q\tau \rfloor$ has been suggested by [31] as a means to get more efficient instantiations of LPN based cryptosystems.

2 Efficient LPN Based Cryptosystems

OWFs and Generic Constructions. A function $\mathsf{F} : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$ is a one-way function (OWF), if it's hard to find a preimage for F for a random outputs. In asymptotic terms, we require that F runs in time $\mathsf{poly}(n)$, and for all D of size $\mathsf{poly}(n)$ we have

$$\Pr_{\mathbf{x} \xleftarrow{\$} \mathbb{Z}_2^n} [\mathsf{D}(\mathsf{F}(\mathbf{x})) = \mathbf{x}' \text{ where } \mathsf{F}(\mathbf{x}) = \mathsf{F}(\mathbf{x}')] = \mathsf{negl}(n)$$

One can construct a OWF from LPN.[14] From a OWF one can construct pseudorandom objects using generic constructions, like pseudorandom generators (PRG) [26], pseudorandom functions (PRF) [17] or pseudorandom permutations (PRP) [35]. The basic secret-key cryptographic tasks[15] like encryption and authentication are usually solved by using a PRP like the AES block-cipher. Constructions using this generic transformations would typically be too inefficient to compete with dedicated designs,[16] as a consequence, today basically all

[12] An $\mathsf{LPN}_{\tau,q}$ sample $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e})$ will satisfy $\|\mathbf{e}\|_1 = \lceil q\tau \rfloor$ with probability $\approx 1/\sqrt{q}$. This implies that an adversary who breaks the search XLPN problem (i.e. outputs the right $\mathbf{s}$) with probability δ, is also an adversary against the search LPN problem with advantage at least $\delta/\sqrt{q}$. To see this, note that conditioned on the LPN sample satisfying $\|\mathbf{e}\|_1 = \lceil q\tau \rfloor$, we have exactly the same distribution as in XLPN, and thus in this case the adversary will be successful with probability δ.

[13] It is important that in the search XLPN problem, the adversary only gets one sample $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e})$. If she can ask for polynomially many samples, then the search to decision reduction does work, but now it's open if this "many sample" version of search XLPN is equivalent to standard LPN.

[14] Fix some random $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{q \times \ell}$. Now $\mathsf{F}(\mathbf{x})$ on input a (sufficiently long) string $\mathbf{x}$, uses $\mathbf{x}$ to sample $\mathbf{s} \sim U_\ell, \mathbf{e} \sim \mathsf{Ber}_\tau^q$. If the weight of $\mathbf{e}$ is unexpectedly high (say $\|\mathbf{e}\|_1 \geq q \cdot \frac{1/2+\tau}{2}$) we output a special symbol $\perp$ (using the Chernoff bound one can show that this happens with exponentially small probability.) Otherwise output $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e})$. As $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e})$ uniquely determines $\mathbf{s}$, any algorithm who finds a preimage must find (randomness used to sample) this unique $\mathbf{s}$, which would contradict the hardness of the search LPN problem.

[15] In the secret-key setting, the honest parties share a secret key not known to the adversary.

[16] We use the term "dedicated design" as opposed to constructions that come with a reductionist proof showing the construction is secure under some standard hardness assumption.

secret-key cryptography is done using a dedicated construction like AES as main ingredient.[17]

The fascinating thing about the LPN problem is that it gives rise to secret-key cryptosystems which not only are provably secure, but in some aspects can even outperform known dedicated constructions. Below we show how the search to decision equivalence for the LPN problem, discussed in the previous section, implies that plain LPN samples are already pseudorandom, and thus give an extremely simple and efficient PRG.

It is an open problem to get a PRF or even PRP from LPN which could compete with dedicated constructions. But simple and efficient schemes for encryption, identification and authentification *not going via a PRF or PRP construction*, do exist. We'll discuss some of them below.[18]

Pseudorandom Generators. A pseudorandom generator is a length increasing function $\mathsf{G} : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ where $\mathsf{G}(U_n)$ is pseudorandom, this means that no efficient distinguisher can distinguish $\mathsf{G}(U_n)$ from the uniform distribution U_m.

The definition of the decisional LPN problem (cf. eq.(1)) already implies that $\mathsf{LPN}_{\tau,\ell}$ samples $\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e}$ are pseudorandom, which gives rise to a simple construction of a PRG [5]: use the (uniformly random) input $\mathbf{r}$ to sample $\mathbf{A}, \mathbf{s}, \mathbf{e}$ and output $\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e}$. As observed by [2], we can fix $\mathbf{A}$ as a public parameter and need not to sample it, or include it in the output.

To show that this function $\mathbf{r} \to \mathbf{A}.\mathbf{s} \oplus \mathbf{e} \in \mathbb{Z}_2^q$ is a PRG, one also must show how to sample $\mathbf{s} \sim U_\ell$ and $\mathbf{e} \sim \mathsf{Ber}_\tau^q$ using a random seed $\mathbf{r}$ which is shorter than the q bit output (recall that a PRG must be length increasing.) This is possible for any $\tau < 0.5, \ell \in \mathbb{Z}$ and sufficiently large q: we need ℓ bits to sample the uniform $\mathbf{s} \sim U_\ell$. But each bit of $\mathbf{e}$ has only $h(\tau)$ bits of entropy (where $h(\tau) = -\tau \log \tau - (1-\tau)\log(1-\tau)$ denotes the binary entropy function.) Thus $(\mathbf{s}, \mathbf{e})$ has $\ell + qh(\tau)$ bits of entropy and can be sampled using roughly that many bits, and this can be done very efficiently (see [2] for details.)

For sufficiently large q we have $\ell + qh(\tau) < q$, thus the stretch (which denotes the number of bits the output is longer than the input) of this PRG is $(1 - h(\tau))q - \ell$. This is linear in the length $\approx \ell + qh(\tau)$ of the seed. Linear stretch is a desirable property of PRGs as the efficiency of constructions which use a PRG crucially depend on its stretch.

[2] suggest a variant of this construction where one uses several ℓ-bit keys simultaneously, let $\mathbf{S} \in \mathbb{Z}_2^{\ell \times n}$ denote n such keys arranged as the columns of a

[17] This contrasts with public-key cryptography, like public-key encryption schemes, which are usually required to be provably secure, possibly using some idealized assumptions like the random oracle model [9]. This is due to the fact that public-key encryption needs much more structure than in the secret-key setting, where one just has to garble enough, and this can be done in any (invertible) way (the art in designing block-ciphers is to do this garbling extremely efficient.)

[18] [8] construct low-depth PRFs from the LWE problem by using a generic transformation from synthesizers to PRF [39]. A synthesizers is a strong type of a PRGs which, informally, is secure even if used on inputs that are somehow correlated. It is not known how to construct efficient synthesizers from LPN.

matrix. For the right choice of $n = \mathsf{poly}(\ell)$ one can use fast matrix multiplication [12] to compute the pseudorandom output $\mathbf{A}.\mathbf{S} \oplus \mathbf{E}$ (with $\mathbf{E} \leftarrow \mathsf{Ber}_\tau^{q \times n}$), which gives a PRG that can be evaluated in time $\tilde{O}(qn)$, which is quasilinear in the seed length. This is an asymptotic running time and it's not clear if this is already useful for input sizes used in practice.[19]

Secret-Key Encryption. A simple encryption scheme from LPN was proposed by [24]. The encryption of a message $\mathbf{m}$ under the secret key $\mathbf{s} \sim U_\ell$ is $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e} \oplus \mathbf{G}.\mathbf{m})$. Here $\mathbf{A} \sim U_{q \times \ell}$ and $\mathbf{e} \sim \mathsf{Ber}_\tau^q$, and $\mathbf{G} \in \mathbb{Z}_2^{q \times \ell}$ is the generator matrix of an error correcting code $\mathbb{Z}_2^\ell \to \mathbb{Z}_2^q$ which allows for efficient correction of $\tau' q$ errors (for some $\tau' > \tau$). To decrypt a ciphertext $(\mathbf{A}, \mathbf{y})$, one computes $\mathbf{G}.\mathbf{m} \oplus \mathbf{e} = \mathbf{y} \oplus \mathbf{A}.\mathbf{s}$. From this noisy codeword $\mathbf{G}.\mathbf{m} \oplus \mathbf{e}$ one can recover the message $\mathbf{m}$ using the error correcting decoding procedure for the code $\mathbf{G}$ if $\|\mathbf{e}\|_1 \leq \tau' q$, which will be the case withe exponentially high probability. The security[20] of this scheme follows from the fact that under the decisional LPN assumption, $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e})$ is pseudorandom, which implies that also the ciphertext $(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e} \oplus \mathbf{G}.\mathbf{m})$ is pseudorandom and thus hides the information of $\mathbf{m}$.

This scheme can not only be proven secure in the standard sense, but also provably satisfies some more exotic security notions. The equivalence of subspace LPN and LNP (discussed in the previous section) implies that this scheme is secure against so called *related key attacks* (RKA). More concretely, the adversary cannot only ask for encryptions under the key $\mathbf{s} \in \mathbb{Z}_2^\ell$, but also under keys $\phi(\mathbf{s})$ where $\phi(.)$ can be any adaptively chosen affine function (but the linear part must have sufficiently high rank $\ell' \leq \ell$, such that the LPN problem with secrets of length ℓ' is still hard.) In [2] it is shown that (a minor variant of) this scheme is secure under a large class of *key-dependent message attacks* (KDM). More concretely, the scheme remains secure even against adversaries who can ask for encryptions of any affine function (no restriction on the rank here) of the secret key.

Secret-Key Identification and Message Authentification. By far most research on LPN based cryptosystems has been published on secret-key identification protocols.[21] In such a protocol, a prover P exchanges messages with a verifier V.[22] P and V share a secret key. If V talks to the honest prover P, we require that finally V outputs accept. A typical application is access control, e.g. a wireless car key which has the role of the prover, and the car being the verifier.

There are several standard security definitions for identification protocols which try to capture the intuitive notion that an adversary not knowing the

[19] For the quasilinear running time, q and n must be in the order of ℓ^6, and thus the seed has size ℓ^{12}.

[20] More precisely, semantic security under chosen message attacks, which means no efficient adversary can distinguish encryptions any two different messages, even when given access to an encryption oracle.

[21] A list of relevant papers is on `http://www.ecrypt.eu.org/lightweight/index.php/HB`.

[22] In the context of RFID implementations, P is called the "tag" and V is the "reader".

secret key should not be able to make V accept. They differ in the power the adversary has before trying to launch such an impersonation attack.

In a *passive attack* the adversary can eavesdrop on several interactions between P and V, before trying to make V accept in a second phase. In an *active attack*, the adversary is additionally allowed to interact with P in the first phase. The strongest notion is a *man-in-the-middle attack* (MIM) where the adversary can arbitrarily interact with P and V (with polynomially many concurrent executions allowed) in the first phase.

Hopper and Blum [25] proposed the first LPN based identification scheme. Their goal was to design a scheme which is so simple that it could even be reliably executed by humans with just pen and paper. Their HB protocol is illustrated in Figure 1. The secret key is $\mathbf{s} \in \mathbb{Z}_2^\ell$ where ℓ is chosen such that the $\mathsf{LPN}_{\tau,\ell}$ problem is hard. The verifier sends as first message a challenge $\mathbf{A} \in \mathbb{Z}_2^{n\times\ell}$ (where n is a statistical security parameter), and the prover answers with an LPN sample $\mathbf{A}.\mathbf{s} \oplus \mathbf{e}$. The verifier accepts if the prover's answer $\mathbf{y}$ is of the form $\mathbf{y} = \mathbf{A}.\mathbf{s} \oplus \mathbf{e}$ for some low-weight $\mathbf{e}$. The expected weight of a correctly generated $\mathbf{e}$ is $n\tau$, the acceptance threshold of the verifier is set to $n\tau'$ for some $\tau < \tau' < 1/2$. This way the probability that a correctly generated $\mathbf{e} \leftarrow \mathsf{Ber}_\tau^n$ has weight $\geq n\tau'$ and thus V would reject (this is the completeness error) is exponentially small in n (this is shown using the Chernoff bound).

$$\begin{array}{lcl}
\underline{\mathsf{P}_{\tau,n}(\mathbf{s}\in\mathbb{Z}_2^\ell)} & & \underline{\mathsf{V}_{\tau',n}(\mathbf{s})} \\
 & \xleftarrow{\mathbf{A}} & \mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{n\times\ell} \\
\mathbf{e} \xleftarrow{\$} \mathsf{Ber}_\tau^n & & \\
\mathbf{z} := \mathbf{A}.\mathbf{s}\oplus\mathbf{e} & \xrightarrow{z} & \text{accept iff } \|\mathbf{z}\oplus\mathbf{A}.\mathbf{s}\|_1 < \tau'\cdot n
\end{array}$$

Fig. 1. The HB identification protocol [25], secure against passive attacks

The HB protocol can be proven secure against passive attacks assuming LPN is hard, but it can be easily broken with an active attack. Subsequently, Juels and Weis [29] proposed the HB^+ protocol, illustrated in Figure 2, with extends HB by one extra round. Their motivation was to find a protocol suitable for light weight devices like RFID tags, where an active attack is easy to launch. The HB^+ protocol is secure against active attacks, but not MIM attacks.[23] The HB^+ protocol has three rounds (not two like HB), which means the prover has to keep state in-between rounds, this is problematic for the devices like RFID tags.

The first two round protocol with active security was proposed in [30]. The design of the protocol is inspired by the subspace LPN problem, and diverges from the design of all previous LPN based protocols as now the randomness $\mathbf{A}$

[23] [29] only prove active security for a sequential version of the HB^+ protocol, where $\mathbf{A}$ is send row-by-row, and thus needs n rounds. Active security of the parallel protocol as in Figure 2 was later proven in [31].

$$\begin{array}{lcl}
\underline{\mathsf{P}_{\tau,n}(\mathbf{s}_1,\mathbf{s}_2\in\mathbb{Z}_2^{\ell})} & & \underline{\mathsf{V}_{\tau',n}(\mathbf{s}_1,\mathbf{s}_2)}\\
\mathbf{A}_1 \xleftarrow{\$} \mathbb{Z}_2^{n\times\ell} & \xrightarrow{\mathbf{A}_1} & \\
 & \xleftarrow{\mathbf{A}_2} & \mathbf{A}_2 \xleftarrow{\$} \mathbb{Z}_2^{n\times\ell}\\
\mathbf{e}\xleftarrow{\$}\mathsf{Ber}_{\tau}^{n} & & \\
\mathbf{z} := \mathbf{A}_1.\mathbf{s}_1\oplus\mathbf{A}_2.\mathbf{s}_2\oplus\mathbf{e} & \xrightarrow{z} & \text{accept iff}\\
 & & \|\mathbf{z}\oplus\mathbf{A}_1.\mathbf{s}_1\oplus\mathbf{A}_2.\mathbf{s}_2\|_1 \le \tau'\cdot n
\end{array}$$

Fig. 2. The HB^+ identification protocol [29,31], secure against active attacks

is chosen by the prover P, and is not as a challenge chosen by V. Instead, the challenge chosen by V is a vector $\mathbf{v}$ which specifies a subset $\mathbf{s}\wedge\mathbf{v}$ ($\wedge$ denotes bitwise AND) of the secret $\mathbf{s}$. The prover answers with a subspace LPN sample $\mathbf{A},\mathbf{A}.(\mathbf{s}\wedge\mathbf{v})\oplus\mathbf{e}$.

$$\begin{array}{lcl}
\underline{\text{Prover } \mathsf{P}_{\tau,n}(\mathbf{s}\in\mathbb{Z}_2^{2\ell})} & & \underline{\text{Verifier } \mathsf{V}_{\tau',n}(\mathbf{s}\in\mathbb{Z}_2^{2\ell})}\\
 & \xleftarrow{\mathbf{v}} & \mathbf{v}\xleftarrow{\$}\{\mathbf{x}\in\mathbb{Z}_2^{2\ell} : \|\mathbf{x}\|_1=\ell\}\\
\text{if } \|\mathbf{v}\|_1\neq\ell \text{ abort} & & \\
\mathbf{A}\xleftarrow{\$}\mathbb{Z}_2^{n\times 2\ell};\ \mathbf{e}\xleftarrow{\$}\mathsf{Ber}_{\tau}^{n} & & \\
\mathbf{z} := \mathbf{A}.(\mathbf{s}\wedge\mathbf{v})\oplus\mathbf{e}\in\mathbb{Z}_2^{n} & \xrightarrow{(\mathbf{A},\mathbf{z})} & \text{if } \mathsf{rank}(\mathbf{A})\neq n \text{ reject}\\
 & & \text{if } \|\mathbf{z}\oplus\mathbf{A}.(\mathbf{s}\wedge\mathbf{v})\|_1 > n\cdot\tau' \text{ reject, else accept}
\end{array}$$

Fig. 3. A two-round identification protocol with active security [30]

All the protocols discussed above can be easily broken by a MIM attack.[24] [30] shows how to transform the protocol from Figure 3 into a message authentication code (MAC). This also gives the first efficient MIM secure protocol from LPN as a MAC easily implies a two-round MIM secure protocol. A more efficient (and generic) transformation using pairwise independent hashing (instead of a permutation) appears in [13].

Public-Key Identification, Commitments and Zero-Knowledge

Public-Key Identification. In a public-key identification protocol the prover and verifier do not share a secret key. Instead, the prover knows a secret key *sk*, and a corresponding public key *pk* is known to everyone (i.e. verifiers and potential adversaries.) This setting is often favorable as it allows much simpler key-management. A verifier must make sure to learn *pk* authentically, but it must

[24] A MIM attack on HB^+ is given in [22]. As outlined in [30], a similar attacks exists for the protocol in Figure 3 (in [30] this attack is phrased as an attack on the protocol when used as a MAC, but exactly the same attacks works in the MIM setting.)

not remain secret. One can construct an identification scheme from any one-way function[25] $f(.)$ by setting $sk = x$ and $pk = f(x)$ for a random x. To prove its identity, the prover uses a zero-knowledge proof of knowledge (ZKPOK) [19] to convince the verifier that he knows the secret x, while not revealing any information beyond that. ZKPOKs exist for any one-way function [20], but this generic constructions are too computationally expensive to be used in practice.

For some particular functions, ZKPOKs exist which are much more efficient than the generic constructions, in particular, many number theoretical functions admit efficient proofs. For example Schnorr's protocol [45], which is a particularly simple 3-round ZKPOK (a so called Σ-protocol) to prove knowledge of the discrete logarithm x of some value g^x (where g is a generator of a prime order cyclic group.)

This elegant number theoretical constructions involve multiplications or even exponentiations over large moduli, and thus are still too computationally expensive for very weak devices like RFID tags. A few alternative protocols based on combinatorial (typically NP-complete) problems were suggested which avoid such expensive operations, including the Permuted Kernels Problem [46], the Permuted Perceptrons Problem [42] and Syndrome Decoding (of random linear codes) [48]. Stern's protocol [48] can be modified [28] (using the equivalence of LPN and decoding random linear codes) to get an efficient ZKPOK for the LPN problem. That is, given $(\mathbf{A}, \mathbf{y})$, one proves knowledge of $\mathbf{s}$ and a (low weight) $\mathbf{e}$ such that $\mathbf{y} = \mathbf{A}.\mathbf{s} \oplus \mathbf{e}$.[26]

String Commitments and Zero-Knowledge. A commitment scheme is the digital analogue of an envelope. The committing party can compute a commitment σ to an input $\mathbf{m}$, this commitment hides the committed message $\mathbf{m}$ (this is called the hiding property). Later the committing party can open σ and reveal $\mathbf{m}$, but he cannot open it to any other $\mathbf{m}' \neq \mathbf{m}$ (this is called the binding property.) The LPN problem allows for very simple perfectly binding[27] string commitments schemes [28]: the commitment to a bit-string $\mathbf{m}$ is $\mathbf{A}.(\mathbf{s}\|\mathbf{m}) \oplus \mathbf{e}$, i.e. it's an LPN sample using a secret whose first part $\mathbf{s}$ is random, and the second part is the message $\mathbf{m}$. $\mathbf{A}$ is a fixed random public matrix. To open a commitment one reveals $\mathbf{s}\|\mathbf{m}$ and the (low weight) $\mathbf{e}$.

[25] And more generally, any NP relation where one can efficiently sample instance/witness pairs and where it's hard to compute a witness for an instance sampled like that.

[26] In general the public-key setting is more demanding than the secret-key one (note that a public-key scheme trivially implies a secret-key one, just use (sk, pk) as the shared secret key.) This is also the case here, the LPN based public-key scheme [28] is at least an order of magnitude less efficient than, say the MIM secure scheme from [30]. Moreover the public-key scheme needs three rounds (as opposed to two for the secret-key setting), and even this is only true in the idealized "random oracle model" as it uses the Fiat-Shamir transformation to get from honest to full zero-knowledge. Without random oracles the round complexity is linear in the statistical security parameter n.

[27] Perfectly binding means that even a computationally unbounded adversary cannot open a commitment in two different ways.

The above-mentioned ZKPOK for LPN can be extended to not only prove knowledge of $\mathbf{s}$, but even to show that for any $\mathbf{X}, \mathbf{y}$ it holds that $\mathbf{X}.\mathbf{s} = \mathbf{y}$ [28]. For the commitment scheme as just described, this protocol can be used prove that, for any $\mathbf{X}, \mathbf{y}$, the value in a commitment satisfies $\mathbf{X}.\mathbf{m} = \mathbf{y}$. Here $\mathbf{X}.\mathbf{m}$ can e.g. be a subset of the bits of $\mathbf{m}$, which is something useful in cut-and-choose proofs. For this last application, the LPN assumption is not enough, but one has to rely on the decisional XPLN assumption discussed in the previous section.

3 Conclusions and Open Problems

A common dogma in the realm of secret-key cryptography is that provably secure schemes cannot be efficient enough to compete with dedicated constructions like the AES block-cipher. Recent constructions based on the hardness of LPN have at least challenged this viewpoint. Although we don't have an efficient block-cipher from LPN (and block-ciphers are used for almost all secret-key tasks), we have direct constructions for the most important tasks like encryption, identification and message authentication. It seems conceivable that for some settings (most notably for lightweight devices like RFIDs[28]) provable security is not just a nice theoretical feature, but actually can lead us to constructions which outperform known dedicated constructions in terms of efficiency vs. practical security (a viewpoint largely accepted in the realm of *pubic-key* cryptography.)

Security of LPN. We have discussed several variants of the LPN problem in Section 1. For some useful variants, like decisional XLPN or LPN where the secret only has high min-entropy, the relation to the standard LPN problem is open. Another intriguing open problem is the following, does hardness of LPN with noise rane τ imply anything for LPN with smaller noise rate $\tau' < \tau$? For all we know, there could be threshold such that LPN with noise rate τ is hard, but easy for any $\tau' < \tau$ (though, this seems not very likely.)

Constructions. What other primitives can we construct from LPN? Two basic primitives we don't know how to get from the (standard) LPN problem are public-key encryption and collision resistant hash functions. Can we construct these, or is there a fundamental reason why the more general LWE problem allows for such objects, but LPN does not?

[28] One disadvantage the LPN based schemes have to classical block-cipher based solutions is that they require more randomness than block-cipher based schemes (e.g. with a block-cipher, we get identification where only the verifier, but not the prover needs to sample random bits. And message authentication codes can be completely deterministic, whereas the LPN based MAC is probabilistic.) This randomness needs make LPN based schemes poor candidates for powerful processors (as e.g. used in laptops, or even smart-phones) where generating randomness is expensive compared to clock-cycles or code-size. On weak devices like RFIDs or smart cards, this is not (or less) the case. Moreover even a priori deterministic computations are randomized on such devices by "masking" or "blinding" in order to protect the computation from side-channel attacks. Potentially, the randomization used for LPN based schemes provides the same effect as masking or blinding for free.

References

1. Applebaum, B., Barak, B., Wigderson, A.: Public-key cryptography from different assumptions. In: Schulman, L.J. (ed.) 42nd ACM STOC, pp. 171–180. ACM Press (2010)
2. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)
3. Arora, S., Ge, R.: New Algorithms for Learning in Presence of Errors. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 403–415. Springer, Heidelberg (2011)
4. Alekhnovich, M.: More on average case vs approximation complexity. In: 44th FOCS, pp. 298–307. IEEE Computer Society Press (2003)
5. Blum, A., Furst, M.L., Kearns, M.J., Lipton, R.J.: Cryptographic Primitives Based on Hard Learning Problems. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 278–291. Springer, Heidelberg (1994)
6. Blum, A., Adam Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM 50(4), 506–519 (2003)
7. Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.A.: On the inherent intractability of certain coding problems. IEEE Trans. Information Theory IT-24(3), 384–386 (1978)
8. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. Cryptology ePrint Archive, Report 2011/401 (2011), http://eprint.iacr.org/
9. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press (1993)
10. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
11. Cash, D., Kiltz, E., Shoup, V.: The Twin Diffie-Hellman Problem and Applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)
12. Coppersmith, D.: Rapid multiplication of rectangular matrices. SIAM J. Comput. 11(3), 467–471 (1982)
13. Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited (manuscript, 2011)
14. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31, 469–472 (1985)
15. Feldman, V., Gopalan, P., Khot, S., Ponnuswami, A.K.: New results for learning noisy parities and halfspaces. In: 47th FOCS, pp. 563–574. IEEE Computer Society Press (2006)
16. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC, pp. 169–178. ACM Press (2009)
17. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. Journal of the ACM 33, 792–807 (1986)
18. Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: ICS, pp. 230–240 (2010)
19. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on Computing 18(1), 186–208 (1989)

20. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM 38(3), 691–729 (1991)
21. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC, pp. 197–206. ACM Press (2008)
22. Gilbert, H., Robshaw, M., Sibert, H.: An active attack against hb+ - a provably secure lightweight authentication protocol. Cryptology ePrint Archive, Report 2005/237 (2005), `http://eprint.iacr.org/`
23. Gilbert, H., Robshaw, M.J.B., Seurin, Y.: HB$^\sharp$: Increasing the Security and Efficiency of HB$^+$. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 361–378. Springer, Heidelberg (2008)
24. Gilbert, H., Robshaw, M.J.B., Seurin, Y.: How to Encrypt with the LPN Problem. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 679–690. Springer, Heidelberg (2008)
25. Hopper, N.J., Blum, M.: Secure Human Identification Protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001)
26. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999)
27. Heyse, S., Kiltz, E., Lyubashevsky, V., Paar, C., Pietrzak, K.: An efficient authentication protocol based on ring-lpn (manuscript, 2011)
28. Jain, A., Pietrzak, K., Tentes, A.: Commitments and efficient zero-knowledge from hard learning problems (manuscript, 2011)
29. Juels, A., Weis, S.A.: Authenticating Pervasive Devices with Human Protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
30. Kiltz, E., Pietrzak, K., Cash, D., Jain, A., Venturi, D.: Efficient Authentication from Hard Learning Problems. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 7–26. Springer, Heidelberg (2011)
31. Katz, J., Shin, J.S., Smith, A.: Parallel and concurrent security of the HB and HB+ protocols. Journal of Cryptology 23(3), 402–421 (2010)
32. Levieil, É., Fouque, P.-A.: An Improved LPN Algorithm. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 348–359. Springer, Heidelberg (2006)
33. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFT: A Modest Proposal for FFT Hashing. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 54–72. Springer, Heidelberg (2008)
34. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors Over Rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)
35. Luby, M., Rackoff, C.: How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal on Computing 17(2) (1988)
36. Lyubashevsky, V.: The Parity Problem in the Presence of Noise, Decoding Random Linear Codes, and the Subset Sum Problem. In: Chekuri, C., Jansen, K., Rolim, J.D.P., Trevisan, L. (eds.) APPROX 2005 and RANDOM 2005. LNCS, vol. 3624, pp. 378–389. Springer, Heidelberg (2005)
37. McEliece, R.J.: A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report 44, 114–116 (1978)
38. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $o(2^{0.054n})$. In: ASIACRYPT (2011)

39. Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of pseudo-random functions. J. Comput. Syst. Sci. 58(2), 336–375 (1999)
40. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher, M. (ed.) 41st ACM STOC, pp. 333–342. ACM Press (2009)
41. Pietrzak, K.: Subspace LWE (2010) (manuscript)
42. Pointcheval, D., Poupard, G.: A new np-complete problem and public-key identification. Des. Codes Cryptography 28(1), 5–31 (2003)
43. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 84–93. ACM Press (2005)
44. Regev, O.: The learning with errors problem (invited survey). In: IEEE Conference on Computational Complexity, pp. 191–204 (2010)
45. Schnorr, C.-P.: Efficient Identification and Signatures for Smart Cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (1990)
46. Shamir, A.: An Efficient Identification Scheme based on Permuted Kernels (Extended Abstract). In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 606–609. Springer, Heidelberg (1990)
47. Stern, J.: A Method for Finding Codewords of Small Weight. In: Cohen, G., Godlewski, P. (eds.) Coding Theory 1986. LNCS, vol. 311, pp. 106–113. Springer, Heidelberg (1988)
48. Stern, J.: A New Identification Scheme Based on Syndrome Decoding. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 13–21. Springer, Heidelberg (1994)