How Practical is Public-Key Encryption Based on LPN and Ring-LPN?

Ivan Damgård[1] and Sunoo Park[2]

[1]Aarhus University
[2]MIT

Abstract

We conduct a study of public-key cryptosystems based on variants of the Learning Parity with Noise (LPN) problem. The main LPN variant in consideration was introduced by Alekhnovich (FOCS 2003), and we describe several improvements to the originally proposed scheme, inspired by similar existing variants of Regev's LWE-based cryptosystem. To achieve further efficiency, we propose the first public-key cryptosystem based on the ring-LPN problem, which is a more recently introduced LPN variant that makes for substantial improvement in terms of both time and space. We also introduce a variant of this problem called the transposed Ring-LPN problem. Our public-key scheme based on this problem is even more efficient. For all cases, we compute the parameters required for various security levels in practice, given the best currently known attacks.

Our conclusion is that the basic LPN-based scheme is in several respects not competitive with existing practical schemes, as the public key, ciphertexts and encryption time become very large already for 80-bit security. On the other hand, the scheme based on transposed Ring-LPN is far better in all these respects. Although the public key and ciphertexts are still larger than for, say, RSA at comparable security levels, they are not prohibitively large; moreover, for decryption, the scheme outperforms RSA for security levels of 112 bits or more. The Ring-LPN based scheme is less efficient, however. Thus, LPN-based public-key cryptography seems to be somewhat more promising for practical use than has been generally assumed so far.

Keywords LPN, ring-LPN, public-key encryption.

1 Introduction

The decisional LPN problem is that of distinguishing from random a set of samples, each of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$, where $\mathbf{a} \in \mathbb{Z}_2^n$ is uniformly random (for some parameter $n \in \mathbb{N}$), $e \leftarrow \mathsf{Ber}_\tau$ where Ber_τ denotes the Bernoulli distribution (with some parameter $\tau \in \mathbb{R}$), and $\mathbf{s} \in \mathbb{Z}_2^n$ is a random secret fixed over all samples. In the search version of the problem, the goal is to find the secret vector $\mathbf{s}$. A more detailed definition is given in Section 2.

LPN samples are computationally very simple to generate, but the problem nevertheless seems to be very hard. The two main types of non-trivial attack on LPN are exhaustive search over possible error vectors, and a series of attacks based on the Blum-Kalai-Wasserman (BKW) algorithm [BKW03]. The original BKW algorithm was estimated to have slightly subexponential time complexity of $2^{O(n/\log n)}$ for $2^{O(n/\log n)}$ samples. Subsequent work by Lyubashevsky gave a variant algorithm with runtime $2^{O(n/\log\log n)}$ for $n^{1+\epsilon}$ samples [Lyu05]. A further modification proposed more recently by Kirchner [Kir11] achieved better runtimes specifically for small τ: his algorithm runs in time $O(2^{\sqrt{n}})$ with $O(n)$ queries when $\tau = O(1/\sqrt{n})$ (this is particularly relevant for the public-key setting). Practical implementations of optimised variants of the above algorithms were done by Levieil and Fouque [LF06] and Bernstein and Lange [BL12].

The computational simplicity of LPN makes it very attractive for cryptographic applications, and indeed, many applications of the "symmetric crypto" type have been suggested [HB01; JW05; GRS08; App+09; KSS10]. Doing public-key cryptography based on LPN seems to be much harder; however, in [Ale03], Alekhnovich suggested a public-key cryptosystem based on a variant of the decisional LPN problem, where the noise rate τ is not constant as in standard LPN, but decreases with increasing n (in fact, $\tau \approx 1/\sqrt{n}$). While this problem might be easier than LPN with constant τ, no separation between the problems in the sense of asymptotic complexity is known.

In [Hey+12], the ring-LPN problem was introduced. This can be thought of as a variant of LPN as follows: suppose we are given n samples of the form described above and arrange the random vectors $\mathbf{a}$ as rows in a matrix $\mathbf{A}$. Then what the adversary is given is of form $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ where each entry in $\mathbf{e}$ is chosen according to the Bernoulli distribution. In ring-LPN, the matrix $\mathbf{A}$ is not chosen at random but instead such that it represents an element r in a ring $R = \mathbb{F}_2[X]/(g)$, where g is a polynomial of degree n. The effect of this is that we can specify $\mathbf{A}$ succinctly by just giving the ring element r and the expensive product $\mathbf{As}$ can be replaced by a much faster multiplication in R. The price is that the assumption is now stronger because we need to assume LPN is hard even when $\mathbf{A}$ has the special structure described.

Our contribution. The question we study in this paper is:

> Given what we know about LPN, how (un)attractive is public-key cryptography based on LPN as an alternative to more well known cryptosystems *in practice*?

We are not aware of any previous attempts to determine a precise answer to this. It seems that a widespread perception among cryptographers has been that LPN-based public-key cryptography must "of course" be totally impractical: that Alekhnovich's version of the LPN problem seems to be easier than standard LPN (due to the limited noise rate), so to ensure security would require huge values of n that would render the whole scheme impracticable. However, it is important to consider that for a practical application of an LPN-based scheme, one must choose concrete values of parameters n and τ, and what then matters is not the asymptotic complexity of solving the underlying problem, but whether those concrete values are vulnerable to attack by state-of-the-art algorithms. Besides, it seems natural to consider whether using the Ring-LPN problem would help in terms of efficiency.

In this paper we study some variants of Alekhnovich's original cryptosystem, which are favourable for analysis as well as for practical efficiency reasons. A basic version of this encryption scheme was first communicated to us by Cash [Cas12], and seems to be folklore, at least in some parts of the community, but we were not able to find any published record of the variants that we consider.

The CCA-secure public-key encryption schemes recently proposed by Döttling, Müller-Quade, and Nascimento [DMQN12] and Kiltz, Masny, and Pietrzak [KMP14] use some similar ideas to the schemes we consider in this paper; however, their constructions are relatively complex in order to achieve CCA security, whereas in this paper we aim to see how simple and efficient a construction we can achieve under practical CPA security.

The basic scheme we consider is similar in structure to Regev's cryptosystem based on the hardness of the Learning With Errors (LWE) problem [Reg05]. Due to this resemblance, we are able to improve the basic scheme to get a better plaintext to ciphertext size ratio in a way similar to a corresponding improvement of Regev's scheme by Peikert, Vaikuntanathan, and Waters [PVW07]. The idea behind several of our proofs of security can be traced to an invited talk given by Micciancio [Mic10] (although the reader should be aware that the talk was primarily about encryption based on LWE).

We also propose two new cryptosystems where we exploit the trick from ring-LPN of specifying a public matrix more succinctly and use the ring structure to implement encryption by multiplication in the ring. As a result, both the public key size and encryption time become essentially linear in the security parameter, instead of quadratic. The first scheme is based on the Ring-LPN problem [Hey+12] and is inspired by similar schemes based on the ring-LWE problem. While this scheme superficially seems to be more efficient than the LPN-based variants, the decryption has a more complicated structure and therefore introduces a larger decryption error. We therefore also suggest a different problem we call Transposed Ring-LPN, which appears to be incomparable to Ring-LPN in terms of difficulty. We construct a cryptosystem based on Transposed Ring-LPN. It has an even simpler decryption algorithm with smaller error.

To analyse the concrete efficiency of our schemes, we consider in Section 4 how the best known attacks would perform against the LPN instances used in the LPN-based cryptosystems we propose. We aim for a 25% probability of incorrect decryption of an encrypted bit, so that incorrect decryptions may be corrected using error correcting codes with a manageable expansion factor of about 5. We find that for 80-, 112-, and 128-bit security, respectively, $n = 9000, 21000$, and 29000 are suitable. These security levels (are thought to) correspond to 1024-, 2048-, and 3072-bit RSA.

This means that for the basic LPN-based scheme, public keys will be prohibitively large: several megabytes already for 80-bit security. For the ring-LPN based scheme, however, the situation is much better: our most efficient scheme is based on Transposed Ring-LPN, and taking 128-bit security as an example, the public key would have size about 230kB, and to send an encrypted 128-bit symmetric key, we would need to send about 36kB. These numbers are much larger than for, say, RSA, but they do not seem totally impractical. As for computing time, our implementation of LPN decryption (to reconstruct a 128-bit symmetric key) outperforms RSA by a factor of about 4.5. The corresponding encryption takes time comparable to a 3000-bit full scale exponentiation (but is of course much slower than RSA with small public exponent). On the other hand, the scheme based on Ring-LPN is not competitive. This is because the decryption error probability is larger, therefore we need to decrease LPN noise parameter to maintain a reasonable decryption error. This in turn forces us to increase n to maintain the security level, and this causes a loss of efficiency.

We did not compare to timings for elliptic curve cryptography (ECC). We expect, however, that LPN decryption will be less competitive here because keys for ECC do not have to grow as fast with increasing security as in RSA. Finally, it should be noted that for security against quantum attacks, neither RSA nor ECC are secure, so one should instead compare to LWE-based cryptosystems; however, this is not in scope of this paper.

In conclusion, we find that LPN-based public-key cryptography is somewhat more practical than the general perception seems to have been so far, at least given current state-of-the-art of attacks and if one believes our computational assumption. It might even be competitive in applications where decryption time is the bottleneck, and security of 112 bits or more is desired.

2 The Cryptosystems

2.1 Learning Parity with Noise

We begin by establishing some notation and formally defining the LPN problem [Blu+94].

Notation. Ber_τ denotes the Bernoulli distribution with parameter τ. Ber_τ^k denotes the distribution of vectors in $\mathbb{Z}_2^k$ where each entry of the vector is drawn independently from Ber_τ. $\mathsf{Bin}_{n,\tau}$ denotes the binomial distribution with n trials, each with success probability τ. $x \leftarrow D$ means that x is drawn from distribution D, and $x \leftarrow S$ means that x is drawn uniformly at random from the set S. A probability $\varepsilon(n)$ is said to be negligible if $\varepsilon(n) \leq 1/p(n)$ for any polynomial p and all large enough n. Where it is clear from context, we sometimes use the term "indistinguishable" in lieu of "computationally indistinguishable".

For a vector $\mathbf{w}$, let w_i denote its i^{th} entry; and for a matrix $\mathbf{W}$, let $\mathbf{w}_i$ denote its i^{th} column, and let $w_{i,j}$ denote the j^{th} entry of its i^{th} row.

Definition 2.1 (Decisional LPN problem)**.** *Take parameters $n \in \mathbb{N}$ and $\tau \in \mathbb{R}$ with $0 < \tau < 0.5$ (the noise rate). A distinguisher D is said to (q, t, ε)-solve the decisional $\mathsf{LPN}_{n,\tau}$ problem if*

$$\left| \Pr_{\mathbf{s},\mathbf{A},\mathbf{e}}[\mathsf{D}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = 1] - \Pr_{\mathbf{r},\mathbf{A}}[\mathsf{D}(\mathbf{A}, \mathbf{r}) = 1] \right| \geq \varepsilon$$

where $\mathbf{s} \leftarrow \mathbb{Z}_2^n$, $\mathbf{A} \leftarrow \mathbb{Z}_2^{q \times n}$, and $\mathbf{r} \leftarrow \mathbb{Z}_2^q$ are uniformly random and $\mathbf{e} \leftarrow \mathsf{Ber}_\tau^q$, and the distinguisher runs in time at most t.

Note that adding the "Bernoulli noise" is essential to make the problem non-trivial, since otherwise the secret can be easily found by Gaussian elimination given $O(n)$ samples.

The decisional and search variants of the LPN problem are polynomially equivalent, meaning that an attack requiring q samples against decisional LPN implies an attack against search LPN requiring polynomial in q samples. More precisely:

Lemma 2.2 (Lemma 1 from [KSS10])**.** *If there exists a distinguisher D that (q, t, ε)-solves the decisional $\mathsf{LPN}_{n,\tau}$ problem, then there is a distinguisher D' that (q', t', ε')-solves the search $\mathsf{LPN}_{n,\tau}$ problem where $q' = O(q \log n/\varepsilon^2)$, $t' = O(tn \log n/\varepsilon^2)$, and $\varepsilon' = \varepsilon/4$.*

The first cryptosystem shall be based on the following computational assumption.

Definition 2.3 (Decisional LPN assumption, DLPN)**.** *For any probabilistic algorithm D that (q, t, ε)-solves the decisional $\mathsf{LPN}_{n,\tau}$ problem for all large enough n, where τ is $\Theta(1/\sqrt{n})$, t is polynomial in n and q is $O(n)$, it holds that ε is negligible as a function of n.*

Note the additional assumption on the size of τ, compared to Definition 2.1. This restriction was introduced in [Ale03] and, in all known LPN-based public-key cryptosystems, it seems to be required for correctness.

2.2 LPN Cryptosystems

We define the basic LPN cryptosystem as follows.

Definition 2.4 (Basic LPN cryptosystem)**.** *The key generation, encryption, and decryption functions of the basic LPN cryptosystem are given below. The parameters are $n \in \mathbb{N}$, the length of the secret, and $\tau \in \mathbb{R}$, the noise rate. All operations are performed over $\mathbb{Z}_2$.*

- $\mathsf{BasicKeyGen}()$*: Choose a secret key $\mathbf{s} \in \mathbb{Z}_2^n$. The public key is $(\mathbf{A}, \mathbf{b})$, where $\mathbf{A} \leftarrow \mathbb{Z}_2^{2n \times n}$, $\mathbf{e} \leftarrow \mathsf{Ber}_\tau^{2n}$ is the error vector, and $\mathbf{b} = \mathbf{As} + \mathbf{e}$.*
- $\mathsf{BasicEnc}(pk = (\mathbf{A}, \mathbf{b}), v)$*: To encrypt message bit $v \in \mathbb{Z}_2$, choose $\mathbf{f} \leftarrow \mathsf{Ber}_\tau^{2n}$ and output the ciphertext $(\mathbf{u}, c)$ where $\mathbf{u} = \mathbf{f}^T\mathbf{A}$ and $c = \mathbf{f}^T\mathbf{b} + v$.*

- $\mathsf{BasicDec}(sk = \mathbf{s}, (\mathbf{u}, c))$*: The decryption is* $d = c + \langle \mathbf{u}, \mathbf{s} \rangle$.

We now prove correctness and security for the basic LPN cryptosystem. Some supporting lemmas are needed.

Lemma 2.5. *Let* $X \sim \mathsf{Bin}_{n,\tau}$*. Then the probability that* X *is even is* $\frac{1}{2} + \frac{(1-2\tau)^n}{2}$.

Proof. The probability generating function of X is $G_X(z) = ((1-\tau) + \tau z)^n$. Define $G(z) = \frac{1}{2}(G_X(z) + G_X(-z))$. Then since terms with odd powers cancel out, $G(z) = \sum_{k=0}^{n} z^{2k} \Pr[X = 2k]$, so $G(1)$ is equal to the total probability that X takes an even value: $\Pr[X \text{ is even}] = G(1) = \frac{1}{2}(G_X(1) + G_X(-1)) = \frac{1}{2} + \frac{(1-2\tau)^n}{2}$. □

Lemma 2.6. *For any* k *such that* $\lim_{n\to\infty} \frac{n}{k} = \infty$*, it holds that* $\lim_{n\to\infty}(1 + \frac{k}{n})^n = e^k$.

Proof. Take any k such that $\lim_{n\to\infty} \frac{n}{k} = \infty$. Then:

$$\lim_{n\to\infty}(1+\frac{k}{n})^n = \lim_{\frac{n}{k}\to\infty}(1+\frac{k}{n})^{\frac{n}{k}\cdot k} = \lim_{n'\to\infty}(1+\frac{1}{n'})^{n'\cdot k} = e^k.$$

□

Lemma 2.7 (Correctness)**.** *For any constant* $\varepsilon > 0$*, it holds that* τ *can be chosen with* $\tau = \Theta(\frac{1}{\sqrt{n}})$ *such that the probability of correct decryption by* $\mathsf{BasicDec}$ *is at least* $1 - \varepsilon$.

Proof. The decrypted bit d is equal to the correct plaintext v if and only if $\mathbf{f}^T\mathbf{e} = 0$, since $d = c + \mathbf{s}^T\mathbf{u} = \mathbf{f}^T\mathbf{b} + v + \mathbf{s}^T\mathbf{f}^T\mathbf{A} = \mathbf{f}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) + v + \mathbf{s}^T\mathbf{f}^T\mathbf{A} = \mathbf{f}^T\mathbf{e} + v$. Let e_i and f_i denote the entries of $\mathbf{e}$ and $\mathbf{f}$ respectively. Define $C_i = e_i \cdot f_i$. Then these $C_i \sim \mathsf{Ber}_{\tau^2}$, independently and identically. Let $C = \sum_i C_i \sim \mathsf{Bin}_{2n,\tau^2}$.

Observe that $\mathbf{f}^T\mathbf{e} = 0$ if and only if C takes an even value. From Lemma 2.5, then, $\Pr[\mathbf{f}^T\mathbf{e} = 0] = \frac{1}{2} + \frac{(1-2\tau^2)^{2n}}{2}$. Take $0 < \tau \leq O(\frac{1}{\sqrt{n}})$: for τ in this range, $\tau^2 n = O(1)$, so $\lim_{n\to\infty} \frac{n}{\tau^2 n} = \infty$. Applying Lemma 2.6 yields: $\lim_{n\to\infty}(1-2\tau^2)^{2n} = e^{-2\tau^2(2n)}$. Hence, for large n, $\Pr[\mathbf{f}^T\mathbf{e} = 0] \approx \frac{1+e^{-2\tau^2(2n)}}{2}$. If $\tau = \frac{c}{\sqrt{n}}$ for some constant c, then the exponent $-2\tau^2(2n)$ is constant. Observe that $\lim_{c\to 0} -2\tau^2(2n) = 0$, so $\lim_{c\to 0} \frac{1+e^{-2\tau^2(2n)}}{2} = 1$. It follows that for $\tau = \Theta(\frac{1}{\sqrt{n}})$, for any constant $\varepsilon > 0$, the probability of correct decryption by $\mathsf{BasicDec}$ is at least $1 - \varepsilon$ provided that c is chosen sufficiently close to 0. □

Remark. Provided that the decryption error rate is low enough, error correcting codes may be employed to essentially eliminate the possibility of incorrectly received bits (for this we need messages of multiple bits, which shall be addressed in more detail later).

Lemma 2.8 (Pseudorandom public keys)**.** *Under the DLPN assumption, the distribution of the public keys* $(\mathbf{A}, \mathbf{b})$ *generated by* $\mathsf{BasicKeyGen}$ *is computationally indistinguishable from uniform over* $\mathbb{Z}_2^{2n\times n} \times \mathbb{Z}_2^{2n}$.

Proof. The public keys generated by $\mathsf{BasicKeyGen}$ are of the form $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$, where $\mathbf{A}$, $\mathbf{s}$, and $\mathbf{e}$ are chosen as in Definition 2.1. Since furthermore q and τ are chosen as in the DLPN assumption, the required indistinguishability follows immediately. □

Lemma 2.9. *For* $m \geq dn$ *for a constant* $d > 1$*, let* $\chi_{m,n}$ *be the distribution of matrices* $\mathbf{M} \in \mathbb{Z}_2^{m\times n}$ *which are sampled by choosing the columns to be a uniformly random linearly independent set. For large* n*,* $\chi_{m,n}$ *is statistically indistinguishable from the uniform distribution over* $\mathbb{Z}_2^{m\times n}$.

Proof. A matrix sampled from the uniform distribution over $\mathbb{Z}_2^{m\times n}$ is (perfectly) indistinguishable from one constructed by taking n column vectors of m bits drawn uniformly from $\mathbb{Z}_2^m$, since matrix columns are independent in the former distribution. For $m \geq dn$, consider generating a matrix by drawing the columns one by one. Each time a new column is drawn, it lies outside the subspace spanned by the column vectors already drawn, except with negligible probability. Therefore, with overwhelming probability, a matrix sampled from $\mathbb{Z}^{m\times n}$ will have full rank, and the result follows. □

Lemma 2.10. *Under the DLPN assumption,* $(\mathbf{S}, \mathbf{f}^T\mathbf{S})$ *is computationally indistinguishable from* $(\mathbf{S}, \mathbf{r})$*, where* $\mathbf{f} \leftarrow \mathsf{Ber}_\tau^{2n}$*,* $\mathbf{S} \leftarrow \mathbb{Z}_2^{2n\times(n+1)}$ *and* $\mathbf{r} \leftarrow \mathbb{Z}_2^{n+1}$*.*

Proof. Take an LPN sample of the form $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ constructed as detailed in Definition 2.1, with $q = 2n + 2$. By Lemma 2.9, this is computationally indistinguishable from $(\mathbf{A}', \mathbf{A}'\mathbf{s} + \mathbf{e})$ where $\mathbf{A}' \sim \chi_{2n+2,n+1}$. Let $\mathbf{H} \in \mathbb{Z}_2^{(2n+2)\times(n+1)}$ be sampled by choosing the column vectors as a uniformly random basis for the orthogonal complement $C \subseteq \mathbb{Z}_2^{2n+2}$ of the columns of A'. C is determined uniformly randomly by the choice of $\mathbf{A}'$, so $\mathbf{H}$ is distributed according to $\chi_{2n+2,n+1}$. It follows, by Lemma 2.9, that $\mathbf{H}$ is indistinguishable from uniformly random.

By construction, $\mathbf{H}^T\mathbf{A} = 0$, so $\mathbf{H}^T(\mathbf{As} + \mathbf{e}) = \mathbf{H}^T\mathbf{e}$. This means that since, under DLPN, $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ is indistinguishable from random, so is $(\mathbf{H}, \mathbf{H}^T\mathbf{e})$. The lemma follows from transposing the last component and truncating to the required dimensions. □

Theorem 2.11 (Security)**.** *Under the DLPN assumption, the basic LPN cryptosystem is secure against chosen plaintext attack.*

Proof. Consider an instance of the basic LPN cryptosystem with parameters n and τ, where the public key is $(\mathbf{A}, \mathbf{b})$. Let $\mathbf{R} \in \mathbb{Z}_2^{2n\times(n+1)}$ be defined as follows.

$$r_{i,j} = \begin{cases} a_{i,j} & \text{for } 1 \leq i \leq 2n, 1 \leq j \leq n \\ b_i & \text{for } 1 \leq i \leq 2n, j = n+1 \end{cases}$$

So $\mathbf{R}$ has the same distribution as the public key. Note that a ciphertext of the message 0 is of the form $(\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b})$ as defined by $\mathsf{BasicEnc}$, which can also be written as $\mathbf{f}^T\mathbf{R}$. We now argue about the joint distribution of public key and ciphertext as follows:

$(\mathbf{R}, \mathbf{f}^T\mathbf{R})$ is indistinguishable from $(\mathbf{S}, \mathbf{f}^T\mathbf{S})$ where $\mathbf{S}$ is uniformly random by Lemma 2.8. Furthermore $(\mathbf{S}, \mathbf{f}^T\mathbf{S})$ is indistinguishable from $(\mathbf{S}, \mathbf{r})$ where $\mathbf{r}$ is random, by Lemma 2.10.

A similar argument easily implies that public key and a ciphertext of the message 1 has distribution indistinguishable from $(\mathbf{S}, \mathbf{r}')$ where $\mathbf{r}'$ is obtained by generating a random vector and flipping the last bit. But this is the same distribution as $(\mathbf{S}, \mathbf{r})$, so we have shown that public key and ciphertexts of 0 and 1 have indistinguishable distributions. The theorem follows. □

The basic LPN cryptosystem can be significantly improved in efficiency by a relatively simple modification reducing the ciphertext expansion factor (the ratio of ciphertext to plaintext length) from $\tilde{O}(n)$ to as low as $O(1)$. This is achieved by re-using the encryption randomness over up to $\ell = O(n)$ public key rows. To allow for this, we require ℓ independent secret keys $\mathbf{s}_i$ and ℓ independent error vectors $\mathbf{e}_i$ – thus, the secret key size increases from $O(n)$ to $O(n^2)$, while the public key size remains asymptotically unchanged. The efficacy of the modification is based on the fact that a large part of the time taken by the original cryptosystem's operations is due to the large matrix $\mathbf{A}$.

This modification to the LPN cryptosystem is very similar in structure to the modification to Regev's LWE-based cryptosystem proposed by Peikert, Vaikuntanathan, and Waters [PVW07]. In the context of LPN, a similar idea was mentioned by Pietrzak in [Pie12].

Definition 2.12 (Multi-bit LPN cryptosystem)**.** *Parameters n and τ below are as in Definition 2.4. Additionally, we introduce $\ell = O(n)$, the length of plaintext that can be encrypted in a single operation. Let $\widetilde{n} = \max(n, \ell)$.*

- $\mathsf{MultiBitKeyGen}()$: *Choose a secret key* $\mathbf{S} \leftarrow \mathbb{Z}_2^{n\times\ell}$. *The public key is* $(\mathbf{A}, \mathbf{B})$, *where* $\mathbf{A} \leftarrow \mathbb{Z}_2^{2\widetilde{n}\times n}$, $\mathbf{E} \leftarrow \mathsf{Ber}_\tau^{2\widetilde{n}\times\ell}$, *and* $\mathbf{B} = \mathbf{AS} + \mathbf{E}$.
- $\mathsf{MultiBitEnc}(pk = (\mathbf{A}, \mathbf{B}), \mathbf{v})$: *To encrypt message* $\mathbf{v} \in \mathbb{Z}_2^\ell$, *choose* $\mathbf{f} \leftarrow \mathsf{Ber}_\tau^{2\widetilde{n}}$ *and output the ciphertext* $(\mathbf{u}, \mathbf{c})$ *where* $\mathbf{u} = \mathbf{f}^T\mathbf{A}$ *and* $\mathbf{c} = \mathbf{f}^T\mathbf{B} + \mathbf{v}^T$.
- $\mathsf{MultiBitDec}(sk = \mathbf{S}, (\mathbf{u}, \mathbf{c}))$: *The decryption is* $\mathbf{d}$ *such that* $\mathbf{d}^T = \mathbf{c} + \mathbf{S}^T\mathbf{u}$.

We now prove correctness and security for the multi-bit cryptosystem.

Lemma 2.13 (Correctness)**.** *For each fixed choice of* $\mathbf{f}$, *encryption followed by decryption of a message* $\mathbf{v}$ *in the multi-bit LPN cryptosystem is equivalent to sending each bit of* $\mathbf{v}$ *through a binary symmetric channel with some error probability* ρ. *Furthermore, for any constant* $\varepsilon > 0$, τ *can be chosen with* $\tau = \Theta(\frac{1}{\sqrt{n}})$ *such that, except with negligible probability,* $\rho \leq \varepsilon$.

Proof. The decryption is equal to the correct plaintext $\mathbf{v}$ if and only if $\mathbf{f}^T\mathbf{E} = \mathbf{0} \in \mathbb{Z}_2^\ell$, since $\mathbf{d}^T = \mathbf{c} + \mathbf{S}^T\mathbf{u} = \mathbf{f}^T\mathbf{B} + \mathbf{v}^T + \mathbf{S}^T\mathbf{f}^T\mathbf{A} = \mathbf{f}^T(\mathbf{AS} + \mathbf{E}) + \mathbf{v}^T + \mathbf{S}^T\mathbf{f}^T\mathbf{A} = \mathbf{f}^T\mathbf{E} + \mathbf{v}^T$. Let $|\mathbf{f}|$ denote the Hamming weight of $\mathbf{f}$. For any given $\mathbf{f}$, the independence of the columns $\mathbf{e}_i$ of $\mathbf{E}$ implies that the transmitted bits do indeed go independently through a noisy channel, which has error probability determined by $|\mathbf{f}|$. By Lemma 2.5, for any given weight $|\mathbf{f}|$, it holds that $\Pr[\mathbf{f}^T\mathbf{e}_i = 0] = \frac{1}{2} + \frac{(1-2\tau)^{|\mathbf{f}|}}{2}$.

Let A denote the event that $|\mathbf{f}| \leq 3\widetilde{n}\tau$. Note that $|\mathbf{f}| \sim \mathsf{Bin}_{2\widetilde{n},\tau}$. By a Chernoff bound, $\Pr[A] < 1 - \left(\frac{\sqrt{e}}{1.5\sqrt{1.5}}\right)^{2\widetilde{n}\tau}$. Since $\left(\frac{\sqrt{e}}{1.5\sqrt{1.5}}\right)^{2\widetilde{n}\tau}$ is exponentially small in $\widetilde{n}\tau$, and $\widetilde{n}\tau = O(n\tau) \to \infty$ as $n \to \infty$, A occurs with overwhelming probability. If A occurs, then the encryption followed by decryption of ℓ bits in the multi-bit LPN cryptosystem is equivalent to sending the bits through a binary symmetric channel with error probability ρ which is at most that of the case where $|\mathbf{f}| = \lfloor 3\widetilde{n}\tau \rfloor$. From Lemma 2.7, the result follows. □

Lemma 2.14 (Pseudorandom public keys)**.** *Under the DLPN assumption, the distribution of the public keys* $(\mathbf{A}, \mathbf{B})$ *generated by* $\mathsf{MultiBitKeyGen}$ *is computationally indistinguishable from uniform over* $\mathbb{Z}_2^{2\widetilde{n}\times n} \times \mathbb{Z}_2^{2\widetilde{n}\times\ell}$.

Proof. We define hybrid distributions $H_0, \cdots, H_\ell$ over matrices $(\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_2^{2\widetilde{n}\times n} \times \mathbb{Z}_2^{2\widetilde{n}\times\ell}$ such that in distribution H_k, the matrix $\mathbf{A}$ and the first k columns of $\mathbf{B}$ are uniformly randomly chosen, and the other columns of $\mathbf{B}$ are chosen according to the procedure for generating a column of $\mathbf{B}$ given by $\mathsf{MultiBitKeyGen}$ (for parameters n and τ). Then H_0 is exactly the distribution of the public keys generated by $\mathsf{MultiBitKeyGen}$, and H_ℓ is completely uniform over $\mathbb{Z}_2^{2\widetilde{n}\times n} \times \mathbb{Z}_2^{2\widetilde{n}\times\ell}$.

For any $k \in \{0, \cdots, \ell-1\}$, we define a simulator $\mathcal{S}_k$ which has access to an oracle $\mathcal{O}$ that returns samples in $\mathbb{Z}_2^{2\widetilde{n}\times n} \times \mathbb{Z}_2^{2\widetilde{n}}$ that are either chosen uniformly at random, or are of the form $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ as specified in the LPN problem (in Definition 2.1). $\mathcal{S}_k$ outputs a pair $(\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_2^{2\widetilde{n}\times n} \times \mathbb{Z}_2^{2\widetilde{n}\times\ell}$ constructed as follows. First, $\mathcal{O}$ is queried, yielding a sample $(\mathbf{A}', \mathbf{b}')$. Then, $\mathcal{S}_k$ sets $\mathbf{A}$ equal to $\mathbf{A}'$; uniformly randomly chooses the first k columns of the output matrix $\mathbf{B}$; sets the column $\mathbf{b}_{k+1}$ equal to $\mathbf{b}'$; and for all $k < j \leq \ell$, $\mathcal{S}_k$ chooses independent secret vectors $\mathbf{s}_j \in \mathbb{Z}_2^n$ uniformly at random, and independent error vectors $\mathbf{e}_j \in \mathbb{Z}_2^{2\widetilde{n}}$ according to $\mathsf{Ber}_\tau^{2\widetilde{n}}$, and sets $\mathbf{b}_j = \mathbf{As}_j + \mathbf{e}_j$.

Observe that if $\mathcal{O}$ samples from the uniform distribution, then the output of $\mathcal{S}_k$ has distribution H_k, and otherwise, $\mathcal{O}$ samples from the LPN distribution, so the output of $\mathcal{S}_k$ has distribution H_{k+1}. It follows that if $\mathsf{LPN}_{n,\tau}$ is hard, then H_k and H_{k+1} are computationally indistinguishable for all $j \in \{0, \cdots, \ell-1\}$. Therefore, H_ℓ is computationally indistinguishable from uniformly random H_0. □

Lemma 2.15. *Under the DLPN assumption, for any* $n, \ell \in \mathbb{N}$, $(\mathbf{S}, \mathbf{f}^T\mathbf{S})$ *is computationally indistinguishable from* $(\mathbf{S}, \mathbf{r})$, *where* $\mathbf{f} \leftarrow \mathsf{Ber}_\tau^{2\cdot\max(n,\ell)}$, $\mathbf{S} \leftarrow \mathbb{Z}_2^{2\cdot\max(n,\ell)\times(n+\ell)}$, *and* $\mathbf{r} \leftarrow \mathbb{Z}_2^{n+\ell}$.

Proof. Exactly as in Lemma 2.10; the only difference is that Lemma 2.10 treats the case $\ell = 1$. □

Theorem 2.16 (Security)**.** *Under the DLPN assumption, the multi-bit LPN cryptosystem is secure against chosen plaintext attack.*

Proof. Consider an instance of the multi-bit LPN cryptosystem with parameters n and τ, with $\ell = O(n)$ and $\widetilde{n} = \max(n, \ell)$, where the public key is $(\mathbf{A}, \mathbf{B})$. Given the public key, one can construct a matrix $\mathbf{R} \in \mathbb{Z}_2^{2\widetilde{n} \times (n+\ell)}$ with entries as follows:

$$r_{i,j} = \begin{cases} a_{i,j} & \text{for } 1 \le i \le 2\widetilde{n}, 1 \le j \le n \\ b_i & \text{for } 1 \le i \le 2\widetilde{n}, n+1 \le j \le n+\ell \end{cases}$$

So $\mathbf{R}$ has the same distribution as the public key. Note that a ciphertext in the multi-bit LPN cryptosystem is of the form $(\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{B} + \mathbf{v})$ as specified in Definition 2.12. The ciphertext can also be written as $\mathbf{f}^T\mathbf{R} \diamond \mathbf{v}$, and where the $\diamond$ operation denotes "add $\mathbf{v}$ to the last ℓ entries in $\mathbf{f}^T\mathbf{R}$". We now argue about the joint distribution of public key and ciphertext as follows:

$(\mathbf{R}, \mathbf{f}^T\mathbf{R} \diamond \mathbf{v})$ is indistinguishable from $(\mathbf{S}, \mathbf{f}^T\mathbf{S} \diamond \mathbf{v})$ where $\mathbf{S}$ is random by Lemma 2.14. Furthermore $(\mathbf{S}, \mathbf{f}^T\mathbf{S} \diamond \mathbf{v})$ is indistinguishable from $(\mathbf{S}, \mathbf{r} \diamond \mathbf{v})$ where $\mathbf{r}$ is random, by Lemma 2.15. Since in this last distribution, $\mathbf{r}$ is sampled independently of $\mathbf{v}$, the distribution of $\mathbf{r} \diamond \mathbf{v}$ is the uniform distribution on $\mathbb{Z}_2^{n+\ell}$ and in particular independent of $\mathbf{v}$. The result follows. □

2.3 Cryptosystems from Ring-LPN and Variants

In this section we present cryptosystems based on ring-LPN and variant assumptions. The ring-LPN assumption (defined formally below) was introduced by Heyse *et al.* in [Hey+12]. In this paper, we introduce a closely related variant assumption called the *transposed ring-LPN assumption*. The encryption schemes based on these assumptions are substantially more efficient and have smaller public keys, compared to the LPN-based schemes discussed so far.

2.3.1 Ring-LPN Assumption

The ring-LPN problem may be viewed as a natural variant of the LPN problem, with some additional structure that allows very fast multiplication and compact representations of samples. Since standard LPN has formed the basis of many cryptographic constructions, the ring-LPN problem is considered an interesting candidate for constructing more efficient systems building upon existing LPN-based schemes, both from a design perspective (e.g. the "Lapin" authentication scheme [Hey+12]) and a cryptanalytic perspective (e.g. Bernstein and Lange's attack against ring-LPN [BL12], which was inspired by Lapin).

Notation. For a polynomial ring $R \in \mathbb{F}_2[X]/(g)$, the distribution Ber_τ^R denotes the distribution over R, where each of the coefficients of the polynomial is drawn independently from Ber_τ. For a polynomial $r \in R$, let $|r|$ denote the weight of r, i.e. the number of nonzero coefficients r has. Let $r[i] \in \mathbb{Z}_2$ denote the coefficient of x^i in r.

Definition 2.17 (Decisional ring-LPN problem)**.** *Take parameters $R = \mathbb{F}_2[X]/(g)$ with $g \in \mathbb{F}_2[X]$ of degree $n-1$, and $\tau \in \mathbb{R}$ with $0 < \tau < 0.5$. Let $\mathcal{U}^R$ denote the uniform distribution over $R \times R$. For any polynomial $s \in R$, let $\Lambda_\tau^{R,s}$ be the distribution over $R \times R$ whose samples are obtained by choosing a polynomial $r \leftarrow R$ and another polynomial $e \to \mathsf{Ber}_\tau^R$ and outputting $(r, rs + e)$. A distinguisher D is said to (q, t, ε)-solve the decisional $\mathsf{RingLPN}_{n,\tau}^R$ problem if*

$$\left| \Pr[\mathsf{D}^{\Lambda_\tau^{R,s}} = 1] - \Pr[\mathsf{D}^{\mathcal{U}^R} = 1] \right| \ge \varepsilon$$

and the distinguisher runs in time at most t and makes at most q queries.

$$|\Pr[s \leftarrow R : \mathsf{D}]^{\Lambda_\tau^{R,s}} = 1] - \Pr[\mathsf{D}^{\mathcal{U}^R} = 1]| \le \varepsilon.$$

Lemma 2.18. *The ring-LPN problem with $s \leftarrow \mathsf{Ber}_\tau^R$ and the ring-LPN problem with $s \leftarrow R$ are of polynomially equivalent hardness.*

The proof of Lemma 2.18 follows from a transformation shown in the context of LWE by Applebaum *et al.*, in Lemma 2 of [App+09]. For the details, we refer the reader to their paper.

Definition 2.19 (Decisional Ring-LPN assumption, DRLPN)**.** *For any probabilistic algorithm that (q, t, ε)-solves the decisional $\mathsf{RingLPN}_{n,\tau}$ problem for all large enough n, where τ is $\Theta(1/\sqrt{n})$, t is polynomial is n and q is $O(n)$, it holds that ε is negligible as a function of n.*

2.3.2 Transposed Ring-LPN Assumption

Definition 2.20. *Let the $\mathsf{LPN}_{n,\tau}(D)$ problem be the variant of the standard $\mathsf{LPN}_{n,\tau}$ problem in which the matrix $\mathbf{A}$ is drawn from distribution D over $\mathbb{Z}_2^{q\times n}$, and all other aspects of the problem are identical to the standard $\mathsf{LPN}_{n,\tau}$ problem.*

Useful distributions. Let $\Psi^{R,l}$ denote the distribution over $\mathbb{Z}_2^{ln\times n}$ whose samples consist of the vertical concatenation of l square matrices in $\mathbb{Z}_2^{n\times n}$, where each square matrix is independently sampled as $\mathsf{mat}(r)$ for uniformly random $r \leftarrow R$. Let $\Psi_0^{R,l}$ denote the distribution over $\mathbb{Z}_2^{ln\times n}$ whose samples $\mathbf{A} \in \mathbb{Z}_2^{ln\times n}$ are obtained by taking $\mathbf{A'} \leftarrow \Psi^{R,l}$ and choosing the columns of $\mathbf{A}$ uniformly randomly from the orthogonal complement of the columns of $\mathbf{A'}$.

Definition 2.21 (Transposed ring-LPN assumption, TRLPN)**.** *For any probabilistic algorithm that (q, t, ε)-solves, for all large enough n, the decisional $\mathsf{LPN}_{n,\tau}(\Psi^{R,l})$ problem or the decisional $\mathsf{LPN}_{n,\tau}(\Psi_0^{R,l})$ problem, where τ is $\Theta(1/\sqrt{n})$, t is polynomial in n, $l = 2$ (and hence $q = 2n$), it holds that ε is negligible as a function of n.*

Before defining cryptosystems based on TRLPN, let us discuss the assumption. Consider first the $\mathsf{LPN}_{n,\tau}(\Psi^{R,l})$ problem. Here, the adversary is presented with samples of form $\mathsf{mat}(r) \cdot \mathbf{s} + \mathbf{e}$, so he gets noisy inner products of the secret and the rows of $\mathsf{mat}(r)$. Since the i^{th} row in $\mathsf{mat}(r)$ is equal to $\mathsf{vec}(r \cdot X^i)$, observe that although the rows are not independent, each row is uniformly random and the rows are *linearly* independent. It therefore seems plausible to us that $\mathsf{LPN}_{n,\tau}(\Psi^{R,l})$ is hard.[1] Note that if we had instead taken $\mathsf{mat}(r)^T \cdot \mathbf{s} + \mathbf{e}$, the product would have been equivalent to a ring product and we would get the ring-LPN assumption.

As for the $\mathsf{LPN}_{n,\tau}(\Psi_0^{R,l})$ problem, the columns of the public matrix are chosen uniformly from a certain subspace of dimension n. Each row should therefore have large entropy and intuitively, it would seem that the rows would be much less correlated than in the first problem. It therefore seems to us that this second problem is even harder. We emphasise, however, that a much more careful study is needed to gain more confidence in the assumption.

2.3.3 Lightness-Preserving Rings

To construct a cryptosystem based on ring-LPN, we consider polynomial rings $R = \mathbb{F}_2[X]/(g)$ with a special "lightness-preserving" condition on the polynomial g, which is defined below. Informally, the lightness-preserving condition ensures that when two low-weight polynomials are multiplied, the result is also of low weight. We begin by showing some useful lemmas about lightness-preserving rings, then present a new cryptosystem based on the ring-LPN assumption.

Notation. Let $|r|$ denote the weight (number of nonzero coefficients) of a polynomial r in R. For any polynomial $r \in R$ with degree $n-1$, let $\mathsf{vec}(r) \in \mathbb{Z}_2^n$ denote the (row) vector whose i^{th} entry is $r[i]$ for all $0 \le i < n$, and let $\mathsf{mat}(r) \in \mathbb{Z}_2^{n\times n}$ be the matrix such that for all $r' \in R$, $\mathsf{vec}(r') \cdot \mathsf{mat}(r) = \mathsf{vec}(r' \cdot r)$. Note that the matrix $\mathsf{mat}(r)$ has the property that its i^{th} row vector is equal to $\mathsf{vec}(r \cdot X^i)$.

Definition 2.22 (Lightness-preserving ring)**.** *A ring $R = \mathbb{F}_2[X]/(g)$ is* lightness-preserving *if for any polynomial $p \leftarrow \mathbb{F}[X]$ of degree up to $2n-1$, the reduced polynomial $p' = p \mod g$ has weight at most $c \cdot |p|$ for a constant c.*

[1] It is commonly believed that generally, the LPN problem remains hard even if the public matrix is not uniformly random but has sufficiently high min-entropy [Pie12].

Lemma 2.23. *For a ring $R = \mathbb{F}_2[X]/(g)$ to be lightness-preserving (where g is irreducible and has degree n), it is sufficient for g to be of the form $X^n + X^m + 1$ where $m \leq n/2$.*

Proof. Take any $p \in \mathbb{F}_2[X]$ of degree up to $2n - 1$. The following simple algorithm reduces p modulo g. (Note that `x^i` denotes the polynomial X^i.)

```
while degree(p) >= n :
   p := p + (g * x^(degree(p)-n))
```

We refer to the coefficients of $X^n, \ldots, X^{2n-1}$ as the *top half* of p's coefficients. Define the *top quarter*, *second quarter*, *third quarter*, and *bottom quarter* similarly. We will write `p` to refer to the variable `p` in the above algorithm, and use p to refer to the original polynomial in $\mathbb{F}_2[X]$.

The number of times that the `while` loop will be iterated is equal to the number of nonzero coefficients that occur in the top half of `p`'s coefficients at any point during the execution of the reduction algorithm. Call the nonzero coefficients of p at the start of the algorithm the *original bits*. On each iteration of the `while` loop, the addition (exclusive-or) operation causes the most significant nonzero coefficient of `p` to be zeroed, and causes up to two other coefficients of `p` to change. Call the nonzero coefficients that are created by the addition operation the *new bits*.

Note that for any given iteration of the loop, it is only possible for a *new bit* to be created in the top half if the (most significant) coefficient being zeroed in this iteration lies in the top quarter. Moreover, such a new bit will lie in the second quarter, and thus its presence cannot cause more new bits to be created in the top half.

It follows that the maximum number of nonzero coefficients that can occur in the top half of `p`'s coefficients during the algorithm's execution is $2 \cdot |p|$. Since $|g| = 3$, the weight of `p` will increase by at most 3 on each iteration, and therefore the weight of `p` will increase by at most $6 \cdot |p|$ over the course of the whole reduction procedure. Hence, the weight of the reduced polynomial $p' = p \bmod g$ is at most $7 \cdot |p|$. It follows that R is lightness-preserving. □

Note that the requirement that g be of the form given in the above lemma not only facilitates our proofs of correctness for the cryptosystem to follow but also allows for more efficient implementations of reduction modulo g. As for whether such "nice" polynomials exist, it is known that for $n = 3^m$ for some $m \in \mathbb{N}$, the polynomial $X^{2n} + X^n + 1$ is irreducible in $\mathbb{F}_2[X]$. It seems reasonable to believe that they also exist for other values on n, and in any case, one may of course search exhaustively for a polynomial given a concrete value of n.

2.3.4 Ring-LPN Cryptosystem

We presently define a new encryption scheme based on the hardness of the ring-LPN problem.

Definition 2.24 (Ring-LPN cryptosystem)**.** *Parameters are $R = \mathbb{F}_2[X]/(g)$, a polynomial ring with g an irreducible lightness-preserving polynomial of degree n, and $\tau \in \mathbb{R}$, the noise rate.*

- $\mathsf{RingKeyGen}()$*: Choose a secret key $s \leftarrow \mathsf{Ber}_\tau^R$. The public key is (a, b), where $b = as + e$ for $a \leftarrow R$ and $e \leftarrow \mathsf{Ber}_\tau^R$.*
- $\mathsf{RingEnc}(pk = (a, b), v)$*: To encrypt message $v \in R$, choose $f, f', f'' \leftarrow \mathsf{Ber}_\tau^R$, and output the ciphertext (u, c) where $u = af + f'$ and $c = bf + f'' + v$.*
- $\mathsf{RingDec}(sk = s, (u, c))$*: The decryption is $d = c + us$.*

Notice that rather than encrypting single-bit messages, the Ring-LPN cryptosystem encrypts messages represented by ring elements. Because there is some small in the decryption (which is characterised formally below), we would actually want the message $v \in R$ to be an error-correcting encoding of the original plaintext.

The key advantages of this cryptosystem are that compared to the LPN cryptosystem, the public key size has changed from $O(n^2)$ to $O(n)$, and the expensive matrix multiplication in encryption has been replaced by efficient ring multiplication.

We now prove correctness and security of the Ring-LPN cryptosystem.

Lemma 2.25. *Let $R = \mathbb{F}_2[X]/(g)$ be a lightness-preserving ring. For any constant $\varepsilon > 0$, it holds that τ can be chosen with $\tau = \Theta(\frac{1}{\sqrt{n}})$, such that for any independently chosen $e, f \leftarrow \mathsf{Ber}_\tau^R$, the weight of the product polynomial $ef \in R$ is less than εn with overwhelming probability.*

Proof. By a Chernoff bound, it holds that $|e| \leq 2n\tau$ and $|f| \leq 2n\tau$ with overwhelming probability. Therefore, the product polynomial ef in $\mathbb{F}_2[X]$ (not reduced modulo g) has weight at most $(2n\tau)^2$, with overwhelming probability. In this case, since R is lightness-preserving, the reduced product polynomial $ef \mod g$ in R has weight at most $c \cdot (2n\tau)^2$ for constant c. Let $\tau = \frac{c'}{\sqrt{n}}$ for constant c'. Notice $c \cdot (2n\tau)^2 = O(n)$, so for any constant $\varepsilon > 0$ we can choose c' small enough that the product $ef \mod g$ has weight less than εn with overwhelming probability. □

Lemma 2.26 (Correctness)**.** *For any constant $\varepsilon > 0$, it holds that τ can be chosen with $\tau = \Theta(\frac{1}{\sqrt{n}})$, such that the number of message coefficients incorrectly decrypted by running* RingDec *on a ciphertext produced by* RingEnc *is less than εn with overwhelming probability.*

Proof. Take any $i \in \{0, \ldots, n-1\}$, and consider the coefficient of X^i in the decrypted message $d \in R$. Denote this coefficient by d_i. The decrypted coefficient d_i is equal to the correct plaintext coefficient v_i if and only if $(ef + f'' + f's)_i = 0$, since

$$d = c + us = bf + f'' + v + (af + f')s = v + ef + f'' + f's.$$

Let $\tau = \frac{c}{\sqrt{n}}$ for some constant c. By Lemma 2.25, for any $\varepsilon' > 0$, it holds that c can be chosen small enough, so that $|ef + f's| < 2\varepsilon' n$ with overwhelming probability. By a Chernoff bound and since $f'' \leftarrow \mathsf{Ber}_\tau^n$, it holds that $|f''| < 2n\tau = O(\sqrt{n})$ with overwhelming probability. The result follows. □

Lemma 2.27 (Pseudorandom public keys)**.** *Under the DRLPN assumption, the distribution of the public keys $(\mathbf{A}, \mathbf{b})$ generated by* RingKeyGen *is computationally indistinguishable from uniform over $R \times R$.*

Proof. The public keys generated by RingKeyGen are of the form $(a, as + e)$, where a, s, and e are chosen as in Definition 2.17. Since furthermore τ is chosen as in the DRLPN assumption, the required indistinguishability follows immediately. □

Theorem 2.28 (Security)**.** *Under the DRLPN assumption, the ring-LPN cryptosystem is secure against chosen plaintext attack.*

Proof. Consider an instance of the ring-LPN cryptosystem with parameters n and τ, where the public key is (a, b). Note that the public key is indistinguishable from random in $R \times R$, by Lemma 2.27. A ciphertext of the message 0 is of the form $(af + f', bf + f'')$ as defined by RingEnc. We consider the distribution of pairs of public key and ciphertext, so for an encryption of 0 we look at $((a, b), (af + f', bf + f''))$. By the DRLPN assumption, $(a, af + f')$ is indistinguishable from (a, r) for random $r \in R$, and similarly $(b, bf + f'')$ is indistinguishable from (b, r'). Given also the pseudorandomness of public keys, it follows that $((a, b), (af + f', bf + f''))$ is indistinguishable from uniformly random in $(R \times R) \times (R \times R)$.

A ciphertext of the message $v \in R$ is of the form $(af + f', bf + f'' + v)$, that is, it is distributed exactly like a ciphertext of the message 0, except that some bits of the second component are flipped. The bits which are flipped are determined by v, which is independent of the public key and of $f, f'f''$. Since the distribution of ciphertexts of the message 0 is pseudorandom as has already been shown, it holds that the distribution of ciphertexts of the message v is also pseudorandom; and by the independence of v from the other ciphertext components, the ciphertexts are independent of v. The theorem follows. □

2.3.5 Transposed Ring-LPN Cryptosystem

Notation. For matrices $\mathbf{A} \in \mathbb{Z}_2^{m \times n}, \mathbf{B} \in \mathbb{Z}_2^{m' \times n}$, let $\mathbf{A}//\mathbf{B} \in \mathbb{Z}_2^{(m+m') \times n}$ denote their vertical concatenation, that is, $\mathbf{A}//\mathbf{B}$ is the matrix consisting of the rows of $\mathbf{A}$ followed by those of $\mathbf{B}$.

Observe that if $\mathbf{M} = \mathsf{mat}(r)$ is an $n \times n$ matrix and $\mathbf{v}$ is a row vector, then $\mathbf{M}$ can be fully specified by just giving r, and $\mathbf{vM}$ can be computed by a single product in R where we translate $\mathbf{v}$ to an element in R in the natural way.

The key idea underlying the cryptosystem presented below is to modify the schemes already presented, such that the public key matrix is specified by ring elements and the costly vector by matrix product in encryption can be replaced by a product in the ring. As we shall see, this means that the product of the secret with the public matrix (in key generation) is not a ring product: this is why our assumption is not the ring-LPN assumption. We now define the TRLPN cryptosystem.

Definition 2.29 (Transposed Ring-LPN (TRLPN) cryptosystem)**.** *Parameters are $R = \mathbb{F}_2[X]/(g)$, a polynomial ring with g an irreducible polynomial of degree n, and $\tau \in \mathbb{R}$, the noise rate.*

- $\mathsf{TRingKeyGen}()$*: Choose a secret key $\mathbf{s} \leftarrow \mathbb{Z}_2^n$. The public key is $(a_1, a_2, \mathbf{b})$, where $a_1, a_2 \leftarrow R$, $\mathbf{e} \leftarrow \mathsf{Ber}_\tau^{2n}$, and $\mathbf{b} = \mathbf{As} + \mathbf{e}$ for $\mathbf{A} = \mathsf{mat}(a_1)//\mathsf{mat}(a_2)$.*
- $\mathsf{TRingEnc}(pk = (a_1, a_2, \mathbf{b}), v)$*: To encrypt message bit $v \in \mathbb{Z}_2$, choose $f_1, f_2 \leftarrow \mathsf{Ber}_\tau^R$, define $\mathbf{f} = \mathsf{vec}(f_1)//\mathsf{vec}(f_2)$, and output the ciphertext $(\mathbf{u}, c)$ where $\mathbf{u} = \mathbf{f}^T\mathbf{A} = \mathsf{vec}(f_1 a_1)//\mathsf{vec}(f_2 a_2)$ and $c = \mathbf{f}^T\mathbf{b} + v$.*
- $\mathsf{TRingDec}(sk = s, (\mathbf{u}, c))$*: The decryption is $d = \mathbf{c} + \langle \mathbf{u}, \mathbf{s} \rangle$.*

Remark. It is not strictly necessary (for correctness or security) that g be irreducible. In fact, it could increase efficiency of implementation if g were not irreducible. However, there are various pitfalls to avoid when choosing a reducible g (for example, it must not have factors of very low degree), in order to maintain security (see [Hey+12] for a more detailed discussion). Thus, for simplicity, we have opted to let g be irreducible here.

The advantages of this cryptosystem over the LPN cryptosystem are that (as with the ring-LPN cryptosystem) the public key size has gone from $O(n^2)$ to $O(n)$, and the expensive matrix multiplication has been replaced by efficient ring multiplication. The advantage of the TRLPN cryptosystem over the ring-LPN cryptosystem is the smaller level of noise in the decryption, meaning that it is possible to achieve the same correctness with significantly smaller parameters.

A very similar modification to that used for the multi-bit LPN cryptosystem yields a multi-bit TRLPN cryptosystem with corresponding advantages. We omit a formal specification of the multi-bit TRLPN cryptosystem, for the sake of brevity and avoiding repetition, since it follows from a straightforward application of the same methodology as above.

The proofs of correctness and security of the basic TRLPN cryptosystem follow.

Lemma 2.30 (Correctness)**.** *For any constant $\varepsilon > 0$, it is possible to choose τ with $\tau = \Theta(\frac{1}{\sqrt{n}})$ such that the probability of correct decryption by $\mathsf{TRingDec}$ is at least $1 - \varepsilon$.*

Proof. Exactly as in the proof of Lemma 2.7. □

Lemma 2.31 (Pseudorandom public keys)**.** *Under the TRLPN assumption, the distribution of the public keys $(a_1, a_2, \mathbf{b})$ generated by $\mathsf{TRingKeyGen}$ is computationally indistinguishable from uniform over $R \times R \times \mathbb{Z}_2^{2n}$.*

Proof. The public keys generated by $\mathsf{TRingKeyGen}$ may be transformed (by setting $\mathbf{A} = \mathsf{mat}(a_1)//\mathsf{mat}(a_2)$) into samples of the form $(\mathbf{A}, \mathbf{As} + \mathbf{e})$, where $\mathbf{A}$, $\mathbf{s}$, and $\mathbf{e}$ are chosen as in Definition 2.21. Since furthermore q and τ are chosen as in the TRLPN assumption, the lemma follows. □

Lemma 2.32. *Under the TRLPN assumption, for any $l \in \mathbb{N}$, it holds that $(\mathbf{R}, \mathbf{f}^T\mathbf{R})$ is computationally indistinguishable from $(\mathbf{R}, \mathbf{r})$, where $\mathbf{f} \leftarrow \mathsf{Ber}_\tau^{2ln}$, $\mathbf{R} \leftarrow \Psi^{R,2l}$, and $\mathbf{r} \leftarrow \mathbb{Z}_2^{ln}$.*

Proof. Take a sample $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ constructed as detailed in Definition 2.21, with $q = 2l$. Choose $\mathbf{H} \in \mathbb{Z}_2^{ln \times n}$ by taking uniformly random column vectors from the orthogonal complement of the columns of $\mathbf{A}$. Note that $\mathbf{H} \sim \Psi_0^{R,l}$. By construction, $\mathbf{A}^T\mathbf{H} = 0$, so $\mathbf{A}^T(\mathbf{Hs} + \mathbf{e}) = \mathbf{A}^T\mathbf{e}$. Since, under the TRLPN assumption, $(\mathbf{H}, \mathbf{Hs} + \mathbf{e})$ is indistinguishable from $(\mathbf{H}, \mathbf{r'})$ for random $\mathbf{r'} \leftarrow \mathbb{Z}_2^{2ln}$, it follows that samples $(\mathbf{A}, \mathbf{A}^T\mathbf{e})$ are indistinguishable from $(\mathbf{A}, \mathbf{r})$ with $\mathbf{r} \leftarrow \mathbb{Z}_2^{ln}$. Transposing the last component gives the result. □

Theorem 2.33 (Security)**.** *Under the TRLPN assumption, the TRLPN cryptosystem is secure against chosen plaintext attack.*

Proof. Consider an instance of the TRLPN cryptosystem with parameters n and τ, where the public key is $(a_1, a_2, \mathbf{b})$. Let $\mathbf{A} = \mathsf{mat}(a_1)//\mathsf{mat}(a_2)$, and let $\mathbf{R} \in \mathbb{Z}_2^{2n \times (n+1)}$ be as follows:

$$r_{i,j} = \begin{cases} a_{i,j} & \text{for } 1 \le i \le 2n, 1 \le j \le n \\ b_i & \text{for } 1 \le i \le 2n, j = n+1 \end{cases}$$

So $\mathbf{R}$ has the same distribution as the public key. Note that a ciphertext of the message $v \in R$ is of the form $(\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b} + v)$ as defined by $\mathsf{TRingEnc}$. The ciphertext can also be written $\mathbf{f}^T\mathbf{R} \diamond v$, where we define the $\diamond$ operation as "add v to the last entry of $\mathbf{f}^T\mathbf{R}$". We now argue about the joint distribution of public key and ciphertext as follows:

By Lemma 2.31, $(\mathbf{R}, \mathbf{f}^T\mathbf{R} \diamond v)$ is indistinguishable from $(\mathbf{S}, \mathbf{f}^T\mathbf{S} \diamond v)$ where $\mathbf{S}$ corresponds to a uniformly random element in $\Psi^{R,2} \times \mathbb{Z}_2^{2n}$. Furthermore, by Lemma 2.32, $(\mathbf{S}, \mathbf{f}^T\mathbf{S} \diamond v)$ is indistinguishable from $(\mathbf{S}, \mathbf{r} \diamond v)$ where $\mathbf{r}$ is uniformly random. Since in this last distribution, $\mathbf{r}$ is independent of v, $\mathbf{r} \diamond v$ is uniform in $\mathbb{Z}_2^{n+1}$ and in particular independent of v. The lemma follows. □

3 Known Attacks

The earliest notable attack on LPN was the Blum-Kalai-Wasserman (BKW) algorithm [BKW03], which is based on the idea that by carefully choosing small sets of vectors from a large set of samples and computing their exclusive-or, we may create "new" LPN samples where only a single coordinate is set (with a slight gain in noise). With enough "new" samples, the secret may be computed correctly with high probability, by a majority vote over the "new" samples for each bit in the secret. The BKW algorithm is estimated to have time complexity $2^{O(n/\log n)}$ for $2^{O(n/\log n)}$ samples. A subsequent variant by Lyubashevsky [Lyu05] ran in time $2^{(n/\log\log n)}$ for $n^{1+\epsilon}$ samples.

Levieil and Fouque's LF1 and LF2 algorithms [LF06] practically but not asymptotically improve upon the above, and furthermore are the first BKW-style LPN attacks with a documented implementation. They modify the final step of the BKW algorithm, where instead of solving equations over one bit as in the original version, they solve equations over $b > 1$ bits at a time, with the help of Walsh-Hadamard transforms. LF2, unlike LF1, makes use of heuristics in this final step.

More recently, Kirchner proposed a modified algorithm [Kir11] using ideas from all the prior work, that greatly decreases the number of samples needed for a successful attack. Bernstein and Lange's yet more recent (implemented) attack [BL12] targeting the Lapin authentication protocol [Hey+12] is in fact applicable to many LPN-based systems, and is a version of Kirchner's algorithm optimised for the fact that Lapin is based on ring-LPN. A slight modification, mentioned briefly in [BL12], can make the attack apply also to standard LPN; this comes at a relatively minor cost in time, and requires slightly fewer queries.

This last attack has the best performance currently known, so we use its timings to determine our parameter choices. We take the attack timings of the ring-LPN version even for the LPN cryptosystems, both because the standard LPN version is only cursorily documented in [BL12], and because we aim to take conservative parameters with a reasonable security margin against slight optimisations (we estimate that the timings will differ by a factor of less than 2^{10}).

4 Parameter Choices

The number of bit operations required for a successful run of the state-of-the-art attack, according to the analysis of [BL12], is equal to $2^{f(n,\tau,a,b,l,W,q)}$, where f is a function of the cryptosystem parameters n, τ and the algorithm parameters a, b, l, W, q, as follows:

$$f(n,\tau,a,b,l,W,q) = \sum_{w\geq 2}\binom{n}{w}\tau^w(1-\tau)^{n-w} + \log_2\Bigg(\ 12q(n^2+n) + a(q-1)n^2 + \Big((q-1)n - 2^b a\Big)\sum_{w\leq W}\left(\binom{n-ab-l}{w}w\right) + l2^l\sum_{w\leq W}\binom{n-ab-l}{w}\Bigg).$$

We enforce that the probability of incorrect decryption of a bit is 25%, in order to allow error correction using codes with a reasonably low expansion factor of about 5. In the case of the LPN and TRLPN cryptosystems, this means $\frac{1}{2} - \frac{(1-2\tau^2)^{2n+2}}{2} = 0.25$. For the ring-LPN cryptosystem, we have an upper bound of $(1+\varepsilon)\tau + 28n\tau^2$ on the decryption error rate, which holds with overwhelming probability for any constant ε: the first term, $(1+\varepsilon)\tau$, comes from a Chernoff bound on f''; and the second term, $28n\tau^2$, comes from the bound of $7\cdot(2n\tau^2)$ on the weight of $ef + f's$ which follows from Lemmas 2.23 and 2.25. Based on this, we require concretely that $1.1\tau + 28n\tau^2 = 0.25$.

Given the low values of τ that result from this, the expected weight of error vectors is very low, and therefore we consider it reasonable to set $l = 1$ and $W = 1$. (Intuitively, the attack algorithm "hopes" that in a set of W error bits, l or fewer bits will be nonzero.) The parameter q can be a small integer, and does not greatly influence performance, so we simply set it at a generous value of 20. Having determined reasonable values for W, l, and q, we find the values of a and b that minimise n for a range of security levels. (Note that a and b are subject to a few additional restrictions detailed in [BL12]; we take these restrictions into account.)

Security level (bits)	n	τ	a	b
80	9000	0.0044	7	14
112	21000	0.0029	7	14
128	29000	0.0024	8	16
196	80000	0.0015	8	16
256	145000	0.0011	7	17

Table 1: Parameters for selected security parameters for LPN and TRLPN cryptosystems

Security level (bits)	n	τ	a	b
80	150000	0.00024	18	13
112	350000	0.00016	38	13
128	500000	0.00013	59	17
196	1500000	0.000099	65	18
256	2700000	0.000057	42	20

Table 2: Parameters for selected security parameters for ring-LPN cryptosystem

We have taken slightly conservative parameters, since we only expect our parameter choices to be close to optimal[2]. The reason why n-values are much larger for Ring-LPN is that the decryption introduces more noise. Therefore, to get a reasonable error rate, we need to reduce τ, and as a result we need larger n to preserve the security level. This depends, of course, on the concrete bounds we have for the decryption error. Improving these bounds would allow smaller n-values, but we would still need significantly larger n for the Ring-LPN scheme.

[2]It may be of interest and reassurance that by using our method to find near-optimal parameters, we find attacks that are slightly better than the concrete examples given in [BL12] itself.

5 Implementation

We compare the performance of the LPN, ring-LPN and TRLPN cryptosystems for various security levels. As a benchmark, we also implemented RSA encryption and decryption operations for the same security levels. The LPN cryptosystems primarily manipulate bit matrices and bit vectors of dimension $O(n)$. The multi-bit LPN cryptosystem implementation performs operations on w plaintext bits in parallel where w, the word size, is in our case 64.

The implementation was written in C++ and made use of the libraries `gf2x`, `gmp`, and `ntl` for some mathematical operations. `gf2x` was used for the ring multiplications for the ring-LPN and TRLPN cryptosystems. The implementations and all programs used for comparison purposes were run on the same machine, with a 3.20GHz Intel Core i5 processor with 4GB of RAM and a 7.2RPM SATA hard drive. The timings given are for a single encryption or decryption operation.

	Time per encryption (ms)			Time per decryption (ms)		
Security level (bits)	80	112	128	80	112	128
Basic LPN cryptosytem	25.400	127.600	239.900	0.004	0.007	0.008
Basic TRLPN cryptosystem	1.100	2.250	3.200	”	”	”
Multi-bit LPN	25.800	128.400	241.700	0.052	0.098	0.128
Multi-bit TRLPN	1.400	3.100	4.400	”	”	”
Ring-LPN cryptosystem	13.200	29.900	42.200	3.100	6.900	9.700
RSA	0.010	0.030	0.060	0.140	0.940	2.890

Table 3: Encryption/decryption times for comparison

We took RSA modulus sizes $1024, 2048$ and 3072 for the three security levels, respectively. The RSA decryption implementation assumes that the standard Chinese remainder optimisation is used to reduce decryption to two exponentiations on half-size numbers. The RSA encryption assumes public exponent $2^{16} + 1$, a de facto standard in practice.

To get a reasonable comparison between the LPN and RSA schemes, we consider a typical application, namely for k-bit security to encrypt and decrypt a k-bit symmetric key. This can be done with one RSA operation. For LPN we need to consider that because of the 25% decryption error per bit we need to expand the plaintext by a factor of about $1/(1 - h(25\%)) \approx 5$ where h is the binary entropy function. So for instance for $k = 128$ the decryption time needed in the basic scheme is $0.008 \cdot 128 \cdot 5 = 5.12$ ms, whereas the multi-bit scheme only needs $0.128 \cdot 5 = 0.64$ ms. (The timings given in Table 3 for the multi-bit schemes assume message length ℓ equal to the security level of the instantiation.) Thus, decryption in the multi-bit scheme is slower than RSA for 80-bit security, but faster for 112-bit, and about 4.5 times faster than RSA for 128-bit security.

6 Conclusion

We have seen that, while basic LPN-based public-key encryption currently seems impractical in standard applications due to the fact that the public key or ciphertext will be very large, the multi-bit TRLPN scheme is much more practical and may even be competitive in applications where decryption time can be considered the bottleneck and 112-bit security or more is desired. The Ring-LPN scheme, on the other hand, seems less competitive than TRLPN.

7 Acknowledgements

We are grateful to Dan Bernstein and Tanja Lange for information about recent attacks on LPN, to Vinod Vaikuntanathan for prompting our detailed consideration of the ring-LPN setting, and to Vadim Lyubashevsky for suggesting the structure of the ring-LPN cryptosystem.

References

[Ale03] Michael Alekhnovich. "More on Average Case vs Approximation Complexity". In: *FOCS*. IEEE Computer Society, 2003, pp. 298–307. ISBN: 0-7695-2040-5.

[App+09] Benny Applebaum et al. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *Advances in Cryptology - CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 595–618. ISBN: 978-3-642-03355-1. DOI: 10.1007/978-3-642-03356-8_35. URL: http://dx.doi.org/10.1007/978-3-642-03356-8_35.

[BL12] Daniel J. Bernstein and Tanja Lange. "Never Trust a Bunny". In: *RFIDSec*. Ed. by Jaap-Henk Hoepman and Ingrid Verbauwhede. Vol. 7739. Lecture Notes in Computer Science. Springer, 2012, pp. 137–148. ISBN: 978-3-642-36139-5.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model". In: *J. ACM* 50.4 (2003), pp. 506–519.

[Blu+94] Avrim Blum et al. "Cryptographic Primitives Based on Hard Learning Problems". In: *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*. CRYPTO '93. London, UK, UK: Springer-Verlag, 1994, pp. 278–291. ISBN: 3-540-57766-1. URL: http://dl.acm.org/citation.cfm?id=646758.759585.

[Cas12] David Cash. Private communication. 2012.

[DMQN12] Nico Döttling, Jörn Müller-Quade, and Anderson C. A. Nascimento. "IND-CCA Secure Cryptography Based on a Variant of the LPN Problem". In: *ASIACRYPT*. Ed. by Xiaoyun Wang and Kazue Sako. Vol. 7658. Lecture Notes in Computer Science. Springer, 2012, pp. 485–503. ISBN: 978-3-642-34960-7.

[GRS08] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. "How to Encrypt with the LPN Problem". In: *ICALP (2)*. Ed. by Luca Aceto et al. Vol. 5126. Lecture Notes in Computer Science. Springer, 2008, pp. 679–690. ISBN: 978-3-540-70582-6.

[Hey+12] Stefan Heyse et al. "Lapin: An Efficient Authentication Protocol Based on Ring-LPN". In: *FSE*. Ed. by Anne Canteaut. Vol. 7549. Lecture Notes in Computer Science. Springer, 2012, pp. 346–365. ISBN: 978-3-642-34046-8.

[HB01] NicholasJ. Hopper and Manuel Blum. "Secure Human Identification Protocols". English. In: *Advances in Cryptology ASIACRYPT 2001*. Ed. by Colin Boyd. Vol. 2248. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, pp. 52–66. ISBN: 978-3-540-42987-6. DOI: 10.1007/3-540-45682-1_4. URL: http://dx.doi.org/10.1007/3-540-45682-1_4.

[JW05] Ari Juels and Stephen A. Weis. "Authenticating Pervasive Devices with Human Protocols". In: *CRYPTO*. Ed. by Victor Shoup. Vol. 3621. Lecture Notes in Computer Science. Springer, 2005, pp. 293–308. ISBN: 3-540-28114-2.

[KSS10] Jonathan Katz, Ji Sun Shin, and Adam Smith. "Parallel and Concurrent Security of the HB and HB$^+$ Protocols". In: *J. Cryptology* 23.3 (2010), pp. 402–421.

[KMP14] Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. "Simple Chosen-Ciphertext Security from Low-Noise LPN". In: *Public-Key Cryptography PKC 2014*. Ed. by Hugo Krawczyk. Vol. 8383. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, pp. 1–18. ISBN: 978-3-642-54630-3. DOI: 10.1007/978-3-642-54631-0_1. URL: http://dx.doi.org/10.1007/978-3-642-54631-0_1.

[Kir11] Paul Kirchner. *Improved Generalized Birthday Attack*. Cryptology ePrint Archive, Report 2011/377. http://eprint.iacr.org/. 2011.

[LF06] Éric Levieil and Pierre-Alain Fouque. "An Improved LPN Algorithm". In: *SCN*. Ed. by Roberto De Prisco and Moti Yung. Vol. 4116. Lecture Notes in Computer Science. Springer, 2006, pp. 348–359. ISBN: 3-540-38080-9.

[Lyu05] Vadim Lyubashevsky. "The Parity Problem in the Presence of Noise, Decoding Random Linear Codes, and the Subset Sum Problem". In: *APPROX-RANDOM*. Ed. by Chandra Chekuri et al. Vol. 3624. Lecture Notes in Computer Science. Springer, 2005, pp. 378–389. ISBN: 3-540-28239-4.

[Mic10] Daniele Micciancio. Invited talk given at PKC '10. Slides available at http://cseweb.ucsd.edu/ daniele/papers/DualitySlides.pdf. 2010.

[PVW07] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. "A Framework for Efficient and Composable Oblivious Transfer". In: *IACR Cryptology ePrint Archive* 2007 (2007), p. 348.

[Pie12] Krzysztof Pietrzak. "Cryptography from Learning Parity with Noise". In: *SOFSEM*. Ed. by Mária Bieliková et al. Vol. 7147. Lecture Notes in Computer Science. Springer, 2012, pp. 99–114. ISBN: 978-3-642-27659-0.

[Reg05] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *STOC*. Ed. by Harold N. Gabow and Ronald Fagin. ACM, 2005, pp. 84–93. ISBN: 1-58113-960-8.