Is Public-Key Encryption Based on LPN Practical?

Ivan Damgård[1] and Sunoo Park[2]

[1]Aarhus University
[2]MIT CSAIL

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 (a variant of) the ring-LPN problem, which is a more recently introduced LPN variant that makes for substantial improvement in terms of both time and space. For both 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 ring-LPN based scheme 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. 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. 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. 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 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. In Döttling, Müller-Quade, and Nascimento's recent proposal of CCA-secure LPN-based public-key cryptography [DMQN12], a somewhat similar scheme was outlined; 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 resulting scheme can be further slightly optimised by the use of all-or-nothing transforms [Riv97], which yields a small advantage in terms of practical parameters. The idea behind the proofs of security we give 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 then propose a new variant of the cryptosystem where we exploit the trick from ring-LPN of specifying a public matrix more succinctly and use the ring structure to implement encryption by just two multiplications in the ring. As a result, both the public key size and encryption time become essentially linear in the security parameter, instead of quadratic, and the method for improving the plaintext to ciphertext ratio still applies. To achieve this, we need a variant of the assumption that is different from what was proposed in [Hey+12]. We believe our assumption is not stronger but rather incomparable: we discuss this in more detail later, but we emphasize that the assumption should of course be studied more carefully before any use in practice. As far as we are aware, this the first public-key scheme based on techniques from ring-LPN.

The question we then consider is as follows: 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 cryp-

tography 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.

In Section 4, we consider 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: taking 128-bit security as an example, the public key in our most efficient scheme 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 3. 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.

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 where security of 128 bits or more is desired.

2 The Cryptosystems

2.1 LPN-Based Cryptosystems

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 $x \xleftarrow{\$} S$ 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".

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 $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^n$, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{q \times n}$, and $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^q$ 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.

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 $\mathbb{Z}_2$.

- $\mathsf{BasicLPNKeyGen}()$*: Choose a secret key $\mathbf{s} \in \mathbb{Z}_2^n$. The public key is $(\mathbf{A}, \mathbf{b})$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{2n \times n}$, $\mathbf{e} \leftarrow \mathsf{Ber}_\tau^{2n}$ is the error vector, and $\mathbf{b} = \mathbf{As} + \mathbf{e}$.*
- $\mathsf{BasicLPNEnc}(pk = (\mathbf{A}, \mathbf{b}), v)$*: 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 $c = \mathbf{f}^T\mathbf{b} + v$.*
- $\mathsf{BasicLPNDec}(sk = \mathbf{s}, (\mathbf{u}, c))$*: 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: $\Pr[X \text{ is even}] = G(1) = \frac{1}{2}(G_C(1) + G_C(-1)) = \frac{1}{2} + \frac{(1-2\tau)^n}{2}$. □

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* BasicLPNDec *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 BasicLPNDec 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* BasicLPNKeyGen *is computationally indistinguishable from uniform over $\mathbb{Z}_2^{2n\times n} \times \mathbb{Z}_2^{2n}$.*

Proof. The public keys generated by BasicLPNKeyGen 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. □

Notation. 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.

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, for any $n \in \mathbb{N}$, $(\mathbf{R}, \mathbf{f}^T\mathbf{R})$ is computationally indistinguishable from $(\mathbf{R}, \mathbf{r})$, where $\mathbf{f} \leftarrow \mathsf{Ber}_\tau^{2n}$, $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{2n\times n}$ and $\mathbf{r} \xleftarrow{\$} \mathbb{Z}^n$.*

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

By construction, $\mathbf{H}^T\mathbf{A} = 0$, so $\mathbf{H}^T(\mathbf{As}+\mathbf{e}) = \mathbf{H}^T\mathbf{As}+\mathbf{H}^T\mathbf{e} = \mathbf{H}^T\mathbf{e}$. 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. □

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 $\mathbf{R} \in \mathbb{Z}_2^{(n+1)\times(2n+2)}$ be defined as follows.

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

By Lemma 2.8, $(\mathbf{A}, \mathbf{b})$ is indistinguishable from random, and hence so is $\mathbf{R}$. A ciphertext of the message 0 is of the form $(\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b})$ as defined by $\mathsf{BasicLPNEnc}$. By Lemma 2.10, for $\mathbf{f}' \leftarrow \mathsf{Ber}_\tau^{2n+2}$, $\mathbf{r} = \mathbf{f}'^T\mathbf{R}$ is indistinguishable from random. Observe that the first n entries of $\mathbf{r}$ are distributed exactly as the entries of $\mathbf{f}^T\mathbf{A}$, and that r_{n+1} is distributed exactly as $\mathbf{f}^T\mathbf{b}$. Therefore, if $(\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b})$ were distinguishable from random, then $\mathbf{r}$ would be distinguishable from random – but this would contradict Lemma 2.10. Therefore, a ciphertext of the message 0 is indistinguishable from random.

For any ciphertext $\kappa = (\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b}+1)$ of the message 1, there is a corresponding ciphertext of the message 0, $\kappa' = (\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b})$, that differs from κ only in that one bit is flipped. Furthermore, $\Pr[\mathsf{BasicLPNEnc}(1) = \kappa] = \Pr[\mathsf{BasicLPNEnc}(0) = \kappa']$, so a ciphertext of 1 has exactly the same distribution as a ciphertext of 0, except that the final ciphertext bit is flipped. Inverting a uniformly random bit yields a uniformly random output, and we have already established that a ciphertext of the message 0 is indistinguishable from random. Hence, a ciphertext of the message 1 is also indistinguishable from random. □

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].

The modified cryptosystem is presented below.

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.

- $\mathsf{MultiLPNKeyGen}()$: *Choose a secret key* $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_2^{n\times\ell}$. *The public key is* $(\mathbf{A}, \mathbf{B})$, *where* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{2n\times n}$, $\mathbf{E} \leftarrow \mathsf{Ber}_\tau^{2n\times\ell}$, *and* $\mathbf{B} = \mathbf{AS} + \mathbf{E}$.
- $\mathsf{MultiLPNEnc}(pk = (\mathbf{A}, \mathbf{B}), \mathbf{v})$: *To encrypt message* $\mathbf{v} \in \mathbb{Z}_2^\ell$, *choose* $\mathbf{f} \leftarrow \mathsf{Ber}_\tau^{2n}$ *and output the ciphertext* $(\mathbf{u}, \mathbf{c})$ *where* $\mathbf{u} = \mathbf{f}^T\mathbf{A}$ *and* $\mathbf{c} = \mathbf{f}^T\mathbf{B} + \mathbf{v}^T$.
- $\mathsf{MultiLPNDec}(sk = \mathbf{S}, (\mathbf{u}, \mathbf{c}))$: *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 $|\mathbf{f}| \leq 3n\tau$. Note that $|\mathbf{f}| \sim \mathsf{Bin}_{2n,\tau}$. By a Chernoff bound, $\Pr[A] < 1 - \left(\frac{\sqrt{e}}{1.5\sqrt{1.5}}\right)^{2n\tau}$. Since $\left(\frac{\sqrt{e}}{1.5\sqrt{1.5}}\right)^{2n\tau}$ is exponentially small in $n\tau$, and $n\tau \to \infty$ 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 $|\mathbf{f}| = \lfloor 3n\tau \rfloor$. From Lemma 2.7, the result follows. □

Lemma 2.14. (Pseudorandom public keys) *If the LPN problem is hard, then the distribution of the public keys* $(\mathbf{A}, \mathbf{B})$ *generated by* $\mathsf{MultiLPNKeyGen}$ *is computationally indistinguishable from uniform over* $\mathbb{Z}_2^{2n\times n} \times \mathbb{Z}_2^{2n\times\ell}$.

Proof. We define hybrid distributions $H_0, \cdots, H_\ell$ over matrices $(\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_2^{2n\times n} \times \mathbb{Z}_2^{2n\times\ell}$ 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 $\mathsf{MultiLPNKeyGen}$ (for parameters n and τ). Then H_0 is exactly the distribution of the public keys generated by $\mathsf{MultiLPNKeyGen}$, and H_ℓ is completely uniform over $\mathbb{Z}_2^{2n\times n} \times \mathbb{Z}_2^{2n\times\ell}$.

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 $\mathbb{Z}_2^{2n\times n} \times \mathbb{Z}_2^{2n}$ 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 $(\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_2^{2n\times n} \times \mathbb{Z}_2^{2n\times\ell}$ 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 $\mathbf{e}_j \in \mathbb{Z}_2^{2n}$ according to Ber_τ^{2n}, 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. □

Theorem 2.15. (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 $\ell = O(n)$, where the public key is $(\mathbf{A}, \mathbf{B})$. Given the public key, one can construct a matrix $\mathbf{R} \in \mathbb{Z}_2^{2(n+\ell)\times(n+\ell)}$ with entries as follows:

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

By Lemma 2.14, $(\mathbf{A}, \mathbf{B})$ is indistinguishable from random, and hence so is $\mathbf{R}$.

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. By Lemma 2.10, for $\mathbf{f}' \sim \mathbf{Ber}_\tau^{2(n+\ell)}$, $\mathbf{r} = \mathbf{f}'^T\mathbf{R}$ is indistinguishable from random. Observe that the first n entries of $\mathbf{r}$ are distributed exactly as the entries of $\mathbf{f}^T\mathbf{A}$, and that the remaining entries of $\mathbf{r}$ are distributed exactly as those of $\mathbf{f}^T\mathbf{B}$. Hence, if $(\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{B})$ were distinguishable from random, then $\mathbf{r}$ would be distinguishable from random – but this would contradict Lemma 2.10. It follows that ciphertexts are indistinguishable from $(\mathbf{A}', \mathbf{b}'+\mathbf{v})$ where $\mathbf{A}' \xleftarrow{\$} \mathbb{Z}_2^n, \mathbf{b}' \xleftarrow{\$} \mathbb{Z}_2^\ell$ are uniformly random.

Now for any $\mathbf{v}, \mathbf{v}' \in \mathbb{Z}_2^\ell$, given any ciphertext $(\mathbf{u}, \mathbf{c})$ of the multi-bit LPN cryptosystem:

$$\Pr[(\mathbf{u}, \mathbf{c}) \text{ is an encryption of } \mathbf{v}] = \Pr[\mathbf{b}' = v + c]$$
$$\Pr[(\mathbf{u}, \mathbf{c}) \text{ is an encryption of } \mathbf{v}'] = \Pr[\mathbf{b}' = v' + c]$$

Since $\mathbf{b}'$ is uniformly random, then, for any ciphertext and any pair of possible corresponding plaintexts $\mathbf{v}, \mathbf{v}'$, it is the case that to an adversary $\mathbf{v}$ and $\mathbf{v}'$ are equally likely to be the correct decryption. Therefore, the multi-bit LPN cryptosystem is secure under chosen plaintext attack. □

Finally, to increase the security of the multi-bit LPN cryptosystem for practical purposes, we propose an additional encryption/decryption step based on all-or-nothing transforms (AONTs). Intuitively, these are invertible transforms that are difficult to invert unless (almost) all transformed bits are known. They were introduced and formalised in [Riv97; Boy99]; a formal definition suitable for our purposes is provided below.

Definition 2.16. (All-or-nothing transform)
A transformation $f : \mathbb{Z}_2^s \to \mathbb{Z}_2^{s'}$ is called an all-or-nothing transform for ℓ missing bits *if the following properties hold:*

- *f is invertible.*
- *Both f and its inverse f^{-1} are efficiently computable (i.e. in polynomial time).*
- *Given up to $s' - \ell$ bits of $f(x)$, for any x, it is computationally infeasible to determine x.*

In the multi-bit LPN cryptosystem as defined above, note that an attacker may learn one bit of plaintext by discovering just n bits of secret. By applying an AONT to the message prior to encryption, we may take advantage of the many independent n-bit secrets in parallel use in the multi-bit cryptosystem. An adversary must, in the modified system, learn almost all transmitted bits in order to gain knowledge of any bit of the plaintext message, and thus the security level of the cryptosystem is increased by a factor of roughly the message length. This

comes at the cost of a slightly larger payload to transmit, as AONTs incur a small message expansion factor, as well as the time taken to perform the AONT and inverse.

Concretely, we propose Optimal Asymmetric Encryption Padding (OAEP) [BR94] as the AONT for the multi-bit LPN cryptosystem. Originally introduced by Bellare and Rogaway for unrelated purposes, OAEP was shown by Boyko to be a provably secure AONT in [Boy99].

Definition 2.17. (Optimal Asymmetric Encryption Padding)
For parameters n and k_0, generator $G : \mathbb{Z}_2^{k_0} \to \mathbb{Z}_2^n$, and hash function $H : \mathbb{Z}_2^n \to \mathbb{Z}_2^{k_0}$, the transform $\mathsf{OAEP} : \mathbb{Z}_2^n \times \mathbb{Z}_2^{k_0} \to \mathbb{Z}_2^{n+k_0}$ is defined as follows:

$$\mathsf{OAEP}^{G,H}(x, r) = x \oplus G(r) \,||\, r \oplus H(x \oplus G(r)),$$

where $||$ denotes concatenation and $\oplus$ denotes the exclusive-or operation.

For appropriate parameter choices for the OAEP[1], it turns out that the increase in the security level of the cryptosystem allows for significantly smaller public/private key sizes, and thus the additional AONT step is found to be worthwhile with the benefits outweighing the costs.

2.2 Ring-LPN Cryptosystems

We turn presently to a different, but related assumption that makes for a substantially more efficient encryption scheme with smaller public keys.

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.

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 the vertical concatenation of $\mathbf{A}$ and $\mathbf{B}$, that is, the matrix whose rows are those of $\mathbf{A}$ followed by those of $\mathbf{B}$.

We will employ a slight variation of the definition of the ring-LPN problem, introduced in [Hey+12]. The problem may be viewed as a natural variant of the LPN problem, with some additional structure that allows for very fast multiplication and compact representations of samples. Since the standard LPN problem 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 by designers (e.g. the Lapin authentication scheme of Heyse *et al.* [Hey+12]) and cryptanalysts (e.g. Bernstein and Lange's attack against ring-LPN, inspired by the Lapin proposal [BL12]).

Informally, the ring-LPN problem is defined by a ring $R = \mathbb{F}_2[X]/(g)$ with $g \in \mathbb{F}_2[X]$ of degree n, and $\tau \in \mathbb{R}$ with $0 < \tau < 0.5$. The adversary gets to see samples of form $(r, rs + e)$ where $s \in R$ is (a random) secret, $r \in R$ is random and e is chosen according to Ber_τ^R. The assumption is that the adversary cannot distinguish such samples from random pairs of ring elements.

In this paper, we will not assume that ring-LPN is hard, but instead require a different, related assumption. Just as does ring-LPN, the alternative assumption allows us to exploit the correspondence between vector by matric products and multiplication in the ring.

[1]Note, in particular, that the security level k_0 of the OAEP must be at least equal to the number of bits of security we hope to gain for the cryptosystem.

Notation. 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 $0 \leq i < n$, 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 $\mathsf{vec}(r \cdot X^i)$.

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 behind the new 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 specify the assumption and cryptosystem more precisely.

Definition 2.18. *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.*

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 $r \xleftarrow{\$} R$. 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 2.19. (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.[2] 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: hence the name for our assumption and cryptosystem.

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.

We define the ring-LPN cryptosystem as follows.

Definition 2.20. (Ring-LPN 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.*

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

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

A very similar modification to the multi-bit LPN cryptosystem yields a multi-bit ring-LPN cryptosystem with corresponding advantages. We omit a formal specification of the multi-bit ring-LPN 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 ring-LPN cryptosystem follow.

Lemma 2.21. (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* $\mathsf{RingLPNDec}$ *is at least* $1 - \varepsilon$.

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

Lemma 2.22. (Pseudorandom public keys) *Under the TRLPN assumption, the distribution of the public keys* $(a_1, a_2, \mathbf{b})$ *generated by* $\mathsf{RingLPNKeyGen}$ *is computationally indistinguishable from uniform over* $R \times R \times \mathbb{Z}_2^{2n}$.

Proof. The public keys generated by $\mathsf{RingLPNKeyGen}$ may be transformed (by setting $\mathbf{A} = \mathsf{mat}(a_1)//\mathsf{mat}(a_2)$) into samples of the form $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$, where $\mathbf{A}$, $\mathbf{s}$, and $\mathbf{e}$ are chosen as in Definition 2.19. Since furthermore q and τ are chosen as in the TRLPN assumption, the required indistinguishability follows immediately. □

Lemma 2.23. *Under the TRLPN assumption,* $(\mathbf{R}, \mathbf{f}^T\mathbf{R})$ *is computationally indistinguishable from* $(\mathbf{R}, \mathbf{r})$, *where* $\mathbf{f} \leftarrow \mathsf{Ber}_\tau^{2ln}$, $\mathbf{R} \leftarrow \Psi^{R,2l}$, *and* $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^{ln}$.

Proof. Take a sample $(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e})$ constructed as detailed in Definition 2.19, 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 $\mathbf{A}^T(\mathbf{H}\mathbf{s} + \mathbf{e}) = \mathbf{A}^T\mathbf{e}$. Since, under DRLPN, $(\mathbf{H}, \mathbf{H}\mathbf{s} + \mathbf{e})$ is indistinguishable from $(\mathbf{H}, \mathbf{r}')$ for random $\mathbf{r}' \xleftarrow{\$} \mathbb{Z}_2^{2ln}$, it follows that samples $(\mathbf{A}, \mathbf{A}^T\mathbf{e})$ are indistinguishable from $(\mathbf{A}, \mathbf{r})$ with $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^{ln}$. Transposing the last component gives the result. □

Theorem 2.24. (Security) *Under the TRLPN 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 $(a_1, a_2, \mathbf{b})$. Let $\mathbf{A} = \mathsf{mat}(a_1)//\mathsf{mat}(a_2)$, and let $\mathbf{R} \in \mathbb{Z}_2^{(n+1)\times(2n+2)}$ be as follows:

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

By Lemma 2.22, $(\mathbf{A}, \mathbf{b})$ is indistinguishable from $(\mathbf{A}, \mathbf{r})$ where $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^{2n}$. Therefore, the distribution of $\mathbf{R}$ is indistinguishable from the distribution $\widetilde{\Psi}$ sampled by taking $\mathbf{A} \leftarrow \Psi^{R,2}$ and appending two uniformly random rows and one uniformly random column.

Take some $\mathbf{f}' \leftarrow \mathsf{Ber}_\tau^{2n+2}$ and let $\mathbf{f}'' \in \mathbb{Z}_2^{2n}$ be the vector identical to $\mathbf{f}'$ in the first $2n$ entries. By Lemma 2.22, the distribution of $(\mathbf{R}, \mathbf{f}'^T\mathbf{R})$ is indistinguishable from the distribution of $(\mathbf{R}', \mathbf{f}'^T\mathbf{R}')$ for $\mathbf{R}' \leftarrow \widetilde{\Psi}$. By Lemma 2.23, $(\mathbf{A}, \mathbf{f}''^T\mathbf{A})$ is indistinguishable from $(\mathbf{A}, \mathbf{r}')$ where $\mathbf{r}' \xleftarrow{\$} \mathbb{Z}_2^n$. Since $\mathbf{f}'^T\mathbf{R}'$ is simply $\mathbf{f}''^T\mathbf{A}$ with some random bits appended, it follows that $(\mathbf{R}, \mathbf{f}'^T\mathbf{R})$ is indistinguishable from $(\mathbf{R}, \mathbf{r}')$ for $\mathbf{r}'$ random.

A ciphertext of the message 0 is of the form $(\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b})$ as defined by $\mathsf{RingLPNEnc}$. Observe that the first n entries of $\mathbf{f}'^T\mathbf{R}$ are distributed exactly as the entries of $\mathbf{f}^T\mathbf{A}$, and that $(n+1)^{th}$ entry is distributed exactly as $\mathbf{f}^T\mathbf{b}$. Therefore, if $(\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b})$ were distinguishable from random, then $\mathbf{r}'$ would be distinguishable from random – this is a contradiction. Therefore, a ciphertext of the message 0 is indistinguishable from random.

For any ciphertext $\kappa = (\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b}+1)$ of the message 1, there is a corresponding ciphertext of the message 0, $\kappa' = (\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b})$, that differs from κ only in that one bit is flipped. Furthermore, $\Pr[\mathsf{RingLPNEnc}(1) = \kappa] = \Pr[\mathsf{RingLPNEnc}(0) = \kappa']$, so a ciphertext of 1 has exactly the same distribution as a ciphertext of 0, except that the final ciphertext bit is flipped. Inverting a uniformly random bit yields a uniformly random output, and we have already established that a ciphertext of the message 0 is indistinguishable from random. Hence, a ciphertext of the message 1 is also indistinguishable from random. □

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 algorithm by Lyubashevsky [Lyu05] had time complexity $2^{(n/\log\log n)}$ 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. Their modification is to 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 (to $O(n)$) the number of samples needed for a successful attack. Kircher's insight was to convert a standard LPN problem to an easier one, in the following way. Given q LPN samples $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s}\rangle + \mathbf{e}_i)$ for $i \in \{1, \ldots, q\}$, we choose a set T containing n elements

each in $\{1, \ldots, q\}$, and assemble the vectors $\mathbf{a}_t$ for $t \in T$ into the rows of an $n \times n$ matrix $\mathbf{A}_T$. Let $\widetilde{\mathbf{e}}_T$ denote the vector whose entries are the bits $b_t = \langle \mathbf{a}_t, \mathbf{s} \rangle + \mathbf{e}_t$ for $t \in T$.

Suppose we choose some random set S and hope that $\mathbf{A}_S$ is invertible. This happens with about 30% probability; as long as there are enough samples, we may re-sample S until invertibility holds. Then for any $T \neq S$ we compute the following:

$$(\mathbf{A}_T \mathbf{A}_S^{-1}, (\mathbf{A}_T \mathbf{A}_S^{-1})(\mathbf{A}_S \mathbf{s} + \widetilde{\mathbf{e}}_S) + (\mathbf{A}_T \mathbf{s} + \widetilde{\mathbf{e}}_T)) = (\mathbf{A}_T \mathbf{A}_S^{-1}, (\mathbf{A}_T \mathbf{A}_S^{-1})\widetilde{\mathbf{e}}_S + \widetilde{\mathbf{e}}_T).$$

Notice that such newly computed pairs will have exactly the distribution of LPN samples with secret $\mathbf{e}_j$; and if $\mathbf{e}_j$ is discovered, then $\mathbf{s}$ can be deduced. We have effectively converted the LPN oracle for secret $\mathbf{s}$ into one for secret $\mathbf{e}_j$ – and the new problem is much easier to solve if $\mathbf{e}_j$ has low weight, as in many cryptographic constructions.

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, and therefore we shall 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 to the algorithm (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:

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

We choose that the probability of incorrect decryption of a bit should be 25%, in order to allow for error correction using codes with a reasonably low expansion factor of about 5; that is, we impose that $\frac{1}{2} - \frac{(1-2\tau^2)^{2n+2}}{2} = 0.25$. Given the low values of τ relative to n that result from this, the expected weight of error vectors is very low, and therefore we consider it reasonable to set $l = 1$ and $W = 1$ for a good attack. (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

We have taken slightly conservative parameters, since we only expect our parameter choices to be close to optimal[3]. For the same reason, we conservatively omitted from our calculation the several extra bits of security gained for the multi-bit cryptosystem in the case that all-or-nothing transforms are used. We expect that in total, the conservative measures taken give our parameter estimates a safety margin of 7 to 14 bits of security.

5 Implementation

We compare the performance of the LPN and ring-LPN cryptosystems and RSA in implementation, for various security levels. The LPN cryptosystems primarily involve manipulation of 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. 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 in the table below are all for a single encryption or decryption operation.

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

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, one may 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 $0.128 \cdot 2 \cdot 5 = 1.28$ ms. So we see that decryption in the multi-bit scheme is slower than RSA for 80-bit security, break-even for 112-bit, and about 3 times faster than RSA for 128-bit security.

[3]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.

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 ring-LPN based schemes are 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.

7 Acknowledgements

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

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.

[BR94] Mihir Bellare and Phillip Rogaway. "Optimal Asymmetric Encryption". In: *EUROCRYPT*. Ed. by Alfredo De Santis. Vol. 950. Lecture Notes in Computer Science. Springer, 1994, pp. 92–111. ISBN: 3-540-60176-7.

[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.

[Boy99] Victor Boyko. "On the Security Properties of OAEP as an All-or-Nothing Transform". In: *CRYPTO*. Ed. by Michael J. Wiener. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 503–518. ISBN: 3-540-66347-9.

[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.

[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.

[Riv97] Ronald L. Rivest. "All-or-Nothing Encryption and the Package Transform". In: *FSE*. Ed. by Eli Biham. Vol. 1267. Lecture Notes in Computer Science. Springer, 1997, pp. 210–218. ISBN: 3-540-63247-6.