Is Public-Key Encryption Based on LPN Practical?

Ivan Damgård* Sunoo Park†

Abstract

We conduct a practically oriented study of the cryptosystem suggested by Alekhnovich based on the Learning Parity with Noise (LPN) problem. We consider several improvements to the scheme, inspired by similar existing variants of Regev's LWE-based cryptosystem. Our conclusion is that LPN-based public-key cryptography indeed seems practical. Based on known attacks on LPN, we found that for 80-bit security, while making very conservative choices of parameters for LPN, the transmission of a key for a symmetric cryptosystem is somewhat slower than for RSA, but not prohibitive for practical use.

1 Introduction

The decisional LPN problem is that of distinguishing from random a set of samples, each of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle \oplus e)$, 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 of the problem is given in Section 2. 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.

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 the Blum-Kalai-Wasserman (BKW) algorithm [1]. The latter was originally estimated (by Blum, et al.) 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 [2]. An alternative variant of BKW is the algorithm LF1, proposed by Levieil and Fouque [3], which has similar asymptotic complexity but performs much better than BKW in practice. For $O(n)$ samples, only exponential time algorithms are known.

*Department of Computer Science, Aarhus University, Aarhus, Denmark.
†Computer Laboratory, University of Cambridge, Cambridge, UK.

This state of affairs makes LPN very attractive for cryptographic applications and indeed many applications of the "symmetric crypto" type have been suggested [4, 5]. Doing public-key cryptography based on LPN seems to be much harder; however, in [6], Alekhnovich suggested a public-key cryptosystem based on a variant of decisional LPN, where τ is not constant as in standard LPN but decreases with increasing n – in fact, $\tau \approx 1/\sqrt{n}$. This problem might be easier than LPN with constant τ, but no separation between the problems in the sense of asymptotic complexity is known.

In this paper we study a variant of Alekhnovich's original cryptosystem, which is favourable for analysis as well as practical efficiency reasons. A basic version of this scheme was first communicated to us by Cash [7], and seems to be folklore, at least in some parts of the community, but we were not able to find any published record of it. It is similar in structure to Regev's cryptosystem based on the hardness of the Learning With Errors (LWE) problem [8]. We are therefore able to improve it to get a better plaintext to ciphertext size ratio in a way similar to a corresponding improvement by Peikert *et al.* of Regev's scheme [9]. The idea behind the proofs of security we give can be traced to an invited talk given by Micciancio [10] (although we warn the reader that the talk was primarily about encryption based on LWE).

The question we ask ourselves in this paper is as follows: given what we know about LPN, is public-key cryptography based on LPN an attractive alternative to more well known cryptosystems *in practice*?

It seems that the general perception among cryptographers has been that the answer must clearly be no: Alekhnovich's version of the LPN problem seems to be easier than standard LPN (due to the limitation on the noise rate) so to ensure security would probably 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. Furthermore, what we know about algorithms for solving LPN strongly suggests that the problem gets harder as the number of samples decreases. For the LPN-based public-key cryptosystem, the adversary only gets $O(n)$ samples, which is the hardest case, so this suggests that relatively small values of n might be sufficient for security.

Indeed, as we shall see, applying the LPN cryptosystem in practice does not seem at all out of the question, when comparing to existing, widespread public-key schemes such as RSA. The preliminary implementation results we have obtained for 80-bit security suggest that a typical application such as transmitting a key for a symmetric cryptosystem can be done based on LPN using a computational effort that is somewhat larger than what we need for RSA, but not prohibitive for a practical application. While the public key is much larger than for RSA (about 150KB), its size is certainly not prohibitive considering the amounts of data that modern applications have to transport routinely.

2 The Cryptosystem

We begin by establishing some notational conventions that shall be used in this paper, and providing a more formal definition of the LPN problem.

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. Where it is clear from the context, we shall 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

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

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

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

Lemma 2.2. (Lemma 1 from [11]) *If there exists a distinguisher D that (q, t, ε)-solves the decisional $\mathsf{LPN}_{n,\tau}$ problem, then there exists 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 hardness assumptions used in this paper are based on the decisional LPN problem; henceforth, the term "LPN problem" shall refer to the decisional variant.

As usual, 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. Using this, we can state the computational assumption we will base the cryptosystem on:

Definition 2.3. (Decisional LPN assumption, DLPN)
Assume probabilistic algorithm D (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 $O(n)$. Then ε is negligible as a function of n.

Adopting the standard notion of computational indistinguishability, this can be equivalently stated as the assumption that $(\mathbf{A}, \mathbf{As} \oplus \mathbf{e})$ is computationally indstinguishable from $(\mathbf{A}, \mathbf{r})$ with the choices of τ and q we made above.

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 $\mathbf{s} \in \mathbb{Z}_2^n$ uniformly at random. For the public key, choose a matrix $\mathbf{A} \in \mathbb{Z}_2^{(2n+2)\times n}$ uniformly at random, and choose an error vector $\mathbf{e} \in \mathbb{Z}_2^{2n+2}$ according to Ber_τ^{2n+2}. Then the public key is the pair $(\mathbf{A}, \mathbf{b})$, where $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$.*
- $\mathsf{BasicLPNEnc}(pk = (\mathbf{A}, \mathbf{b}), v)$*: To encrypt a message bit $v \in \mathbb{Z}_2$, choose a vector $\mathbf{f} \in \mathbb{Z}_2^{2n+2}$ according to Ber_τ^{2n+2}. Then the ciphertext is the pair $(\mathbf{u}, c)$ 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 $d = c + \langle \mathbf{u}, \mathbf{s} \rangle$.*

Remark. In [12], Döttling, Müller-Quade, and Nascimento present an LPN-based CPA-secure public key encryption scheme similar to the above. The difference is that their encryption process outputs a pair of the form $(\mathbf{u} + \mathbf{e}_1, c + e_2)$, where $\mathbf{u}$ and c are defined as above, and $\mathbf{e}_1 \sim \mathsf{Ber}_\tau^n$ and $e_2 \sim \mathsf{Ber}_\tau$ are chosen during encryption. This yields a higher decryption error than for the scheme presented in this paper and so, considering that the hardness assumption upon which the two cryptosystems are based is the same (that of the LPN problem), the scheme presented here seems to compare favourably.

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

$$G_X(z) = \sum_{k=0}^{n} z^k \Pr[X = k] = ((1-\tau) + \tau z)^n.$$

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 $C_i = e_i \cdot f_i$. These C_i are independent and identically distributed with the distribution Ber_{τ^2}. Let $C = \sum_i C_i$. Then $C \sim \mathsf{Bin}_{2n+2,\tau^2}$.

Observe that $\mathbf{f}^T\mathbf{e} = 0$ if and only if C takes an even value. From Lemma 2.5, then, $\Pr[\mathbf{f}^T\mathbf{e} = 0] = \frac{1}{2} + \frac{(1-2\tau^2)^{2n+2}}{2}$. Take $0 < \tau \le O(\frac{1}{\sqrt{n}})$: 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:

$$\lim_{n\to\infty}(1-2\tau^2)^{2n+2} = \lim_{n\to\infty}(1-\frac{2\tau^2(2n+2)}{2n+2})^{2n+2} = e^{-2\tau^2(2n+2)}.$$

Hence, for large n,

$$\Pr[\mathbf{f}^T\mathbf{e} = 0] = \frac{1}{2} + \frac{(1-2\tau^2)^{2n+2}}{2} \approx \frac{1+e^{-2\tau^2(2n+2)}}{2}.$$

If $\tau = \frac{c}{\sqrt{n}}$ for some constant c, then the exponent $-2\tau^2(2n+2)$ of the above equation is constant. Observe that $\lim_{c\to 0} -2\tau^2(2n+2) = 0$, so $\lim_{c\to 0} \frac{1+e^{-2\tau^2(2n+2)}}{2} = 1$. 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. In practice, it may be desirable to use error-correcting codes to increase tolerance for occasional incorrectly decrypted bits.

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 $\mathbb{Z}_2^{(2n+2)\times n} \times \mathbb{Z}_2^{2n+2}$.*

Proof. The public keys generated by BasicLPNKeyGen are of the form $(\mathbf{A}, \mathbf{A}\mathbf{s}\oplus\mathbf{e})$, 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 $m = O(n)$ with $m \geq n$, 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 computationally indistinguishable from the uniform distribution over $\mathbb{Z}_2^{m \times n}$.*

Proof. We refer to the result of Kahn, Komlós, and Szemerédi in [13]: that the probability, p_k, that a uniformly random square matrix $\mathbf{M}_k \in \mathbb{Z}_2^{k \times k}$ is singular is exponentially small in k, and in fact $p_k < 0.999^k$. Let $\mathbf{M}_m^{(n)}$ denote the matrix formed by the first n columns of $\mathbf{M}_m$. For any $m \geq n$, $\mathbf{M}_m^{(n)}$ is uniformly random in $\mathbb{Z}_2^{m \times n}$, and furthermore if $\mathbf{M}_m$ is nonsingular then $\mathbf{M}_m^{(n)}$ has linearly independent columns. Thus,

$$\Pr[\mathbf{M}_m^{(n)} \text{ has full rank}] \geq \Pr[\mathbf{M}_m \text{ nonsingular}] \geq 1 - 0.999^m = 1 - 0.999^{O(n)}.$$

Therefore, $\chi_{m,n}$ is computationally indistinguishable from uniform over $\mathbb{Z}_2^{m \times n}$ with polynomial-time sampling, for large n. □

Lemma 2.10. *Under the DLPN assumption, for any $c \in \mathbb{N}$, $(\mathbf{R}, \mathbf{f}^T\mathbf{R})$ is computationally indistinguishable from $(\mathbf{R}, \mathbf{r})$, where $\mathbf{f} \in \mathbb{Z}_2^{2(n+c)}$ is drawn from $\mathsf{Ber}_\tau^{2(n+c)}$, $\mathbf{R} \in_R \mathbb{Z}_2^{2(n+c) \times (n+2c)}$ and $\mathbf{r} \in_R \mathbb{Z}^{n+2c}$.*

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

By construction, $\mathbf{H}^T\mathbf{A} = 0$, so $\mathbf{H}^T(\mathbf{As} \oplus \mathbf{e}) = \mathbf{H}^T\mathbf{As} \oplus \mathbf{H}^T\mathbf{e} = \mathbf{H}^T\mathbf{e}$. This means that since, under DLPN, $(\mathbf{A}, \mathbf{As} \oplus \mathbf{e})$ is indistinguishable from random, so is $(\mathbf{H}, \mathbf{H}^T\mathbf{e})$. The lemma then finally follows from noting that indistinguishability from random still holds if we transpose the last component. □

Theorem 2.11. *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 $(\mathbf{A}, \mathbf{b})$. Given the public key, one can construct a matrix $\mathbf{R} \in \mathbb{Z}_2^{(n+1) \times (2n+2)}$ with entries as follows:

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

By Lemma 2.8, $(\mathbf{A}, \mathbf{b})$ is indistinguishable from random. It follows that $\mathbf{R}$ is also indistinguishable from random.

A ciphertext of the message 0 is of the form $(\mathbf{f}^T\mathbf{A}, \mathbf{f}^T\mathbf{b})$ as defined by LPNEnc. By Lemma 2.10, $\mathbf{r} = \mathbf{f}^T\mathbf{R}$ is indistinguishable from random. Observe that the first n entries of $\mathbf{r}$ are exactly the entries of $\mathbf{f}^T\mathbf{A}$, and that $r_{n+1} = \mathbf{f}^T\mathbf{b}$. 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, $\Pr[\mathsf{LPNEnc}(1) = \kappa] = \Pr[\mathsf{LPNEnc}(0) = \kappa']$, 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.

Therefore, ciphertexts produced by LPNEnc are 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 [9]. The modified cryptosystem is presented below.

Definition 2.12. (Improved 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. All operations are, as before, performed over $\mathbb{Z}_2$.

- $\mathsf{LPNKeyGen}()$*: Choose a secret key $\mathbf{S} \in \mathbb{Z}_2^{n\times\ell}$ uniformly at random. For the public key, choose a matrix $\mathbf{A} \in \mathbb{Z}_2^{2(n+\ell)\times n}$ uniformly at random, and choose an error matrix $\mathbf{E} \in \mathbb{Z}_2^{2(n+\ell)\times\ell}$ according to $\mathsf{Ber}_\tau^{2(n+\ell)\times\ell}$. Then the public key is the pair $(\mathbf{A}, \mathbf{B})$, where $\mathbf{B} = \mathbf{AS} + \mathbf{E}$.*
- $\mathsf{LPNEnc}(pk = (\mathbf{A}, \mathbf{B}), \mathbf{v})$*: To encrypt a message $\mathbf{v} \in \mathbb{Z}_2^\ell$, choose a vector $\mathbf{f} \in \mathbb{Z}_2^{2(n+\ell)}$ according to $\mathsf{Ber}_\tau^{2(n+\ell)}$. Then the ciphertext is the pair $(\mathbf{u}, \mathbf{c})$ where $\mathbf{u} = \mathbf{f}^T\mathbf{A}$ and $\mathbf{c} = \mathbf{f}^T\mathbf{B} + \mathbf{v}^T$.*
- $\mathsf{LPNDec}(sk = \mathbf{S}, (\mathbf{u}, \mathbf{c}))$*: 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 improved cryptosystem.

Lemma 2.13. (CORRECTNESS) *For each fixed choice of* $\mathbf{f}$*, encryption followed by decryption of a message* $\mathbf{v}$ *in the improved 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 $\mathbf{f}^T\mathbf{E} = \mathbf{0}$ (where $\mathbf{0}$ denotes the zero vector 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 $\Pr[\mathbf{f}^T\mathbf{e}_i = 0] = \frac{1}{2} + \frac{(1-2\tau)^{|\mathbf{f}|}}{2}$. Observe that this probability increases as $|\mathbf{f}|$ decreases.

Let A denote the event that $|\mathbf{f}| \leq 3(n+\ell)\tau$. $|\mathbf{f}|$ is distributed according to $\mathsf{Bin}_{2(n+\ell),\tau}$. By a Chernoff bound,

$$\Pr[A] < 1 - \left(\frac{\sqrt{e}}{1.5\sqrt{1.5}}\right)^{2(n+\ell)\tau}.$$

Note that $\left(\frac{\sqrt{e}}{1.5\sqrt{1.5}}\right)^{2(n+\ell)\tau}$ is exponentially small in $(n+\ell)\tau$, and that $(n+\ell)\tau \to \infty$ as $n \to \infty$, so A occurs with overwhelming probability.

If A occurs, then the encryption followed by decryption of ℓ bits in the improved 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 $|\mathbf{f}| = \lfloor 3(n+\ell)\tau \rfloor$. From Lemma 2.7, it follows that for any constant $\varepsilon > 0$, τ can be chosen subject to $\tau = \Theta(\frac{1}{\sqrt{n}})$ such that $\rho \leq \varepsilon$. □

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* $\mathsf{LPNKeyGen}$ *is computationally indistinguishable from uniform over* $\mathbb{Z}_2^{2(n+\ell)\times n} \times \mathbb{Z}_2^{2(n+\ell)\times \ell}$*.*

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

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 $\mathbb{Z}_2^{2(n+\ell)\times n} \times \mathbb{Z}_2^{2(n+\ell)}$ that are either chosen uniformly at random, or are of the form $(\mathbf{A}, \mathbf{As} \oplus \mathbf{e})$ as specified in the LPN problem (in Definition 2.1). $\mathcal{S}_k$

outputs a pair $(\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_2^{2(n+\ell)\times n} \times \mathbb{Z}_2^{2(n+\ell)\times \ell}$ constructed as follows. First, $\mathcal{O}$ is queried, yielding a sample $(\mathbf{A}', \mathbf{b}')$. Then, $\mathcal{S}_k$ sets the output matrix $\mathbf{A}$ 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 $\mathbf{e}_j \in \mathbb{Z}_2^{2(n+\ell)}$ according to $\mathsf{Ber}_\tau^{2(n+\ell)}$, and sets $\mathbf{b}_j = \mathbf{A}\mathbf{s}_j + \mathbf{e}_j$.

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. *If the LPN problem is hard, then the improved LPN cryptosystem is secure against chosen plaintext attack.*

Proof. Consider an instance of the improved 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:

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

By Lemma 2.14, $(\mathbf{A}, \mathbf{B})$ is indistinguishable from random. Therefore, $\mathbf{R}$ is also indistinguishable from random.

A ciphertext in the improved 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, $\mathbf{r} = \mathbf{f}^T\mathbf{R}$ is indistinguishable from random. Observe that the first n entries of $\mathbf{r}$ are exactly the entries of $\mathbf{f}^T\mathbf{A}$, and that the remaining entries of $\mathbf{r}$ are exactly 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 $\mathbf{A}' \in_R \mathbb{Z}_2^n, \mathbf{b}' \in_R \mathbb{Z}_2^\ell$ are uniformly random.

Now for any $\mathbf{v}, \mathbf{v}' \in \mathbb{Z}_2^\ell$, given any ciphertext $(\mathbf{u}, \mathbf{c})$ of the improved 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 improved LPN cryptosystem is secure under chosen plaintext attack. □

3 Parameters

We seek to estimate concrete parameter sizes sufficient for a secure implementation of the LPN cryptosystem against the best currently known attacks. We note that the existing algorithms for attacking LPN (discussed previously in Section 1) do not target the particular challenges of attacking this sort of cryptosystem: they address the search rather than the decisional variant of the problem; they do not consider separately the case where the noise rate is limited to a maximum of $\frac{1}{\sqrt{n}}$; and they require exponentially many samples, while in the case of the cryptosystem only $O(n)$ samples are available. We aim to take these discrepancies into consideration in forming a conservative estimate.

Given the absence of known algorithms targeting the decisional LPN problem, we consider it reasonable to base our parameter estimates on the existing attacks against search LPN, for the reason that the two variants of the problem are "polynomially equivalent" as stated in Lemma 2.2, and the numbers of samples required by the existing algorithms are far greater than polynomial relative to the number of samples made available in the cryptosystem.

Our estimate is based on the Levieil and Fouque's data on the performance of the LF1 algorithm. We deemed this appropriate for a number of reasons. First, LF1 performs better than BKW and the number of samples required is the same for both: $2^{O(n/\log n)}$. Furthermore, Lyubashevsky's algorithm, which requires only $n^{1+\varepsilon}$ samples, makes use of BKW in the final step in such a way that its performance for $n^{1+\varepsilon}$ samples cannot be better than that of the BKW algorithm (and therefore also LF1) for $2^{O(n/\log n)}$ samples. Finally, and importantly for practical considerations, LF1 is the only one of these attacks that has thus far been implemented, and therefore the only one for which there are concrete as well as asymptotic performance measures available.

We seek appropriate parameters to compare our implementation with 1024-bit RSA, which is considered 80-bit secure. Data are given only for certain pairs (n, τ) in [3], so we were not able to obtain concrete values for exactly 80-bit. Rather than extrapolating for those exact security levels, we have taken from the included parameter values some values that yield slightly better than the required security in each case, with $\tau \approx \frac{1}{\sqrt{n}}$.

We estimate $n = 768$ and $\tau = 0.015$ to be sufficient for 80-bit security against LF1, and furthermore note that these parameter values provide more than 80-bit security against exhaustive search over low-weight error vectors. Taking these parameter values, the decryption error probability for the basic LPN cryptosystem is 25%. We run our implementation with these values of n and τ for a conservative comparison against 1024-bit RSA, of which details shall be given in Section 4.

4 Implementation

Our implementation is of the basic LPN cryptosystem. An implementation of this cryptosystem primarily manipulates bit matrices and bit vectors of dimension $O(n)$. We implement a library performing up to w operations in parallel where w, the word size, is in our case equal to 64. Encryption, then, may be done for up to w message bits at

no more cost than for a single message bit (for any given key pair); and random and Bernoulli sampling may furthermore be achieved more efficiently than in the case of encrypting one bit at a time.

Remark. We compared the performance of the above described implementation with a simple bit-slicing method, and found that the latter was less efficient.

The implementation was written in C++. For randomness we used the `rand()` function of the C standard library. The implementation and all programs used for comparison purposes were run on the same machine, with a 2.66GHz Intel Core 2 Duo processor with 4GB of RAM.

The following are preliminary timing results for our implementation of the basic LPN cryptosystem with the parameters specified in Section 3. Finally, for comparison purposes, we also present timings for 1024-bit (considered 80-bit secure), obtained using the open source toolkit OpenSSL. The benchmarking function that we used is called by the command `openssl speed rsa1024`.

80-bit security	Mean time taken (μs)	
	for encryption	for decryption
Basic LPN cryptosystem	821	4
OpenSSL RSA	580	30

Recall that in the basic cryptosystem, each encryption operation encrypts a single message bit. Accordingly, the timings in the first row of the table above are for the encryption/decryption of a single bit. With the decrpytion error probability of 0.25 that is given by the present choice of parameters, when we consider using error correcting codes to reliably transmit messages, the message expansion factor is $\frac{1}{1-h(\tau)}$ where h is the binary entropy function. In our case the expansion factor is therefore 5.3. It follows that, for example, to transmit a 128-bit symmetric key using iterations of the basic LPN cryptosystem it will be necessary to transmit 678 message bits.

We emphasize that the timings are preliminary and we will update these in upcoming versions, including also timings for the improved scheme. Nevertheless, it is apparent from the table that even if we need to send several hundred bits to get a symmetric key through, we end up spending decryption time that is slower than RSA, but far from prohibitive for practical application.

For encryption, the RSA time is for a full-scale exponentiation, and could of course be reduced by using a small public exponent. In comparison, naive usage of the basic LPN scheme is much slower. On the other hand, the encryption time for the LPN scheme can also be reduced in several ways: First, one may expect that using the improved LPN scheme will reduce the encryption time per bit sent dramatically. Second, in a practical implementation scenario, there may be idle time that can be allotted to preprocessing. In this case, since $\mathbf{f}^T\mathbf{A}$ and $\mathbf{f}^T\mathbf{b}$ (or $\mathbf{f}^T\mathbf{B}$, in the improved system) do not depend on the message to be encrypted, they could be precomputed offline. We found that more than 99% of encryption time is comprised of matrix multiplication, so such preprocessing would reduce the online encryption time very significantly. Such preprocessing can also be done for some discrete-log based schemes, but not for RSA as far as we are aware.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] David Cash. Private communication. 2012.

[8] 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.

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

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

[11] 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.

[12] Nico Döttling, Jörn Müller-Quade, and Anderson C. A. Nascimento. "IND-CCA secure Cryptography based on a variant of the LPN Problem". To appear at AsiaCrypt 2012. 2012.

[13] Jeff Kahn, János Komlós, and Endre Szemerédi. "On the Probability That a Random ± 1-Matrix Is Singular". In: *J. American Mathematical Society* 8.1 (1995), pp. 223–240.