HELEN: a Public-key Cryptosystem Based on the LPN Problem (Extended Abstract)

Alexandre Duc and Serge Vaudenay

Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

Abstract. We propose HELEN, a new code-based public-key cryptosystem whose security is based on the hardness of the Learning from Parity with Noise problem (LPN) and the decisional minimum distance problem. We show that the resulting cryptosystem achieves indistinguishability under chosen plaintext attacks (IND-CPA security). Using the Fujisaki-Okamoto generic construction, HELEN achieves IND-CCA security in the random oracle model. We further propose concrete parameters.

Keywords: Code-based cryptosystem, learning from parity with noise problem, minimum distance problem, random linear code, public-key cryptostem.

1 Introduction

In this paper, we present HELEN a new public-key cryptosystem, whose security relies on the hardness of the *Learning from Parity with Noise problem* (LPN) and the *minimum distance problem* which are both NP-hard.[1]

We encrypt a duplicated bit b by hiding it using a random codeword as well as a random biased noise vector. The bits of this mask corresponds to queries to an LPN problem with hidden vector $\boldsymbol{r}$. For decryption, the random linear codeword is removed by multiplying the ciphertext with h. The noise is removed by majority logic decoding.

Related Work. A private-key encryption scheme named LPN-C was proposed by Gilbert, Robshaw and Seurin [15]. LPN-C was shown IND-CPA secure. The construction of HELEN presents some similarities with the trapdoor cipher TCHo [3,11] by Aumasson et al. which similarly encrypts a message by adding some random biased noise and some contribution from a linear code.

More closely related cryptosystems were proposed. Gentry et al. proposed an LWE-based cryptosystem [14] in which users share a common random matrix and whose private key (resp. public key) consists in a random error vector (resp. its syndrome). Extensions to $p = 2$ have been open so far. Our procedure is different from theirs in the sense that we hide a low-parity check equation in a matrix so that this matrix looks random, whereas they pick a totally random matrix. Similarly, Alekhnovich proposed a scheme based on the Average-nearest-codeword conjecture [1]. Applebaum et al. proposed a scheme, which is very similar to ours but which uses sparse matrices instead of random ones. Thus, the security reduces to the less-studied 3LIN problem instead of LPN. This problem is similar to the LPN problem except that queries are done with vectors with exactly 3 ones instead of random vectors. Also, the authors do not provide any concrete parameters [2]. So, to the best of our knowledge, we propose for the first time a concrete PKC whose security is based on LPN.

2 Preliminaries

We denote the Hamming weight of a bitstring x by $\mathsf{wt}(x)$. We write $x \xleftarrow{U} \mathcal{D}$ if an element x is drawn uniformly at random in a domain $\mathcal{D}$. We denote the Bernoulli distribution with parameter p by $\mathrm{Ber}(p)$.We write S_p^n the sequence of n independent Bernoulli trials with parameter p. We write $\mathrm{S}_p^n(r)$ when we need to specify the seed r used to generate this sequence. Given a permutation $\sigma \in \mathfrak{S}_n$, the group of all permutations over n elements, and given $h \in \{0,1\}^n$, we write $\sigma \star h$ when we apply σ on the bits of h. Given a $k \times n$ matrix G, we write $\sigma \star G$ when we apply σ on the columns of G.

The *Learning from Parity with Noise* (LPN) problem has been well studied both in learning theory and in cryptography. The goal of this problem is to find out an unknown vector $\boldsymbol{u}$, given some noisy versions of its scalar product with some known random vector.

[1] HELEN stands for Hidden Equation for Linear Encryption with Noise.

Definition 1 (LPN Oracle). *An* LPN oracle $\Pi_{\boldsymbol{u},p}$ *for a hidden vector* $\boldsymbol{u} \in \{0,1\}^k$ *and* $0 < p < \frac{1}{2}$ *is an oracle returning an LPN vector, i.e., vectors of the form*

$$\langle \boldsymbol{a} \xleftarrow{U} \{0,1\}^k , \boldsymbol{a} \cdot \boldsymbol{u} \oplus \nu \rangle ,$$

where, $\nu \leftarrow \mathrm{Ber}(p)$.

Problem 2 (Learning from Parity with Noise Problem). The (k,p)-Learning from Parity with Noise Problem ((k,p)-LPN) consists, given an LPN Oracle $\Pi_{\boldsymbol{u},p}$, to recover the hidden vector $\boldsymbol{u}$.

We say that an algorithm $\mathcal{A}$ (t,n,δ)-solves the (k,p)-LPN problem if $\mathcal{A}$ runs in time at most t, makes at most n oracle queries and

$$\Pr\left[\boldsymbol{u} \xleftarrow{U} \{0,1\}^k : \mathcal{A}^{\Pi_{\boldsymbol{u},p}}(1^k) = \boldsymbol{u}\right] \geq \delta .$$

The LPN problem is NP-Hard [4] and no good algorithm is known for the average case.

The LPN problem has also a decisional form. The problem is the following: let U_{k+1} be an oracle returning random $k+1$-bit vectors. Then, an algorithm $\mathcal{A}$ (t,n,δ)-solves the (k,p)-*decisional LPN problem* (D-LPN) if $\mathcal{A}$ runs in time at most t, makes at most n oracle queries and

$$\left|\Pr\left[\boldsymbol{u} \xleftarrow{U} \{0,1\}^k : \mathcal{A}^{\Pi_{\boldsymbol{u},p}}(1^k) = 1\right] - \Pr\left[\mathcal{A}^{U_{k+1}}(1^k) = 1\right]\right| \geq \delta .$$

It is shown that D-LPN and LPN are equivalent [16,20].

The best algorithms solving the LPN problem are improvements of the BKW algorithm [6,18,12]. They have complexity $2^{O(k/\log k)}$.

In our security proof, we will also need to bound the complexity of finding a low-weight parity-check equation in a random linear code which is the same as finding a low-weight codeword in the dual code. This problem of finding a low-weight codeword is also called the minimum distance problem.

Problem 3 (Minimum Distance Problem). The (n,k,w)-*decisional minimum distance problem* is the following. Given an $(n-k) \times n$ matrix H and given $w \in \mathbb{N}, w \geq 0$, is there a *non-zero* $\boldsymbol{x} \in \mathbb{F}_2^n$ with $\mathsf{wt}(\boldsymbol{x}) \leq w$ such that $\boldsymbol{x}H^t = \boldsymbol{0}$?

The computation counterpart of this problem consists in finding such an x.

Many algorithms solving this problem were developed, e.g. [17,22,7,8,9,10]. A lower-bound on its complexity is proposed in [5].

3 The Cryptosystem

We show how to encrypt one single bit b. The scheme can easily be extended to multiple bits using an error-correcting code. Our message space is $\mathcal{M} = \{0,1\}$. We denote the cryptosystem by HELEN.

HELEN uses the following parameters which are described below: n, k, p, w, c, and $\mathcal{H}$. We encode first our message bit b with a binary $[n,1]$-error-correcting code C_1, for $n \in \mathbb{N}$. The goal of this code is to be able to recover b when errors occur. Let $c \in \{0,1\}^n$ be the generating matrix of this code (in fact it is a vector). We encode b as $b \cdot c$. This message is hidden by a random codeword from a random binary linear $[n,k]$-code C_2 which has a low-weight parity-check equation $h \in \{0,1\}^n$ and a generator matrix $G \in \{0,1\}^{k \times n}$. The parameter $k \in \mathbb{N}$ determines the dimension of the codeword space in C_2 and needs to be tuned so that the system has the required security. The parity-check equation h will be the *private key* of our system while G will be the *public key*. Since h is a parity check equation of the code generated by G, we have $h \cdot G^t = 0$. We denote the weight of h by w and the set of all possible h by $\mathcal{H}$. We require $\mathcal{H}$ to verify the following property: there should exist a subgroup P of $\mathfrak{S}_n$ such that for any $\sigma \in P$ and any $h \in \mathcal{H}$, $\sigma \star h \in \mathcal{H}$. The group P defines a *group action* on the set $\mathcal{H}$. We require P to be a *transitive* group action, i.e, for any two $h, h' \in \mathcal{H}$, there exists a $\sigma \in P$ such that $\sigma \star h = h'$. We require also that $h \cdot c^t = 1$ for all $h \in \mathcal{H}$. As an example, $\mathcal{H}$ can be the set of all vectors of odd weight w and dimension n and $c = (1, \ldots, 1)$.

3.1 Encryption

A bit $b \in \mathcal{M}$ is encrypted as $\mathsf{BEnc}(G, b; r_1 \| r_2) = b \cdot c \oplus r_1 G \oplus \nu$, where c is the generator vector for C_1, G is the generator matrix for C_2, $r_1 \in \{0,1\}^k$ is random and $\nu := \mathrm{S}_p^n(r_2)$, i.e., it is the n first bits generated by the source S_p with random seed r_2. The ciphertext space is, thus, $\mathcal{C} = \{0,1\}^n$. The complexity of encryption is $O(kn)$.

3.2 Decryption

We define $b' \coloneqq \mathsf{BDec}(h, y) = h \cdot y^t$. Thus, given a ciphertext $y \in \{0,1\}^n$, we recover the original message by first removing the noise due to C_2. This is done by applying h on y since $h \cdot G^t = 0$. Hence, $h \cdot y^t = (h \cdot c^t \cdot b^t) \oplus \nu'$, for $\nu' \coloneqq h \cdot \nu^t$ a noise with

$$\Pr[\nu' = 1] = \frac{1 - (1-2p)^w}{2} =: P_{\mathsf{error}} \ .$$

So $\mathsf{BDec}(j, \mathsf{BEnc}(G, b)) = b$ with probability $\frac{1+(1-2p)^w}{2}$. This implements a BSC with parameter P_{error}.

Note that it is necessary that $h \cdot c^t = 1$ for all vector $h \in \mathcal{H}$ if one wants to be able to recover b. When $\mathcal{H}$ includes all vectors of weight w, this condition is equivalent to setting c to the all-one vector and w to an odd number. Note that the complexity of decryption is $O(n)$.

3.3 Key Generation

We need now to generate a code that is indistinguishable from a random code but that contains a known secret parity-check equation h of low weight. Let w be the required weight of h and let $\mathcal{H}$ be the set of all possible private keys. We propose the following key generation scheme.

1. Draw a random vector h of length n in the set $\mathcal{H}$. This vector will be the private key.
2. Let $0 < u \leq n$ be any index of h such that $h_i = 1$, e.g., $\max\{i \colon h_i = 1\}$.
3. Let $g_{ij} \leftarrow \mathrm{Ber}(\frac{1}{2})$, for $1 \leq i \leq k$ and $1 \leq j \leq n$, $j \neq u$.
4. Let $g_{iu} = \sum_{\substack{1 \leq j \leq n \\ j \neq u}} g_{ij}$ for $1 \leq i \leq k$, where the sum is taken over $\mathbb{F}_2$.
5. Return the matrix $G \coloneqq [g_{ij}]_{\substack{1 \leq i \leq k \\ 1 \leq j \leq n}}$ and the vector h.

The resulting public key size is $k \times n$ bits, since we have to store the matrix G. The private key is $w \log n$ bits long. The key generation complexity is $O(k \times n)$.

3.4 Security

We will reduce the semantic security of our scheme to the LPN problem and the low-weight codeword problem when $\mathcal{H}$ contains all the vectors of weight w.

Theorem 4. *Let $\varepsilon_0 \coloneqq \frac{(\#\mathcal{H}-1)(\#\mathcal{H}+2)}{2^{k+1}}$. If the (n,k,w)-decisional minimum distance problem is (t_1, ε_1)-computationally unsolvable, and if the (k,p)-decisional LPN problem is (t_2, ε_2)-hard, then there exists a constant τ such that our cryptosystem is $(\min\{t_1, t_2 - \tau kn\}, 2(\varepsilon_0 + \varepsilon_1 + \varepsilon_2))$ -IND-CPA-secure .*

The scheme can easily be made IND-CCA-secure using standard techniques [13].

4 Selection of Parameters

To compare different parameters, we will normalize them with the capacity of a binary symmetric channel (BSC) with parameter P_{error}. Recall that the capacity of the BSC is $C \coloneqq 1 - H_2(P_{\mathsf{error}})$ with $H_2(p) \coloneqq -p \log(p) - (1-p)\log(1-p)$. We normalize by this factor, since we know that such a rate is achievable by the channel coding theorem. This gives us a good way of comparing the parameters.

We propose two sets of parameters. Some which minimizes the n/C ratio to minimize the number of transmitted bits and some with a smaller kn/C ratio to minimize the encryption/decryption complexity. We give in Table 1 concrete parameters for different security parameters λ. We believe that breaking our scheme requires a complexity of $O\left(2^{\lambda}\right)$.

We also computed asymptotic parameters for our system.

$$k = \Theta\left(\lambda^2\right) \qquad n = \Theta\left(\lambda^4\right) \qquad w = \Theta(\lambda) \qquad p = \Theta(1/\lambda) \ .$$

Table 1. Parameters for our cryptosystem

λ	k	n	w	p	kn	n/C	kn/C
64	4 500	18 000	33	0.01	$2^{26.3}$	$2^{16.4}$	$2^{28.6}$
64	2 200	16 000	23	0.02	$2^{25.0}$	$2^{17.1}$	$2^{28.2}$
80	5 600	28 000	35	0.01	$2^{27.2}$	$2^{17.2}$	$2^{29.7}$
80	2 800	27 000	25	0.02	$2^{26.2}$	$2^{18.1}$	$2^{29.6}$

References

1. Alekhnovich, M.: More on Average Case vs Approximation Complexity. In: FOCS. pp. 298–307. IEEE Computer Society (2003)
2. Applebaum, B., Barak, B., Wigderson, A.: Public-key cryptography from different assumptions. In: Schulman [21], pp. 171–180
3. Aumasson, J.P., Finiasz, M., Meier, W., Vaudenay, S.: TCHo: A Hardware-Oriented Trapdoor Cipher. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP. Lecture Notes in Computer Science, vol. 4586, pp. 184–199. Springer (2007)
4. Berlekamp, E.R., McEliece, R.J., Van Tilborg, H.C.A.: On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory 24(3), 384–386 (1978)
5. Bernstein, D.J., Lange, T., Peters, C.: Smaller Decoding Exponents: Ball-Collision Decoding. In: Rogaway, P. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 6841, pp. 743–760. Springer (2011)
6. Blum, A., Kalai, A., Wasserman, H.: Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model. J. ACM 50(4), 506–519 (2003)
7. Canteaut, A., Chabanne, H.: A Further Improvement of the Work Factor in an Attempt at Breaking McEliece's Cryptosystem. In: Charpin, P. (ed.) EUROCODE (1994)
8. Canteaut, A., Chabaud, F.: A New Algorithm for Finding Minimum-Weight Words in a Linear Code: Application to McEliece's Cryptosystem and to Narrow-Sense BCH Codes of Length 511. IEEE Transactions on Information Theory 44(1), 367–378 (1998)
9. Canteaut, A., Sendrier, N.: Cryptoanalysis of the Original McEliece Cryptosystem. In: Ohta, K., Pei, D. (eds.) ASIACRYPT. Lecture Notes in Computer Science, vol. 1514, pp. 187–199. Springer (1998)
10. Finiasz, M., Sendrier, N.: Security Bounds for the Design of Code-Based Cryptosystems. In: Matsui [19], pp. 88–105
11. Finiasz, M., Vaudenay, S.: When Stream Cipher Analysis Meets Public-Key Cryptography. In: Biham, E., Youssef, A.M. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 4356, pp. 266–284. Springer (2006)
12. Fossorier, M.P.C., Mihaljevic, M.J., Imai, H., Cui, Y., Matsuura, K.: An Algorithm for Solving the LPN Problem and Its Application to Security Evaluation of the HB Protocols for RFID Authentication. In: Barua, R., Lange, T. (eds.) INDOCRYPT. Lecture Notes in Computer Science, vol. 4329, pp. 48–62. Springer (2006)
13. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M.J. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 1666, pp. 537–554. Springer (1999)
14. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) STOC. pp. 197–206. ACM (2008)
15. Gilbert, H., Robshaw, M.J.B., Seurin, Y.: How to Encrypt with the LPN Problem. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP (2). Lecture Notes in Computer Science, vol. 5126, pp. 679–690. Springer (2008)
16. Katz, J., Shin, J.S.: Parallel and Concurrent Security of the HB and HB$^+$ Protocols. In: Vaudenay, S. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 4004, pp. 73–87. Springer (2006)
17. Lee, P.J., Brickell, E.F.: An Observation on the Security of McEliece's Public-Key Cryptosystem. In: EUROCRYPT. pp. 275–280 (1988)
18. Levieil, É., Fouque, P.A.: An Improved LPN Algorithm. In: Prisco, R.D., Yung, M. (eds.) SCN. Lecture Notes in Computer Science, vol. 4116, pp. 348–359. Springer (2006)
19. Matsui, M. (ed.): Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings, Lecture Notes in Computer Science, vol. 5912. Springer (2009)
20. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) STOC. pp. 84–93. ACM (2005)
21. Schulman, L.J. (ed.): Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010. ACM (2010)
22. Stern, J.: A method for finding codewords of small weight. In: Cohen, G.D., Wolfmann, J. (eds.) Coding Theory and Applications. Lecture Notes in Computer Science, vol. 388, pp. 106–113. Springer (1988)