HELEN: a Public-key Cryptosystem Based on the LPN and the Decisional Minimal Distance Problems*

Alexandre Duc** and Serge Vaudenay

Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

Abstract. We propose HELEN, a 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. Our cryptosystem looks like the Alekhnovich cryptosystem. However, we carefully study its complexity and we further propose concrete optimized parameters.

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

1 Introduction

Every public-key cryptosystem relies on problems that are believed computationally hard. The two mostly used problems are the integer factorization problem [54,52] and the discrete logarithm problem [22]. However, these two problems can be solved in polynomial time on a quantum computer. It is thus important to develop new cryptosystem that are secure even on quantum computers and to correctly propose some parameters depending on the required security.

In this paper, we present HELEN, a public-key cryptosystem, the security of which relies on the hardness of the *Learning from Parity with Noise problem* (LPN) and the *minimum distance problem* which are both NP-hard.[1] The former consists in recovering an unknown vector while given access to noisy versions of its scalar product with random vectors. There is also no known polynomial-time algorithm on quantum computers. In short, the keys in HELEN consists in a low-weight parity check equation h (the private key) which is hidden in a random matrix G (the public key) such that it is indistinguishable from a totally random matrix. The matrix G spans a linear code. Our cryptosystem looks like the Alekhnovich cryptosystem [1]. However, we carefully study its

* This paper is an extended version of [19]

** Supported by a grant of the Swiss National Science Foundation, 200021_143899/1

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

complexity, we further propose concrete and optimized parameters, and we make incorrectness small.

We encrypt a duplicated bit by hiding it using a random linear codeword as well as a random biased noise vector. For decryption, the random linear codeword is removed by multiplying the ciphertext with h. The noise is removed by majority logic decoding. With a proper parameter choice, the probability of decrypting erroneously the message is small. We show in a further section how to reduce this probability of error as well as how to encrypt multiple bits at the same time using HELEN.

Related Work. The LPN problem is well studied in the cryptographic community. There is an authentication protocol based on the LPN problem named HB by Hopper and Blum [34]. This protocol was later improved into the HB^+ protocol by Juels and Weis [36]. However, HB^+ was shown vulnerable to man-in-the-middle attacks [28]. Several variants were proposed [12,21,47] but all of them suffer from the same vulnerability [29]. A new variant $\text{HB}^\#$ was proposed by Gilbert, Robshaw and Seurin [30] to improve the transmission cost of the protocol and its securtiy against man-in-the-middle attacks but an attack was also found in this variant [49]. Two more recent versions were introduced based on the hardness of some variant of the LPN problem, namely Ring-LPN [32] and subspace LPN [38].

Among other work based on the LPN problem, a PRNG is presented by Blum et al. in [10] along with a one-way function and a private-key encryption scheme based on some hard learning problems. A private-key encryption scheme named LPN-C was proposed by Gilbert, Robshaw and Seurin [31]. LPN-C was shown IND-CPA secure.

The construction of HELEN [19] presents some similarities with the trapdoor cipher TCHo [20,3,24] by Aumasson et al. which similarly encrypts a message by adding some random biased noise and some contribution from a linear code. In TCHo, this noise is introduced using an LFSR whose feedback polynomial has a multiple of low weight.

A class of lattice-based cryptosystems introduced by Regev is based on the worst-case complexity of the *learning with errors* (LWE) problem [53,50,43,57], which is a generalisation of the LPN problem on fields $\mathbb{F}_q$ with $q > 2$. The last two introduce the *ring-LWE* problem, an algebraic variant of the LWE problem. According to the authors, it is the first truly practical lattice-based cryptosystem based on the LWE problem.

Other well-known post-quantum cryptosystems include the McEliece cryptosystem [46] and its dual the Niederreiter cryptosystem [48], which are code-based making use of Goppa codes. In lattice-based cryptosystem, one has to mention NTRU [33] based on the hardness of the shortest vector problem in a particular class of lattices. We refer the reader to [7] for a more exhaustive survey on post-quantum cryptosystems.

More closely related cryptosystems were proposed. Gentry et al. proposed an LWE-based cryptosystem [27] 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 problem to distinguish $(A, Ax + e)$ with x following uniform distribution and e either in $\binom{n}{n^\delta}$ or $\binom{n}{n^\delta+1}$ with $\delta < 1/2$ which he conjectures to be hard [1]. Our scheme differs with the scheme proposed in [1] in the following ways. First, we encode the bit so that decryption is correct with constant probability ϕ and which is independent from the encrypted bit b (in [1], this probability is just known to be close to one for $b = 0$ and $1/2$ for $b = 1$). Finally, we propose concrete parameters and asymptotic parameters for our scheme. 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 of weight 3 instead of random vectors. Also, the authors do not provide any concrete parameters [2]. n Asiacrypt 2012, Döttling et al. presented an IND-CCA secure cryptosystem based on Alekhnovich's scheme, but again, no concrete parameters are given [18]. IND-CCA security is obtained using a technique by Dolev et al. [17] based on one-time signatures and a tool by Rosen and Segev [55]. 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 by log the logarithm in base two. The concatenation of two bitstrings x and y is written $x \| y$. We consider vectors as row vectors. The transpose of a vector $\boldsymbol{v}$ is denoted by $\boldsymbol{v}^t$. We denote the Hamming weight of a bitstring x by $\mathsf{wt}(x)$. We write $x \overset{U}{\longleftarrow} \mathcal{D}$ if an element x is drawn uniformly at random in a domain $\mathcal{D}$. A function $f(\lambda)$ is *negligible* if for all $d \in \mathbb{R}$ we have $f(\lambda) = O\left(\lambda^{-d}\right)$. We denote the Bernoulli distribution with parameter p by $\mathrm{Ber}(p)$, i.e., if $x \leftarrow \mathrm{Ber}(p)$, we have $\Pr[x = 1] = p$ and $\Pr[x = 0] = 1 - p$. We write S_p^n to denote 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 σ 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. That is, $(\sigma \star h)_i = h_{\sigma^{-1}(i)}$. Given a $k \times n$ matrix G, we write $\sigma \star G$ when we apply σ on the columns of G, i.e., $(\sigma \star G)_{i,j} = G_{i,\sigma^{-1}(j)}$.

We will need the following Chernoff bound.

Lemma 1. *Let $X_1, \ldots, X_n$ be iid random variables such that $X_i \sim \mathrm{Ber}(p)$. Let $X \coloneqq \sum_{i=1}^n X_i$.*

$$\Pr\left[X \geq (p + \epsilon)n\right] \leq e^{-2n\epsilon^2} .$$

Notation. Given some initial parameters $\mathit{\Pi}$ and a predicate P, we write

$$\Pr\left[P(v_1,\ldots,v_m;r_p): \quad \begin{array}{l} v_1 \leftarrow f_1(\mathit{\Pi};r_1) \\ \vdots \\ v_m \leftarrow f_m(\mathit{\Pi},v_1,\ldots,v_{m-1};r_m) \end{array}\right]$$

to denote the probability (over the randomnesses $r_1,\ldots,r_m,r_p$) that there exist $v_1 \leftarrow f_1(\mathit{\Pi};r_1),\ldots,v_m \leftarrow f_m(\mathit{\Pi},v_1,\ldots,v_m;r_m)$ such that $P(v_1,\ldots,v_m;r_p)$.

2.1 Security Notions

Definition 2 (Public-key Encryption Scheme). *Given a function $\varphi(\lambda)$, a $\varphi(\lambda)$-cryptosystem over a given message space $\mathcal{M}$ and random coin space $\mathcal{R}$ consists of three* polynomial-time *algorithms:*

- *a probabilistic key-generation algorithm $\mathsf{Gen}(1^\lambda;\rho_g)$ taking as input some security parameter 1^λ in unary representation and some random coins ρ_g, and producing a secret key K_s and a public key K_p;*
- *a probabilistic encryption algorithm $\mathsf{Enc}(K_p,m;r)$ taking as input a public key K_p and a message $m \in \mathcal{M}$ with some random coins $r \in \mathcal{R}$, and producing a ciphertext y in the ciphertext space $\mathcal{C}$;*
- *a deterministic decryption algorithm $\mathsf{Dec}(K_s,c)$ taking as input a secret key K_s and a ciphertext $c \in \mathcal{C}$, and producing a message or an error.*

The cryptosystem must satisfy the following correctness property:

$$\max_{m\in\mathcal{M}} \Pr\left[\mathsf{Dec}(K_s,\mathsf{Enc}(K_p,m;\rho)) \neq m\colon \qquad (K_s,K_p) \leftarrow \mathsf{Gen}(1^\lambda;\rho_g)\right] \leq \varphi(\lambda)\ .$$

We will also use the following security notions and acronyms. Adaptive Chosen Ciphertext Attack is denoted CCA, Chosen Plaintext Attack CPA, Indistinguishability IND and one-wayness OW.

Definition 3 (IND-CPA-security). *A cryptosystem is said (t,ε)-*IND-CPA*-secure or (t,ε)-*semantically secure *against chosen plaintext attacks if no adversary $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ with running time bounded by t can distinguish the encryption of two different plaintexts m_0 and m_1 with a probability higher than ε.*[2] *More formally, for all $\mathcal{A}$ bounded by t,*

$$\Pr\left[\mathcal{A}_2(K_p,c;\rho)=b: \quad \begin{array}{l} (K_s,K_p) \leftarrow \mathsf{Gen}(1^\lambda;\rho_g) \\ m_0,m_1 \leftarrow \mathcal{A}_1(K_p;\rho) \\ r \xleftarrow{U} \mathcal{R};\ b \xleftarrow{U} \{0,1\} \\ c \leftarrow \mathsf{Enc}(K_p,m_b;r) \end{array}\right] \leq \frac{1}{2}+\varepsilon\ .$$

Asymptotically, a cryptosystem is IND-CPA*-secure if for any polynomial $t(\lambda)$ there exists a negligible function $\varepsilon(\lambda)$ such that it is $(t(\lambda),\varepsilon(\lambda))$-*IND-CPA*-secure.*

[2] We include in the running time the size of the code of $\mathcal{A}$ in a fixed RAM model of computation to avoid trivial adversaries.

IND-CPA-security can also be represented in the simple real-or-random game model [6,5].[3]

Definition 4 (Simple real-or-random IND-CPA game security). *A cryptosystem is (t, ε)-*IND-CPA*-secure in the real-or-random game model if no adversary $\mathcal{A}$ with running time bounded by t can distinguish the encryption of a chosen plaintext m_0 to a random one with a probability higher than ε. More formally, for all $\mathcal{A}$ bounded by t,*

$$\Pr\left[\mathcal{A}_2(K_p, c; \rho) = b : \begin{array}{l} (K_s, K_p) \leftarrow \mathsf{Gen}(1^\lambda; \rho_g) \\ m_0 \leftarrow \mathcal{A}_1(K_p; \rho);\ m_1 \overset{U}{\longleftarrow} \mathcal{M} \\ r \overset{U}{\longleftarrow} \mathcal{R};\ b \overset{U}{\longleftarrow} \{0,1\} \\ c \leftarrow \mathsf{Enc}(K_p, m_b; r) \end{array}\right] \leq \frac{1}{2} + \varepsilon \ .$$

Asymptotically, a cryptosystem is IND-CPA*-secure in the real-or-random game model if for any polynomial $t(\lambda)$ there exists a negligible function $\varepsilon(\lambda)$ such that it is $(t(\lambda), \varepsilon(\lambda))$-*IND-CPA*-secure in the real-or-random game model.*

A (t, ε)-IND-CPA-secure system in the real-or-random game model is $(t, 2\varepsilon)$-IND-CPA-secure [5]. Conversely, a (t, ε)-IND-CPA-secure system is (t, ε)-IND-CPA-secure in the real-or-random game model. Asymptotically, both models are equivalent.

Definition 5 (IND-CCA-security). *A cryptosystem is said (t, ε)-*IND-CCA*-secure or (t, ε)-*secure *against adaptive chosen ciphertext attacks if no adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, with access to a decryption oracle $\mathcal{O}_{K_s}$ and with running time bounded by t can distinguish the encryption of two different plaintexts m_0 and m_1 with a probability higher than ε. More formally, for all $\mathcal{A}$ bounded by t,*

$$\Pr\left[\mathcal{A}_2^{\mathcal{O}_{K_s}}(K_p, c; \rho) = b : \begin{array}{l} (K_s, K_p) \leftarrow \mathsf{Gen}(1^\lambda; \rho_g) \\ m_0, m_1 \leftarrow \mathcal{A}_1^{\mathcal{O}_{K_s}}(K_p; \rho) \\ r \overset{U}{\longleftarrow} \mathcal{R};\ b \overset{U}{\longleftarrow} \{0,1\} \\ c \leftarrow \mathsf{Enc}(K_p, m_b; r) \end{array}\right] \leq \frac{1}{2} + \varepsilon \ ,$$

where $\mathcal{O}_{K_s,c}(y) = \mathsf{Dec}(K_s, y)$ for $y \neq c$ and $\mathcal{O}_{K_s,c}(c) = \perp$. Asymptotically, a cryptosystem is IND-CCA*-secure if for any polynomial $t(\lambda)$ there exists a negligible function $\varepsilon(\lambda)$ such that it is $(t(\lambda), \varepsilon(\lambda))$-*IND-CCA*-secure.*

Definition 6 (Statistical distance). *Given two discrete distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ over a set $\mathcal{Z}$, we define the* statistical distance *between $\mathcal{D}_0$ and $\mathcal{D}_1$ by*

$$d(\mathcal{D}_0, \mathcal{D}_1) \coloneqq \frac{1}{2} \sum_{z \in \mathcal{Z}} |\mathcal{D}_1(z) - \mathcal{D}_0(z)| \ .$$

[3] In our definition of real-or-random game model, we consider only *simple* adversaries, i.e., adversaries who can query the oracle once. This definition is enough to prove the IND-CPA-security of our scheme.

Definition 7. *Given two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$, a* distinguisher *between them is an algorithm $\mathcal{A}$ that takes as input one sample x from either $\mathcal{D}_0$ or $\mathcal{D}_1$ and has to decide which distribution was used. Its* advantage *is*

$$\mathrm{Adv}_{\mathcal{A}}(\mathcal{D}_0, \mathcal{D}_1) = \Pr\left[\mathcal{A}(x) = 1 \colon x \leftarrow \mathcal{D}_1\right] - \Pr\left[\mathcal{A}(x) = 1 \colon x \leftarrow \mathcal{D}_0\right] \ .$$

We know that for all $\mathcal{A}$, $\mathrm{Adv}_{\mathcal{A}}(\mathcal{D}_0, \mathcal{D}_1) \leq d(\mathcal{D}_0, \mathcal{D}_1)$. Equality is reached for $\mathcal{A}$ defined by $\mathcal{A}(x) = 1$ iff $\mathcal{D}_1(x) \geq \mathcal{D}_0(x)$.

We say that $\mathcal{D}_0$ and $\mathcal{D}_1$ are ϵ-statistically indistinguishable *if $d(\mathcal{D}_0, \mathcal{D}_1) \leq \epsilon$.*

We say that the two distributions are (t, ε)-computationally indistinguishable *if for any distinguisher $\mathcal{A}$ with running time bounded by t,*

$$|\,\mathrm{Adv}_{\mathcal{A}}(\mathcal{D}_0, \mathcal{D}_1)| \leq \varepsilon \ .$$

Asymptotically, two distributions depending on a parameter λ are computationally indistinguishable *if for any polynomial $t(\lambda)$ there exists a negligible function $\varepsilon(\lambda)$ such that, they are $(t(\lambda), \varepsilon(\lambda))$-computationally indistinguishable.*

2.2 The Learning from Parity with Noise Problem

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. More formally

Definition 8 (LPN Oracle). *An* LPN oracle $\varPi_{\boldsymbol{u},p}$ *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, $\nu \leftarrow \mathrm{Ber}(p)$. Note that the output is a $k+1$-bit vector.

Problem 9 (Learning from Parity with Noise Problem). The (k, p)-Learning from Parity with Noise Problem ((k, p)-LPN) consists, given an LPN Oracle $\varPi_{\boldsymbol{u},p}$, 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

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

The Decisional LPN Problem. 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

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

It is shown [37,53] that if there exists an algorithm $\mathcal{A}$ that (t, n, δ)-solves the (k, p)-D-LPN problem, then there is an algorithm $\mathcal{A}'$ that $(t', n', \delta/4)$-solves the (k, p)-LPN problem, with $t' := O\left(t \cdot k\delta^{-2} \log k\right)$ and $n' := O\left(n \cdot \delta^{-2} \log k\right)$. Thus, the hardness of the LPN problem implies that the output of the LPN vector oracle is indistinguishable from a random source.

We say that the (k, p)-D-LPN problem is (t, ϵ)-hard, if there is no known algorithm that solves it with running time bounded by t and advantage higher than ϵ.

Algorithms that Solve the LPN Problem The first subexponential algorithm to solve the LPN problem was given by Blum, Kalai, and Wasserman in [11] and they estimated its complexity to $2^{O(k/\log k)}$. We denote this algorithm by BKW algorithm.

The idea of the BKW algorithm is to first query the LPN oracle to obtain a large amount of LPN vectors. It searches then for basis vectors e_j by finding a low amount of vectors that xor to e_j. If the number of vectors that xor to e_j is small, the noise for this vector will be small as well. Using different independent instances that xor to the same e_j, one can recover the jth bit of $\boldsymbol{u}$ with good probability. All this procedure can be done using a large amount of queries.

The BKW algorithm was analyzed in details and improved in [40,25]. We give here the complexity of the improvement given in [40] that we will use as a security bound in our cryptosystem.

Theorem 10 ([40], Theorem 2). *For $b \geq 1$, let $a := k/b$ and $q := (8b+200) \times (1-2p)^{-2^a} + (a-1) \times 2^b$. There exists an algorithm that $(kaq, q, \frac{1}{2})$-solves the (k, p)-LPN problem.*

Some parameters along with their security are given in [40, Section 5.2]. This algorithm requires a subexponential (in k) number of samples. When the number of samples is polynomial (as it is in our case), Lyubashevsky showed that one can scramble randomly the samples to get more of them with a higher noise level [42]. Then, the problem is solvable in $2^{O(k/\log\log k)}$. More precisely, one can transform the (k, p)-LPN problem with $k^{1+\epsilon}$ samples in the (k, p')-LPN problem with enough samples to use the BKW algorithm and with

$$p' = \frac{1}{2} - \frac{1}{2}\left(\frac{1}{4} - \frac{p}{2}\right)^{\frac{2k}{\epsilon \log k}} . \tag{1}$$

Combining this idea with Theorem 10, we get the following time complexity ($\mathsf{T}_{\mathsf{LPN}}$) for solving LPN and we will use it as a security bound in our cryptosystem.

Theorem 11 (LPN with limited number of queries). *For $b \geq 1$, let $q := k^{1+\epsilon}$, and let*

$$\mathsf{T}_{\mathsf{LPN}} := \min_{0<a\leq k} \left(k \times a \times \left(\left(\frac{8k}{a} + 200\right) \times (1-2p')^{-2^a} + (a-1) \times 2^{\frac{k}{a}}\right)\right) , \tag{2}$$

where p' is given in Equation (1). *There exists an algorithm that* $(\mathsf{T}_{\mathsf{LPN}}, q, \frac{1}{2})$*-solves the* (k, p)*-LPN problem.*

2.3 Finding a Low-weight Codeword in a Random Linear Code

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 12 (Minimum Distance Problem (MDP)). The (n, k, w)-*decisional minimum distance problem* is the following. Given an $(n-k) \times n$ matrix H drawn uniformly 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.

Its hardness remained open for a long time. It was even set the "open problem of the month" in [35]. It was finally shown to be NP-hard by Vardy [59] using a reduction from the decisional syndrome decoding problem. Many algorithms solving this problem were developed (e.g. [39,58,13,14,15,23].)

Finally, a general lower-bound on the complexity of the information set decoding algorithm was derived by Finiasz and Sendrier [23] using idealized algorithms. However, it was shown in [9,45] and very recently in [4] that it is possible to do better than this bound.

A new lower-bound for information set decoding is proposed in [9]. This bound is much simpler and we give it in Assumption 13.

Assumption 13 ([9]). Let $r := n - k$. Given an $[n, k]$-code and given a weight w, if $\binom{n}{w} \leq 2^r$, the cost of finding a parity-check equation of weight w is lower-bounded by

$$\mathsf{T}_{\mathsf{MDP}}(w, n, k) := \min_i \frac{\binom{n}{w}}{2\binom{k}{w-i}\sqrt{\binom{r}{i}}} \ , \tag{3}$$

bit operations, with $r = n - k$.

We will assume this lower-bound for our cryptosystem. Note that a similar analysis for linear codes over a general field $\mathbb{F}_q$ is presented in [51].

3 The Cryptosystem

We will first consider how to encrypt one single bit b. Hence, our message space is $\mathcal{M} = \{0, 1\}$. We denote the cryptosystem by HELEN. We generalize the encryption to multiple bits in Section 6.

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 $\sigma \star h = h'$. In the following, $\mathcal{H}$ will be the set of all vectors of weight w and dimension n but we keep this more general $\mathcal{H}$ for further improvements. We also hide then the message further by adding some low weight random noise vector $\nu \in \{0,1\}^n$ produced by a source S_p.

For correct decryption, we require also that $h \cdot c^t = 1$ for all $h \in \mathcal{H}$. When $\mathcal{H}$ contains all the vectors of weight w, this condition implies $c = (1, \dots, 1)$ (see (4) below).

In the following, we describe more precisely the cryptosystem. All algorithms are summarized in Algorithm 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

$$b' := \mathsf{BDec}(h, y) = h \cdot y^t \ .$$

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, we get

$$b' := \mathsf{BDec}(h, y) = h \cdot y^t = (h \cdot c^t \cdot b^t) \oplus \nu' \ ,$$

for $\nu' := h \cdot \nu$ a noise with

$$\Pr[\nu' = 1] = \frac{1 - (1-2p)^w}{2}$$

by Lemma 15. Note that it is necessary that

$$h \cdot c^t = 1 \tag{4}$$

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. The resulting bit b' is then different from b with probability φ, which is given in the following theorem.

Theorem 14. *HELEN is a φ-cryptosystem, where*

$$\varphi := \frac{1 - (1 - 2p)^w}{2} \ .$$

Note that the complexity of decryption is $O\left(n\right)$.

Lemma 15. *Let X be a random variable defined as the sum modulo 2 of w iid Bernoulli random variables equals to 1 with probability p and to 0 else. Then*

$$\Pr[X = 1] = \frac{1 - (1 - 2p)^w}{2} \ .$$

Proof. We have

$$1 - 2\Pr[X = 1] = \mathbb{E}\left[(-1)^X\right] = (1 - 2p)^w$$

which shows the result. □

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 $0 < u \le n$ 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 $1 \le i \le k$ and $1 \le j \le n$, $j \ne u$.
4. Let
$$g_{iu} = \sum_{\substack{1 \le j \le n \\ j \ne u}} g_{ij} h_j$$
for $1 \le i \le k$, where the sum is taken over $\mathbb{F}_2$.
5. Return the matrix $G := [g_{ij}]_{\substack{1 \le i \le k \\ 1 \le j \le n}}$ 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 $O\left(k \times n\right)$. Note that we have $hG^t = 0$.

Algorithm 1 Algorithm to generate keys, to encrypt, and to decrypt.

```
Key Generation:
Input: Lengths $k, n$ and a set $\mathcal{H}$.
Output: A private key $h$ and a public key $G$.
Draw a random vector $h$ of length $n$ in the set $\mathcal{H}$.
Let $0 < u \leq n$ be any index of $h$ such that $h_i = 1$ , e.g., $\max\{i \colon h_i = 1\}$.
Let $g_{ij} \leftarrow \mathrm{Ber}(\frac{1}{2})$, for $1 \leq i \leq k$ and $1 \leq j \leq n$, $j \neq u$.
Let
    $$g_{iu} = \sum_{\substack{1 \leq j \leq n \\ j \neq u}} g_{ij} h_j$$
    for $1 \leq i \leq k$, where the sum is taken over $\mathbb{F}_2$.
return the matrix $G \coloneqq [g_{ij}]_{\substack{1 \leq i \leq k \\ 1 \leq j \leq n}}$ and the vector $h$.
Encryption:
Input: A bit $b$ to encrypt, a public key $G$, two random seeds $r_1$ and $r_2$, a length $n$,
    an $n$-bit vector $c$, and a noise parameter $p$.
Output: A ciphertext $y$ encrypted under the public key $G$.
Let $\nu \coloneqq \mathrm{S}_p^n(r_2)$.
return $y \leftarrow b \cdot c \oplus r_1 G \oplus \nu$.
Decryption:
Input: A ciphertext $y$ and a private key $h$.
Output: The original plaintext $b$ with probability $\varphi$ defined in Theorem 14.
return $b' \leftarrow h \cdot y^t$.
```

4 Security Analysis

We will reduce the security of our scheme to the LPN problem presented in Section 2.2. To do this, we will proceed in two steps. First, we show that the code we construct for C_2 is computationally indistinguishable from a random matrix.

4.1 Link to Random Codes

We will compare the distributions of the output of different generators and show that their statistical distance is negligible using various lemmas. We conclude in Theorem 19. The first generator is our key generation algorithm.

Generator A: Run the key generation algorithm to obtain G and h and return $A \coloneqq G$.

Generator G_1: Run generator A until the resulting matrix G has only one parity check equation in $\mathcal{H}$ and return $G_1 \coloneqq G$.

Generator G_2: Draw a random $k \times n$ matrix G_2 until it has *a single* parity check equation in $\mathcal{H}$ and return G_2.

Generator G_3: Draw a random $k \times n$ matrix G_3 until it has *at least one* parity check equation in $\mathcal{H}$ and return G_3.

Generator B: Return a random $k \times n$ matrix B.

In the following, we show that the statistical distance between A and G_3 is negligible for suitable parameters.

Lemma 16.

$$d(G_1, G_2) = 0 \ .$$

Proof. Recall that there exists a subgroup P of $\mathfrak{S}_n$ that acts transitively on $\mathcal{H}$. Clearly, G_2 generates a uniform distribution among all G's which have a unique parity check equation in $\mathcal{H}$. So, we just have to prove that G_1 has the same distribution. Clearly, $hG^t = 0$ if and only if $(\sigma \star h) \times (\sigma \star G)^t = 0$.

Also, A generates uniformly a pair (h, G) with $h \in \mathcal{H}$ and G such that $hG^t = 0$. Let $\mathcal{G}_h$ be the set of all G's for which h is the only element of $\mathcal{H}$ satisfying $hG^t = 0$. For any $h \in \mathcal{H}$ and any $G \in \mathcal{G}_h$, we have

$$\Pr[G_1 \to G] = \frac{1}{\#\mathcal{H} \times \#\mathcal{G}_h}$$

Due to the above property on the action $\star$, any σ induces a permutation from $\mathcal{G}_h$ to $\mathcal{G}_{\sigma \star h}$. Since the action is further transitive, all $\mathcal{G}_h$'s have same cardinality. Hence, G_1 generates a uniform distribution among all the G's which have a unique parity check equation in $\mathcal{H}$. □

Lemma 17.

$$d(G_2, G_3) \leq \frac{(\#\mathcal{H} - 1)\#\mathcal{H}}{2^{k+1}} \ .$$

Proof. Let $p_1(G_3)$ denote the probability that generator G_3 has exactly one parity-check equation in $\mathcal{H}$. The best distinguisher between G_2 and G_3 outputs 1 if and only if the generated matrix has two or more parity-check equations in $\mathcal{H}$. So, $d(G_2, G_3) = 1 - p_1(G_3)$.

Let a (resp. b) be the probability that a random matrix verifies at least one (resp. two) parity-check equations in $\mathcal{H}$. Then

$$a \geq 2^{-k},$$

since any parity-check equation is verified with probability exactly 2^{-k}. Similarly,

$$b \leq \frac{(\#\mathcal{H})(\#\mathcal{H} - 1)}{2} \times 2^{-2k}$$

Thus,

$$d(G_2, G_3) = 1 - p_1(G_3) = \frac{b}{a} \leq \frac{(\#\mathcal{H} - 1)\#\mathcal{H}}{2} \times 2^{-k} \ .$$

□

Lemma 18.

$$d(A, G_1) \leq \frac{\#\mathcal{H} - 1}{2^k} \ .$$

Proof. Let $p_1(A)$ denote the probability that the output of generator A has exactly one parity-check equation in $\mathcal{H}$. The best distinguisher between A and G_1 checks if the generated matrix has only one parity-check equation. So, $d(A, G_1) = 1 - p_1(A) \leq \frac{\#\mathcal{H}-1}{2^k}$ since we are looking for a second parity-check equation in a random matrix which has already one of them. □

Theorem 19. *Assume that there exists a subgroup P of $\mathfrak{S}_n$ that acts transitively on $\mathcal{H}$. Then,*

$$d(A, G_3) \leq \frac{(\#\mathcal{H} - 1)(\#\mathcal{H} + 2)}{2^{k+1}} =: \mathsf{D}_{\mathsf{A},\mathsf{G}_3} \ . \tag{5}$$

Proof. We apply the triangular inequality to the following path:

$$A \leftrightarrow G_1 \leftrightarrow G_2 \leftrightarrow G_3 \ .$$

Summing the distances proven in Lemmas 18, 16, and 17 we get the wanted result. □

We want now to link this distribution with the distribution of an uniformly distributed $k \times n$ matrix, i.e., a matrix produced by generator B. We will need suitable parameters such that G_3 is *computationally indistinguishable* from B.

The best distinguisher between G_3 and B consists in deciding whether the output of the unknown generator has a parity-check equation in $\mathcal{H}$ or not. As discussed, the decisional problem is believed as hard as the computational problem. Hence, we extend Assumption 13 to the following one.

Assumption 20. For any distinguisher between G_3 and B, the complexity over advantage ratio is lower bounded by $\mathsf{T}_{\mathsf{MDP}}(w, n, k)$, which is defined in (3).

So, by selecting parameters such that the right-hand side of (5) is negligible and such that $\mathsf{T}_{\mathsf{MDP}}(w, n, k) \geq 2^\lambda$, for a security parameter λ, any game involving our cryptosystem produces a computationally indistinguishable outcome when the key generator is replaced by B.

4.2 Semantic Security

Now that we have B computationally indistinguishable from A, we can link our cryptosystem with the LPN problem.

Theorem 21. *Let $\varepsilon_0 := d(A, G_3)$ as defined in Theorem 19. 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*

$$(\min\{t_1, t_2 - \tau kn\}, 2(\varepsilon_0 + \varepsilon_1 + \varepsilon_2)) \text{ -}\mathsf{IND\text{-}CPA}\text{-}\textit{secure} \ .$$

Proof. We introduce the following three games Γ_0, Γ_1 and Γ_2. Γ_0 is the $\mathsf{IND\text{-}CPA}$ game for our cryptosystem in the simple real-or-random model. Γ_1 is the $\mathsf{IND\text{-}CPA}$ game in the same model but using generator B instead of A. Γ_2 is the (k,p)-D-LPN game.

By the assumptions, we know that the best advantage between Γ_0 and Γ_1 is $\varepsilon_1 + \varepsilon_2$.

For the best advantage between Γ_1 and Γ_2, we do the following. Recall that in the simple real-and-random game this model, the adversary submits first a chosen plaintext b using an algorithm $\mathcal{A}_1^{\mathsf{ror}}(G)$. Then, given a n-bit word u, has to decide using an algorithm $\mathcal{A}_2^{\mathsf{ror}}(G,u)$, whether u is the encryption of b or is a random bitstring. Let $(\mathcal{A}_1^{\mathsf{ror}}(G), \mathcal{A}_2^{\mathsf{ror}}(G,u))$ be an $\mathsf{IND\text{-}CPA}$ adversary for our cryptosystem when G is generated using generator B.

We show that using this adversary, we can solve the D-LPN problem. We query first the unknown oracle of the D-LPN problem n times to obtain n-vectors $\alpha_1, \ldots, \alpha_n$. Note that each of these α_i has exactly $k+1$ bits. We create now the $k \times n$ matrix $\tilde{G}$ using the first k bits of α_i as column i, for $1 \le i \le n$. Using $\mathcal{A}_1^{\mathsf{ror}}(\tilde{G})$, we recover a plaintext b. Let $z \coloneqq b \cdot c \oplus (\alpha_{1|k+1} \| \ldots \| \alpha_{n|k+1})$, where $\alpha_{i|k+1}$ denotes the $k+1$-th bit of α_i. If the unknown oracle returns random bitstrings, then z will be random as well. However, if it is an LPN oracle, then z is a valid ciphertext of b using the public key $\tilde{G}$. Note also that the matrix $\tilde{G}$ follows the same distribution as the output of generator B.

Hence, using $\mathcal{A}_2^{\mathsf{ror}}(\tilde{G}, z)$, we can decide whether z is a ciphertext corresponding to b or not. The complexity of this simulation is τkn for a constant $\tau > 0$ large enough. Thus, the advantage between game Γ_1 and Γ_2 is zero.

Since the D-LPN problem is supposed (t_2, ε_2)-hard, we get that our cryptosystem when we use generator B is $(t_2 - \tau kn, \varepsilon_2)$-IND-CPA-secure in the simple real-or-random model. Similarly, we get that the original cryptosystem is $(\min\{t_1, t_2 - \tau kn\}, \varepsilon_0 + \varepsilon_1 + \varepsilon_2)$-$\mathsf{IND\text{-}CPA}$-secure in the simple real-or-random model. Thus, our cryptosystem is $(\min\{t_1, t_2 - \tau kn\}, 2(\varepsilon_0 + \varepsilon_1 + \varepsilon_2))$-$\mathsf{IND\text{-}CPA}$-secure in the standard model [6]. □

Hence, we reduced the semantic security of our cryptosystem to the hardness of the decisional LPN problem with n queries and noise parameter p.

Note that since we encrypt one single bit, an $\mathsf{IND\text{-}CPA}$ adversary has to distinguish $\mathsf{BEnc}(G,0)$ from $\mathsf{BEnc}(G,1)$ which is equivalent to $\mathsf{OW\text{-}CPA}$ security.

5 Selection of Parameters

To summarize, we need to tune the following security parameters for HELEN:

- The dimension k of the code C_2 generated by G,
- The ciphertext length n (also the length of the codewords in C_2),
- The weight w of the secret key, and
- The noise probability p.

For our cryptosystem to be semantically secure, we need the parameters to verify Theorem 21. In particular, this implies that the D-LPN problem should be hard, that finding a low-weight parity-check equation in the code is hard as well, i.e., that $\mathsf{T}_{\mathsf{MDP}}(w, n, k) \geq 2^{\lambda}$ and that the statistical distance $\mathsf{D}_{\mathsf{A},\mathsf{G}_3}$ defined in Theorem 19 is lower than $2^{-\lambda}$. We need also w to be odd. For the LPN problem, we want $\mathsf{T}_{\mathsf{LPN}} \geq 2^{\lambda}$, where $\mathsf{T}_{\mathsf{LPN}}$ is given in Equation (2).

Recall that the probability of decrypting incorrectly a bit is

$$P_{\mathsf{error}} := \frac{1-(1-2p)^w}{2} \ . \tag{6}$$

Hence, 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 $C := 1 - H_2(P_{\mathsf{error}})$ with $H_2(p) := -p\log(p) - (1-p)\log(1-p)$. 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 (I) which minimizes the n/C ratio to minimize the number of transmitted bits and some (II) with a smaller kn/C ratio to minimize the encryption/decryption complexity. We give in Table 1 concrete parameters for different security parameters λ.

Table 1. Parameters for our cryptosystem

	λ	k	n	w	p	kn	n/C	kn/C	$\mathsf{T}_{\mathsf{MDP}}$	$\mathsf{D}_{\mathsf{A},\mathsf{G}_3}$	$\mathsf{T}_{\mathsf{LPN}}$	C
I	64	4 500	18 000	33	0.01	$2^{26.3}$	$2^{16.4}$	$2^{28.6}$	$2^{65.3}$	2^{-3813}	$\geq 2^k$	0.20
II	64	2 200	16 000	23	0.02	$2^{25.0}$	$2^{17.1}$	$2^{28.2}$	$2^{64.7}$	2^{-1707}	$\geq 2^k$	0.11
I	80	5 600	28 000	35	0.01	$2^{27.2}$	$2^{17.2}$	$2^{29.7}$	$2^{80.5}$	2^{-4832}	$\geq 2^k$	0.18
II	80	2 800	27 000	25	0.02	$2^{26.2}$	$2^{18.1}$	$2^{29.6}$	$2^{80.4}$	2^{-2232}	$\geq 2^k$	0.10

In Table 2, we compare for concrete parameters HELEN with the code-based McEliece cryptosystem [46] and with an LWE-based cryptosystem [41]. Note that for encryption and decryption time, we neglect the cost of encoding and decoding.

We propose the following asymptotic parameters for our system:

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

Indeed, we obtain $\mathsf{T}_{\mathsf{MDP}}$ and $\mathsf{T}_{\mathsf{LPN}} \geq 2^{\lambda}$, $\mathsf{D}_{\mathsf{A},\mathsf{G}_3} \leq 2^{-\lambda}$, $P_{\mathsf{error}} = \frac{1}{2} - \frac{1}{e^{O(1)}}$, and $C > 0$. In Table 3, we compare the asymptotic parameters.

6 Encrypting More than One Bit

In this section, we show how to encrypt more than one bit using HELEN. Taking advantage of an efficient coding scheme, we can also improve the proba-

Table 2. Comparison with other cryptosystems

Name	λ	Message expansion	Pub key size	Encryption time	Decryption time
HELEN I	80	$2^{17.2}$	$2^{27.2}$	$O\left(2^{29.7}\right)$	$O\left(2^{17.2}\right)$
McEliece [8]	80	1.29	$2^{18.8}$	$O\left(2^{21.0}\right)$	$O\left(2^{21.3}\right)$
LWE [41]	128	22	$2^{17.5}$	$O\left(2^{24}\right)$	$O\left(2^{18.5}\right)$
Ring-LWE [41]	128	22	$\approx 2^{10}$	$O\left(2^{24}\right)$	$O\left(2^{18.5}\right)$

Table 3. Asymptotic comparison with other cryptosystems. The $\Theta(.)$'s have been omitted.

Name	Message expansion	Public key size	Private key size	Key generation	Encryption	Decryption
HELEN	λ^2	λ^4	$\lambda \log \lambda$	λ^4	λ^4	λ^2
TCHo	λ^2	λ^2	$\lambda \log \lambda$	$\lambda^6 \log \lambda \log \log \lambda$	λ^5	λ^4
McEliece	1	λ^2	λ^2	λ^3	λ^2	$\lambda^2 \log \lambda$
RSA	1	λ^3	λ^3	λ^{12}	λ^6	λ^9
NTRU	1	λ	λ	λ^3	λ^2	λ^2

bility of decrypting correctly the message. In addition to the previous parameters n, k, p, w and $\mathcal{H}$ we add a $[\mu, \kappa]$-error-correcting code. Let Encode be this $[\mu, \kappa]$-error-correcting code. Let also Decode be an efficient decoding algorithm corresponding to this code.

Encryption: We encrypt a plaintext $m \in \{0,1\}^\kappa$ in two steps. First we compute $b_1 \| \ldots \| b_\mu := \mathsf{Encode}(m)$. The ciphertext c is then $\mathsf{BEnc}(G, b_1) \| \ldots \| \mathsf{BEnc}(G, b_\mu)$. The complexity of encryption is $O\left(\mu k n + T_{\mathsf{Encode}}\right)$, where T_{Encode} is the complexity of the encoding algorithm.

Decryption: To decrypt, we first decrypt each block of n bits using BDec to recover $b'_1 \| \ldots \| b'_\mu$, where each $b'_i \neq b_i$ with probability $(1-(1/2p)^w)/2 =: P_{\mathsf{error}}$. The complexity of decryption is $O\left(\mu n + T_{\mathsf{Decode}}\right)$, where T_{Decode} is the complexity of the decoding algorithm. Let ρ be the maximum number of errors the error-correcting code can correct. Then, the probability of decrypting incorrectly the message is

$$\sum_{i=\rho+1}^{\mu} \binom{\mu}{i} (P_{\mathsf{error}})^i (1-P_{\mathsf{error}})^{\mu-i} \leq \exp\left[-2\mu\left(\frac{\rho}{\mu} - P_{\mathsf{error}}\right)^2\right] =: \phi \qquad (7)$$

by Lemma 1.

Theorem 22. *HELEN with parameter μ, κ is a ϕ-cryptosystem, where ϕ is given in* (7).

6.1 Security

Theorem 23. *Let ε_b be the* $\mathsf{IND\text{-}CPA}$ *advantage for the elementary cryptosystem HELEN with $\mu = \kappa = 1$. Then, the advantage of an* $\mathsf{IND\text{-}CPA}$ *adversary against the full cryptosystem HELEN with parameter μ and κ is smaller than $\mu\varepsilon_b$.*

The proof is a standard hybrid argument and can be found in Appendix A.

6.2 IND-CCA-security

Obviously HELEN is not $\mathsf{IND\text{-}CCA}$-secure, since it is clearly malleable. It suffices to change one single bit of the ciphertext and to submit it to the decryption oracle to decrypt the plaintext with good probability. To achieve $\mathsf{IND\text{-}CCA}$ security, one can use well-known construction like the Fujisaki-Okamoto hybrid construction [26]. This construction uses two random oracles H_1 and H_2 as well as a symmetric encryption scheme. However, such a construction work only if the cryptosystem is $\varGamma$-uniform.

Definition 24 ($\varGamma$-uniformity). *Let* Enc *be an asymmetric encryption scheme, with key generation algorithm* $\mathsf{Gen}(1^\lambda)$ *and encryption algorithm* $\mathsf{Enc}(K_p, m; r)$ *over the message space $\mathcal{M}$ and the random coins space $\mathcal{R}$.* Enc *is* $\varGamma$-uniform *if for any plaintext $m \in \mathcal{M}$, for any keys drawn by* Gen *and for any $y \in \{0,1\}^*$, we have*

$$\Pr\left[h \xleftarrow{U} \mathcal{R} : y = \mathsf{Enc}(K_p, m; h)\right] \leq \varGamma \ ,$$

i.e., the probability that a plaintext and a ciphertext match is bounded.

Lemma 25. *HELEN is $(1-p)^n$-uniform.*

Proof. Recall that the HELEN encryption of b is $y = b \cdot c \oplus r_1 G \oplus \mathrm{S}_p^n(r_2)$, for random coins r_1 and r_2. We need to bound the probability (taken over r_1 and r_2) that a given plaintext x and ciphertext y match. Since in HELEN we consider only $p < \frac{1}{2}$, the most probable ciphertext corresponds to $y = b \cdot c \oplus r_1 G$, i.e., when S_p^n is the zero bitstring. This happens with probability $(1-p)^n$. When we take the average over the possible r_1, this probability can only decrease. Hence, HELEN is $(1-p)^n$-uniform. □

Theorem 26. *Let q_1(resp. q_2) be the number of queries an adversary makes to H_1 (resp. H_2). Let q_d be the number of queries performed to the decryption oracle. Then, if HELEN is (t, ϵ)-*$\mathsf{IND\text{-}CPA}$*-secure, the Fujisaki-Okamoto hybrid construction using a one-time pad for symmetric encryption with key length ℓ is (t_1, ϵ_1)-*$\mathsf{IND\text{-}CCA}$*-secure in the random oracle model, where*

$$\begin{aligned} t_1 &\coloneqq t - O\left((q_1 + q_2) \times (k + \ell)\right) \\ \epsilon_1 &\coloneqq (2(q_1 + q_2)\epsilon + 1)(1 - (1-p)^n - 2^{-\ell})^{-q_d} - 1 \ . \end{aligned}$$

Proof. Since HELEN is $\mathsf{OW\text{-}CPA}$ secure and $(1-p)^n$-uniform, the result follows from [26, Theorem 14]. □

7 Conclusion

Further Work. HELEN can be extended in multiple ways. A first idea is to use different $\mathcal{H}$ to reduce the probability of error and, hence, to reduce the transmission overhead. This implies also to verify that Assumption 20 holds for this new $\mathcal{H}$. Another idea would be to encrypt a message in $\mathbb{F}_q$ for $q > 2$. The codes C_1 and C_2 described in Section 3 need then to be modified accordingly as well as the noise we add. This new extension could then be linked to the learning with error (LWE) problem [53], a generalization of the LPN problem over a finite field $\mathbb{F}_q$. Finally, the LPN problem deserves some more analysis in particular when p is not fixed.

In conclusion, HELEN is a code-based public-key cryptosystem based on the hardness of some well-known problems. Since its margin of progression is still large, HELEN can become a competitive cryptosystem with truly practical parameters.

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 [56], 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. Becker, A., Joux, A., May, A., Meurer, A.: Decoding Random Binary Linear Codes in $2^{n/20}$: How $1 + 1 = 0$ Improves Information Set Decoding. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT. Lecture Notes in Computer Science, vol. 7237, pp. 520–536. Springer (2012)
5. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation (Full Version) (1997), available at `http://cseweb.ucsd.edu/users/mihir`
6. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A Concrete Security Treatment of Symmetric Encryption (Extended Abstract). In: FOCS. pp. 394–403 (1997)
7. Bernstein, D.J.: Introduction to post-quantum cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography, pp. 1–14. Springer (2009)
8. Bernstein, D.J., Lange, T., Peters, C.: Attacking and Defending the McEliece Cryptosystem. In: Buchmann, J., Ding, J. (eds.) PQCrypto. Lecture Notes in Computer Science, vol. 5299, pp. 31–46. Springer (2008)
9. 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)
10. Blum, A., Furst, M.L., Kearns, M.J., Lipton, R.J.: Cryptographic Primitives Based on Hard Learning Problems. In: Stinson, D.R. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 773, pp. 278–291. Springer (1993)
11. Blum, A., Kalai, A., Wasserman, H.: Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model. J. ACM 50(4), 506–519 (2003)

12. Bringer, J., Chabanne, H., Dottax, E.: HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks. In: SecPerU. pp. 28–33. IEEE Computer Society (2006)
13. 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)
14. 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)
15. 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)
16. Chekuri, C., Jansen, K., Rolim, J.D.P., Trevisan, L. (eds.): Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th International Workshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings, Lecture Notes in Computer Science, vol. 3624. Springer (2005)
17. Dolev, D., Dwork, C., Naor, M.: Non-Malleable Cryptography (Extended Abstract). In: Koutsougeras, C., Vitter, J.S. (eds.) STOC. pp. 542–552. ACM (1991)
18. Döttling, N., Müller-Quade, J., Nascimento, A.C.A.: IND-CCA Secure Cryptography Based on a Variant of the LPN Problem. In: Wang, X., Sako, K. (eds.) ASIACRYPT. Lecture Notes in Computer Science, vol. 7658, pp. 485–503. Springer (2012)
19. Duc, A., Vaudenay, S.: HELEN: a Public-key Cryptosystem Based on the LPN and the Decisional Minimal Distance Problems (Extended Abstract). In: Yet Another Conference on Cryptography (2012)
20. Duc, A., Vaudenay, S.: TCHo: A Code-Based Cryptosystem. In: Kranakis, E. (ed.) Advances in Network Analysis and its Applications, Mathematics in Industry, vol. 18, pp. 149–179. Springer Berlin Heidelberg (2013)
21. Duc, D.N., Kim, K.: Securing HB^{+} against GRS man-in-the-middle attack. In: Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security (2007)
22. El Gamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In: CRYPTO. pp. 10–18 (1984)
23. Finiasz, M., Sendrier, N.: Security Bounds for the Design of Code-Based Cryptosystems. In: Matsui [44], pp. 88–105
24. 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)
25. 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)
26. 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)
27. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) STOC. pp. 197–206. ACM (2008)

28. Gilbert, H., Robshaw, M., Sibert, H.: Active attack against HB$^+$: a provably secure lightweight authentication protocol. Electronics Letters 41(21), 1169–1170 (2005)
29. Gilbert, H., Robshaw, M.J.B., Seurin, Y.: Good Variants of HB$^+$ Are Hard to Find. In: Tsudik, G. (ed.) Financial Cryptography. Lecture Notes in Computer Science, vol. 5143, pp. 156–170. Springer (2008)
30. Gilbert, H., Robshaw, M.J.B., Seurin, Y.: HB$^\#$: Increasing the Security and Efficiency of HB$^+$. In: Smart, N.P. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 4965, pp. 361–378. Springer (2008)
31. 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)
32. Heyse, S., Kiltz, E., Lyubashesvky, V., Paar, C., Pietrzak, K.: An Efficient Authentication Protocol Based on Ring-LPN. ECRYPT Workshop on Lightweight Cryptography 2007 (2011), http://www.uclouvain.be/crypto/ecrypt_lc11/static/pre_proceedings_2.pdf
33. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A Ring-Based Public Key Cryptosystem. In: Buhler, J. (ed.) ANTS. Lecture Notes in Computer Science, vol. 1423, pp. 267–288. Springer (1998)
34. Hopper, N.J., Blum, M.: Secure Human Identification Protocols. In: Boyd, C. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 2248, pp. 52–66. Springer (2001)
35. Johnson, D.S.: The NP-Completeness Column: An Ongoing Guide. J. Algorithms 3(2), 182–195 (1982)
36. Juels, A., Weis, S.A.: Authenticating Pervasive Devices with Human Protocols. In: Shoup, V. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 3621, pp. 293–308. Springer (2005)
37. 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)
38. Kiltz, E., Pietrzak, K., Cash, D., Jain, A., Venturi, D.: Efficient Authentication from Hard Learning Problems. In: Paterson, K.G. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 6632, pp. 7–26. Springer (2011)
39. Lee, P.J., Brickell, E.F.: An Observation on the Security of McEliece's Public-Key Cryptosystem. In: EUROCRYPT. pp. 275–280 (1988)
40. 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)
41. Lindner, R., Peikert, C.: Better Key Sizes (and Attacks) for LWE-Based Encryption. In: Kiayias, A. (ed.) CT-RSA. Lecture Notes in Computer Science, vol. 6558, pp. 319–339. Springer (2011)
42. Lyubashevsky, V.: The Parity Problem in the Presence of Noise, Decoding Random Linear Codes, and the Subset Sum Problem. In: Chekuri et al. [16], pp. 378–389
43. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: Gilbert, H. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 6110, pp. 1–23. Springer (2010)
44. 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)

45. May, A., Meurer, A., Thomae, E.: Decoding Random Linear Codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT. Lecture Notes in Computer Science, vol. 7073, pp. 107–124. Springer (2011)
46. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. DSN progress report 42(44), 114–116 (1978)
47. Munilla, J., Peinado, A.: HB-MP: A further step in the HB-family of lightweight authentication protocols. Computer Networks 51(9), 2262–2267 (2007)
48. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory 15(2), 159–166 (1986)
49. Ouafi, K., Overbeck, R., Vaudenay, S.: On the Security of HB$^{\#}$ against a Man-in-the-Middle Attack. In: Pieprzyk, J. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 5350, pp. 108–124. Springer (2008)
50. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher, M. (ed.) STOC. pp. 333–342. ACM (2009)
51. Peters, C.: Information-Set Decoding for Linear Codes over F_q. In: Sendrier, N. (ed.) PQCrypto. Lecture Notes in Computer Science, vol. 6061, pp. 81–94. Springer (2010)
52. Rabin, M.: Digitalized signatures and public-key functions as intractable as factorization (1979)
53. 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)
54. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
55. Rosen, A., Segev, G.: Chosen-Ciphertext Security via Correlated Products. In: Reingold, O. (ed.) TCC. Lecture Notes in Computer Science, vol. 5444, pp. 419–436. Springer (2009)
56. 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)
57. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient Public Key Encryption Based on Ideal Lattices. In: Matsui [44], pp. 617–635
58. 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)
59. Vardy, A.: The Intractability of Computing the Minimum Distance of a Code. IEEE Transactions on Information Theory 43(6), 1757–1766 (1997)

A Proof of Theorem 23

Proof. Let $\mathcal{A} \coloneqq (\mathcal{A}_1, \mathcal{A}_2)$ be an $\mathsf{IND\text{-}CPA}$ adversary HELEN with parameter μ, κ. Given $i \in \{1, \ldots, \mu\}$, we define $\mathcal{B}_i \coloneqq (\mathcal{B}_{i,1}(G), \mathcal{B}_{i,2}(G, c))$ as follows.

$\mathcal{B}_{i,1}(G)$:

1. Let $m_0, m_1 \leftarrow \mathcal{A}_1(G)$
2. Let $b_1^0 \| \ldots \| b_\mu^0 \leftarrow \mathsf{Encode}(m_0)$, the encoding of m_0
3. Let $b_1^1 \| \ldots \| b_\mu^1 \leftarrow \mathsf{Encode}(m_1)$, the encoding of m_1
4. Return b_i^0, b_i^1.

$\mathcal{B}_{i,2}(G, c)$:

```
1. Compute $c_1 \leftarrow \mathsf{BEnc}(G, b_1^1), \ldots, c_{i-1} \leftarrow \mathsf{BEnc}(G, b_{i-1}^1)$.
2. Let $c_i = c$
3. Compute $c_{i+1} \leftarrow \mathsf{BEnc}(G, b_{i+1}^0), \ldots, c_\mu \leftarrow \mathsf{BEnc}(G, b_\mu^0)$.
4. Set $y \coloneqq c_1 \| \ldots \| c_\mu$
5. return $\mathcal{A}_2(G, y)$
```

We know that $\mathrm{Adv}\,\mathcal{B}_i \leq \varepsilon_b$. We have

$$\Pr\left[\mathcal{A} \to 0 \mid m_0 \text{ encrypted}\right] = \Pr\left[\mathcal{B}_1 \to 0 \mid b_1^0 \text{ encrypted}\right]$$

and

$$\Pr\left[\mathcal{A} \to 0 \mid m_1 \text{ encrypted}\right] = \Pr\left[\mathcal{B}_\mu \to 0 \mid b_\mu^1 \text{ encrypted}\right] \ .$$

Also,

$$\Pr\left[\mathcal{B}_i \to 0 \mid b_i^1 \text{ encrypted}\right] = \Pr\left[\mathcal{B}_{i+1} \to 0 \mid b_{i+1}^0 \text{ encrypted}\right] \ .$$

Hence,

$$\begin{aligned} \mathrm{Adv}\,\mathcal{A} &= \left(\Pr\left[\mathcal{A} \to 0 \mid m_0 \text{ encrypted}\right] - \Pr\left[\mathcal{A} \to 0 \mid m_1 \text{ encrypted}\right]\right) \\ &= \sum_{i=1}^{\mu} \left(\Pr\left[\mathcal{B} \to 0 \mid b_i^0 \text{ encrypted}\right] - \Pr\left[\mathcal{B} \to 0 \mid b_i^1 \text{ encrypted}\right]\right) \leq \mu\varepsilon_b \ . \end{aligned}$$

□