Semantic Security for the McEliece Cryptosystem without Random Oracles

Ryo Nojima[1], Hideki Imai[23], Kazukuni Kobara[3], and Kirill Morozov[3]

[1] National Institute of Information and Communications Technology (NICT), Japan
[2] Department of Electrical, Electronic and Communication Engineering,
Chuo University, Japan
[3] National Institute of Advanced Industrial Science and Technology (AIST), Japan
ryo-no@nict.go.jp,{h-imai,k-kobara,kirill.morozov}@aist.go.jp

Abstract. In this paper, we formally prove that padding the plaintext with a random bit-string provides the semantic security against chosen plaintext attack (IND-CPA) for the McEliece (and its dual, the Niederreiter) cryptosystems under the standard assumptions.
Such padding has recently been used by Suzuki, Imai and Kobara in the context of RFID security. Our proof relies on the technical result by Katz and Shin from Eurocrypt '05 showing "pseudorandomness" implied by learning parity checks with noise (LPN) problem.
We do not need the random oracles as opposed to the known generic conversions, while they provide stronger protection as compared to our scheme – against (adaptive) chosen ciphertext attack, i.e., IND-CCA(2).
In order to show that the padded version of the cryptosystem remains practical, we provide the estimates for suitable key size together with corresponding work required for successful attack.

1 Introduction

The *semantic security* (a.k.a. *indistinguishability*) defined by Goldwasser and Micali [14] is the security notion for a public-key cryptosystem (PKC) whose intuitive meaning is that a ciphertext does not leak any useful information about the plaintext but its length. For example, even if an attacker knows that the plaintext is either "0" or "1", the ciphertext does not help him almost at all. Since this notion appeared, a number of semantically secure public-key encryption schemes have been proposed [9, 1, 8, 24].

At the same time, the problem of enhancing the existing (not semantically secure) cryptosystems with such useful property also arose. Two examples of such schemes are the *McEliece* [22] and the *Niederreiter* [23] cryptosystems whose security is ensured under the following two assumptions: a) hardness of the

The first author's work has been done when he was at the University of Tokyo, Japan.

bounded distance decoding of random binary linear codes[†] or, equivalently, the *learning parity with noise (LPN)* and b) indistinguishability of the scrambled generating and parity-check matrices of a Goppa code from random ones.[‡] From the security point of view, these cryptosystems has a one-wayness property. Informally, this means that given a randomly chosen ciphertext, it is hard to completely recover the corresponding plaintext.

Motivation: The main motivation to continue research on the McEliece-style cryptosystems is the following: a) As it was pointed out in the original paper [22], the hardware implementation of the McEliece PKC would be very fast as it only requires matrix operations for encryption/decryption (as long as one can afford storing keys of hundreds of kilobytes in size); b) Not only a public-key encryption but also the other primitives (e.g., signatures [7], identity-based identification and signature schemes [6]) can be built based on the McEliece-style assumptions; c) this PKC is secure against quantum adversaries that makes it a good candidate for the post-quantum world.

Our contribution: Our main observation is that if some fixed part of the plaintext is made random then due to the construction of the cryptosystem it makes the ciphertext pseudorandom from the attacker's point of view. As easy as it looks, this fact, to the best of the authors' knowledge, has not been proved or even stated explicitly in the related literature. The paper fills this gap by providing the formal proof of this fact, establishing connections to the adjacent areas of cryptography and discussing the future research directions which this result invokes. Additionally, we estimate the time-complexity of breaking this version of the McEliece PKC (which we call the *randomized McEliece cryptosystem*) and show the suitable size of a public-key for the practical use in this paper.

A bit more formally, let $E_{pk}(\cdot)$ be an encryption algorithm of the McEliece (or the Niederreiter) cryptosystem whose message space is $\{0,1\}^k$, $m \in \{0,1\}^{k_2}$ a message, and $r \in \{0,1\}^{k_1}$ a random sequence, where $k = k_1 + k_2$. Then, the ciphertext corresponding to m becomes $E_{pk}([r|m])$, where $[A|B]$ denotes a concatenation of two vectors (or, in general, matrices) A and B.

In other words, we show that this padding yields an encryption secure under chosen plaintext attack (IND-CPA), if the McEliece (or the Niederreiter) cryptosystem is used, under the standard assumptions.

Some details: We note that the aforementioned scheme perhaps appear implicitly or explicitly in many previous works. This paper was inspired by the work of Suzuki, Kobara and Imai [30] where it was suggested (without a formal proof) for increasing the security of encryption.

[†] So far, there exists no polynomial algorithm for this problem. Some evidence for its hardness is provided by the fact that the general decoding problem is NP-complete [3].

[‡] This has been believed to be true for a long time and was also utilized for cryptographic applications, e.g., [7, 6].

The technical tool which we use to prove the security of our scheme is the technical lemma by Katz and Shin [15] which established a pseudorandomness of the queries to the oracle in the LPN problem. The key difference from their setting is that we have a scrambled generating (or parity-check) matrix of the Goppa code (which is assumed to be pseudorandom – instead of the oracle which is equivalent to a random matrix). The main technical result of our work, Lemma 4, states that substituting a random matrix by a pseudorandom one preserves the pseudorandomness of the output. Then, under the above assumptions the proof of Proposition 1 stating semantic security of the McEliece cryptosystem with randomized plaintext follows as well as the similar result for the Niederreiter cryptosystem.

Related works: Regarding the conversions from one-way cryptosystems to semantically secure ones, one must first mention the straightforward application of Goldreich-Levin (hardcore) predicate theorem [13] or Yao's XOR lemma which would immediately imply the needed result. The obvious problem is that such conversion is quite inefficient.

The list of more elaborated conversions includes (but is not limited to) [2, 11, 26, 18]. The optimal asymmetric encryption padding (OAEP) by Bellare and Rogaway [2] is the first result of such kind but it dealt with one-way trapdoor permutations (while the cryptosystems we consider are only the trapdoor functions) and needed some fixing in the general case [28].

Fujisaki and Okamoto [11] and Pointcheval [26] independently suggested a conversion from any one-way PKC to a PKC semantically secure against chosen ciphertext attack (IND-CCA2). Finally, Kobara and Imai [18] presented a more efficient conversion than the above two, tailored specifically for the McEliece cryptosystem and arming the latter with the semantic security against adaptively-chosen ciphertext attack (IND-CCA2). We emphasize that all the proofs of security for all the above mentioned conversions were in the random oracle model, while our result does not need this assumption.

Organization of the rest of the paper: In Section 2, we provide some basic notation and definitions, and describe the original versions of the PKC's in question. In Section 3, their randomized versions are introduced along with related security definitions and the main result is stated, while its proof is presented in Section 4. In Section 5, the security parameters for the randomized McEliece cryptosystem are estimated. In Section 6, we conclude our work and discuss open questions.

2 Preliminaries

In this paper, we consider a w-error correcting (n, k)-linear binary code and, throughout this paper, we regard k, n, and w as security parameters. Especially, the code we concentrate on is the binary Goppa code and the relationships between these parameters are $n = 2^m$ and $k \geq n - mw$ for every positive integer m. We denote the probabilistic polynomial-time as PPT and we often call the algorithm *efficient* if its running time is polynomial. Let $s \xleftarrow{\$} S$ denote the operation of selecting s uniformly at random from the set S. If $\mathcal{D}$ is a probability distribution over S then $s \leftarrow \mathcal{D}$ denotes the operation of selecting s at random according to $\mathcal{D}$. Let $\mathcal{U}_n$ denote the uniform distribution over $\{0,1\}^n$. Let $\mathcal{U}_{r,c}$ be the uniform distribution over $r \times c$ random binary matrices and let $\mathcal{E}_{n,w}$ be the uniform distribution over $\{0,1\}^n$ of Hamming weight w.

A public-key encryption scheme is composed of a triplet of algorithms $\Pi = (\mathsf{Gen}_\Pi, \mathsf{Enc}_\Pi, \mathsf{Dec}_\Pi)$. The key generation algorithm Gen_Π is a PPT algorithm which on input 1^k ($k \in \mathbb{N}$) outputs a pair of public and secret keys, (pk, sk), in polynomial time. We assume that the public-key pk defines a message space denoted by M. The encryption algorithm Enc_Π is a PPT algorithm which, on input pk and a plaintext $m \in M$, outputs a ciphertext $c \in \{0,1\}^*$. The decryption algorithm Dec_Π is a polynomial-time algorithm which takes sk and c as input and outputs a message m. We require that for any key pair (pk, sk) obtained from Gen_Π, and any plaintext $m \in M$, $\mathsf{Dec}_\Pi(sk, \mathsf{Enc}_\Pi(pk, m)) = m$.

The semantic security against chosen-plaintext attack (IND-CPA) is one of the most natural practical requirements for a public-key cryptosystem. Its intuitive meaning is that a ciphertext does not leak any useful information about the plaintext but its length.

Let $\Pi = (\mathsf{Gen}_\Pi, \mathsf{Enc}_\Pi, \mathsf{Dec}_\Pi)$ be a public-key encryption scheme and let $D = (D_1, D_2)$ be a PPT algorithm. For every $k \in \mathbb{N}$, we define

$$\mathtt{Adv}^{\mathrm{sem}}_{D,\Pi}(k) = \Pr\left[\begin{array}{c|c} (pk, sk) \leftarrow \mathsf{Gen}_\Pi(1^k), & \\ (m_0, m_1) \leftarrow D_1(pk), & D_2(y) = b \\ b \xleftarrow{\$} \{0,1\}, & \\ y \leftarrow \mathsf{Enc}_\Pi(pk, m_b) & \end{array}\right] - \frac{1}{2}.$$

Also we define the advantage function of the scheme as follows. For any t,

$$\mathtt{Adv}^{\mathrm{sem}}_{\Pi}(k, t) = \max_D \left\{\mathtt{Adv}^{\mathrm{sem}}_{D,\Pi}(k)\right\},$$

where the maximum is over all A with time-complexity t. We say that Π is semantically secure if the function $\mathrm{Adv}^{\mathrm{sem}}_{\Pi}(k, t)$ is negligible for every polynomial bounded t and every sufficiently large k.

Let us now describe the original cryptosystems to be considered in this work.

2.1 McEliece Public-Key Cryptosystem

The McEliece cryptosystem [22] consists of a triplet of probabilistic algorithms $\mathrm{ME} = (\mathsf{Gen}_{\mathrm{ME}}, \mathsf{Enc}_{\mathrm{ME}}, \mathsf{Dec}_{\mathrm{ME}})$ and $M = \{0,1\}^k$.

- Key generation algorithm: The PPT key generation algorithm $\mathsf{Gen}_{\mathrm{ME}}$ works as follows:
  1. Generate a $k \times n$ generator matrix $\mathbf{G}'$ of a binary Goppa code, where we assume that there is an efficient error-correction algorithm $\mathsf{Correct}$ which can always correct up to w errors.

  2. Generate a $k \times k$ random non-singular matrix $\mathbf{S}$.
  3. Generate a $n \times n$ random permutation matrix $\mathbf{P}$.
  4. Set $\mathbf{G} = \mathbf{S}\mathbf{G}'\mathbf{P}$, and output $pk = (\mathbf{G}, w)$ and $sk = (\mathbf{S}, \mathbf{G}', \mathbf{P})$.
- The encryption algorithm: The PPT encryption algorithm $\mathsf{Enc}_{\mathrm{ME}}$ takes a plaintext $m \in \{0,1\}^k$ and the public-key pk as input and outputs ciphertext $c = m\mathbf{G} \oplus e$, where $e \leftarrow \mathcal{E}_{n,w}$.
- The decryption algorithm: Given ciphertext c and secret-key sk, the polynomial-time decryption algorithm $\mathsf{Dec}_{\mathrm{ME}}$ works as follows:
  1. Compute $c\mathbf{P}^{-1} = (m\mathbf{S})\mathbf{G}' \oplus e\mathbf{P}^{-1}$, where $\mathbf{P}^{-1}$ denotes the inverse matrix of $\mathbf{P}$.
  2. Compute $m\mathbf{S} = \mathsf{Correct}(c\mathbf{P}^{-1})$.
  3. Output $m = (m\mathbf{S})\mathbf{S}^{-1}$.

2.2 The Niederreiter Public-Key Cryptosystem

Niederreiter [23] proposed a dual version of the McEliece cryptosystem where the public-key is a scrambled parity-check matrix $\mathbf{H}$, a plaintext is $m \in \{0,1\}^n$ of weight w, and the corresponding ciphertext c is of the form $c = m\mathbf{H}$.

The Niederreiter cryptosystem consists of three PPT algorithms NR =($\mathsf{Gen}_{\mathrm{NR}}$, $\mathsf{Enc}_{\mathrm{NR}}$, $\mathsf{Dec}_{\mathrm{NR}}$) and $M = \subset \{0,1\}^n$ is a set of strings of weight w.

- Key generation algorithm: The PPT key generation algorithm $\mathsf{Gen}_{\mathrm{NR}}$ works as follows:
  1. Generate a $(n-k) \times n$ parity check matrix $\mathbf{H}''$ of a binary Goppa code, where we assume that there is an efficient error correcting algorithm $\mathsf{Correct}$ which can correct up to w errors.
  2. Generate $(n-k) \times (n-k)$ random non-singular matrix $\mathbf{S}$.
  3. Generate $n \times n$ random permutation matrix $\mathbf{P}$.
  4. Let $\mathbf{H}' = \mathbf{S}\mathbf{H}''\mathbf{P}$, let $\mathbf{H} = \mathbf{H}'^T$ and output $pk = (\mathbf{H}, w)$ and $sk = (\mathbf{S}, \mathbf{H}'', \mathbf{P})$.
- The encryption algorithm: The polynomial-time encryption algorithm $\mathsf{Enc}_{\mathrm{NR}}$ takes a plaintext $m \in \{0,1\}^n$ of weight w and pk as input and outputs ciphertext $c = m\mathbf{H}$.
- The decryption algorithm: Given ciphertext c and secret-key sk, the polynomial-time decryption algorithm $\mathsf{Dec}_{\mathrm{NR}}$ works as follows:
  1. Compute $\mathbf{S}^{-1}c^T = \mathbf{H}''(\mathbf{P}m^T)$, where S^{-1} denotes the inverse matrix of S
  2. Compute $\mathbf{P}m^T = \mathsf{Correct}(\mathbf{S}^{-1}c^T)$.
  3. Output $m^T = \mathbf{P}^{-1}(\mathbf{P}m^T)$.

3 Randomized Versions and Main Result

3.1 Randomized McEliece Cryptosystem

It is easy to see that the original McEliece cryptosystem [22] is not IND-CPA. Suppose that the adversary obtains a ciphertext c, and he knows that c is a ciphertext of either m_0 or m_1, then he can verify which one is a corresponding plaintext by simply computing the weight of $m_0\mathbf{G} \oplus c$ and check it to be w or not. An intuitive way to avoid such the situation is concatenating a random sequence r to a message m and encrypting $[r|m]$. Such padding has been often employed in the previous schemes, but so far there has been no formal proof for semantic security which it provides.

Let $k_1, k_2 \in \mathbb{N}$ be two integers such that $k = k_1 + k_2$ and $k_1 = bk$, where $b < 1$ is a positive rational number, e.g., $b = \frac{9}{10}$. Here, we denote by k_1 the length of the random string r and by k_2 the length of the message m. The encryption algorithm $\mathsf{Enc}_{\mathrm{RME}}$ just encrypts $[r|m]$ instead of m itself. The decryption algorithm $\mathsf{Dec}_{\mathrm{RME}}$ is almost the same as $\mathsf{Dec}_{\mathrm{ME}}$. The difference is that it outputs only the last k_2 bits of the decrypted string.

3.2 Randomized Niederreiter Cryptosystem

Similar situation occurs in the Niederreiter cryptosystem as well. In [30], the authors proposed the RFID authentication scheme based on the Niederreiter cryptosystem. Their idea was essentially to use the random padding for enhancing security of the Niederreiter cryptosystem. However, no claim of semantic security for this scheme have been made.

Let n_1, and n_2 be some integers with $n = n_1 + n_2$ and $n_1 = bn$ for some positive rational number b, e.g., $b = \frac{9}{10}$. Here we assume that $r \in \{0,1\}^{n_1}$ is the random sting of weight $w_1 = \lceil \frac{n_1 w}{n_1 + n_2} \rceil$ and $m \in \{0,1\}^{n_2}$ is the message of weight $w_2 = \lfloor \frac{n_2 w}{n_1 + n_2} \rfloor$. The encryption algorithm $\mathsf{Enc}_{\mathrm{RNR}}$ encrypts $[r|m]$ where r is randomly chosen. Also the decryption algorithm $\mathsf{Dec}_{\mathrm{RNR}}$ is the same as $\mathsf{Dec}_{\mathrm{NR}}$ except that it outputs only the last n_2 bits of the decrypted plaintext.

3.3 Security of the Original Cryptosystems

In order to prove the security of these schemes, we use the same assumptions as for the original PKC.

Generally, we can categorize the attacks to the McEliece and the Niederreiter cryptosystems into the following two cases:

Structural Attack: Recover the original structure of the secret key from the scrambled generator matrix $\mathbf{G}$ or the scrambled parity check matrix $\mathbf{H}$.

Direct Decoding: Decode the plaintext m directly from $m\mathbf{G} \oplus e$ or $m\mathbf{H}$.

If we employ Goppa codes on $\mathbb{F}_2$ from codes on $\mathbb{F}_{2^m}$ then there is no efficient algorithm which can extract the secret-key from the public key in the McEliece or the Niederreiter cryptosystems as long the weak keys [20] are avoided. Moreover, there is no algorithm which can efficiently distinguish the matrices defined by the public-keys of the those cryptosystems and the same size random matrices. The time complexity of the currently best algorithm [7] is still super-polynomial. Intuitively this algorithm works as follows: enumerate Goppa polynomials and verify whether each corresponding code and the generator matrix $\mathbf{G}$ (or the generator matrix converted from parity check matrix $\mathbf{H}$) are "permutation equivalent" or not by using the *support splitting algorithm* [27], which results in a $n^w(1+o(1))$-time algorithm. Actually, in the worst-case, the problem of deciding permutation equivalence can reduce to the graph isomorphism problem [25]. To prove security of the randomized cryptosystems, we assume that the matrices $\mathbf{G}$ and $\mathbf{H}$ are indistinguishable from the same size random matrices, respectively, for any PPT algorithm. The formal statements are given in Subsections 3.4 and 3.5.

For the excellent review of the security of both PKC's, we refer the reader to [17].

3.4 Security of the Randomized McEliece Cryptosystem

Definition 1 (Indistinguishability of G). *Let D be a probabilistic algorithm. For every $k \in \mathbb{N}$, we define*

$$\begin{aligned}\mathtt{Adv}^{\mathrm{ind}}_{D,\mathbf{G}}(k) = \Pr\left[((\mathbf{G}, w), sk) \leftarrow \mathsf{Gen}_{\mathrm{ME}}(1^k) \mid D(\mathbf{G}, w) = 1\right] \\ - \Pr\left[\mathbf{R} \leftarrow \mathcal{U}_{k,n} \mid D(\mathbf{R}, w) = 1\right].\end{aligned}$$

Also we define the advantage function of the problem as follows. For any t,

$$\mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k,t) = \max_D \left\{\mathtt{Adv}^{\mathrm{ind}}_{D,\mathbf{G}}(k)\right\},$$

where the maximum is over all D with time-complexity t. We say $\mathbf{G}$ is indistinguishable *if, for every polynomial bounded t and every sufficiently large k, $\mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k,t)$ is negligible.*

In this paper, we assume that $\mathbf{G}$ is indistinguishable. This assumption was also utilized in [7, 6].

To prove the security, we also need to assume the learning parity with noise (LPN) problem is hard.

Definition 2 (LPN problem). *Let r, a be binary vectors of length k and let $z = \langle r, a \rangle$, where $\langle r, a \rangle$ is the dot product of r and a modulo 2. Also we consider Bernoulli distribution $\mathcal{B}_\theta$ with parameter $\theta \in (0, \frac{1}{2})$, and let $\mathcal{Q}_{r,\theta}$ be the distribution defined by*

$$\left\{a \leftarrow \{0,1\}^k, \nu \leftarrow \mathcal{B}_\theta \mid (a, \langle r, a \rangle \oplus \nu)\right\}.$$

Let A be a probabilistic algorithm. For every $k \in \mathbb{N}$, we define

$$\mathtt{Adv}^{\mathrm{oneway}}_{A,\mathrm{LPN}_\theta}(k) = \Pr\left[r \leftarrow \{0,1\}^k \mid A^{\mathcal{Q}_{r,\theta}} = r\right].$$

We define the advantage function of the problem as follows. For any t and q,

$$\mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k,t,q) = \max_A \left\{\mathtt{Adv}^{\mathrm{oneway}}_{A,\mathrm{LPN}_\theta}(k)\right\},$$

where the maximum is over all A with time-complexity t and query-complexity q. We say that the LPN$_\theta$ problem is hard *if $\mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k,t,q)$ is negligible for every sufficiently large k and polynomially bounded t, and q.*

We assume that the LPN$_\theta$ problem is hard for some θ satisfying $w = \lfloor \theta(n+1) \rfloor$. In fact, all known algorithms for solving this problem are still super-polynomial time [4]. Especially, for fixed q and small amount of noise, the best ones are the information set decoding attacks due to Leon [19], Stern [29], Canteaut and Chabaud [5], and its time complexity is roughly

$$\binom{n}{k} \cdot \binom{n-w}{k}^{-1}, \tag{1}$$

where w is the weight of the noise.

With the above two assumptions, we can prove that the randomized McEliece cryptosystem is semantically secure.

Proposition 1. *The randomized McEliece cryptosystem is IND-CPA secure if the LPN$_\theta$ problem is hard and* $\mathbf{G}$ *is indistinguishable.*

The proof is given in Section 4.2

3.5 Security of the Randomized Niederreiter Cryptosystem

Definition 3 (Indistinguishability of H). *Let D be a probabilistic algorithm. For every $k \in \mathbb{N}$, we define*

$$\begin{aligned}\mathtt{Adv}^{\mathrm{ind}}_{D,\mathbf{H}}(k) = \Pr\left[((\mathbf{H},w),sk) \leftarrow \mathsf{Gen}_{\mathrm{NR}}(1^k) \mid D(\mathbf{H},w)=1\right] \\ - \Pr\left[\mathbf{R} \leftarrow \mathcal{U}_{n,n-k} \mid D(\mathbf{R},w)=1\right].\end{aligned}$$

Also we define the advantage function of the problem as follows. For any t,

$$\mathtt{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k,t) = \max_D \left\{\mathtt{Adv}^{\mathrm{ind}}_{D,\mathbf{H}}(k)\right\},$$

where the maximum is over all D with time-complexity t. We say $\mathbf{H}$ *is* indistinguishable *if* $\mathtt{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k,t)$ *is negligible for every polynomially bounded t and every sufficiently large k.*

In this paper, we assume that $\mathbf{H}$ is indistinguishable.

We can prove that the randomized Niederreiter cryptosystem has semantic security if the following problem is hard for every PPT algorithm. The problem is similar to the LPN problem but, to the best of the authors' knowledge, there exists no proof that these two problems are equivalent in terms of the *average case* time-complexity.

Definition 4 (Syndrome Decoding Problem). *Let D be a probabilistic algorithm. For every $k \in \mathbb{N}$, we define*

$$\mathtt{Adv}^{\mathrm{oneway}}_{D,\mathrm{SD}_w}(k) = \Pr\left[\mathbf{H} \leftarrow \mathcal{U}_{n,n-k}, r \leftarrow \mathcal{E}_{n,w} \mid D((\mathbf{H},w), r\mathbf{H}) = r\right].$$

Also we define the advantage function of the problem as follows. For any t,

$$\mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_w}(k,t) = \max_D \left\{\mathtt{Adv}^{\mathrm{oneway}}_{D,\mathrm{SD}_w}(k)\right\},$$

where the maximum is over all D with time-complexity t. We say that the syndrome decoding problem SD_w is hard *if* $\mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_w}(k,t)$ *is negligible for every polynomially bounded t and every sufficiently large k.*

We assume that the SD_w problem is hard. With the above two assumptions we can prove the following proposition.

Proposition 2. *The randomized Niederreiter cryptosystem is IND-CPA secure if the SD_w problem is hard and* $\mathbf{H}$ *is indistinguishable.*

The proof is given in Section 4.3

4 Security Analysis

4.1 Intermediate Lemma

Before describing the proofs of randomized versions being semantically secure, we characterize these cryptosystems.

We denote a set of random numbers utilized inside Enc_{Π} by R, and we explicitly denote the randomness used inside the algorithm by $\mathsf{Enc}_{\Pi}(pk, m; r)$, where $r \in R$.

Definition 5. *The public-key encryption scheme* $\Pi = (\mathsf{Gen}_{\Pi}, \mathsf{Enc}_{\Pi}, \mathsf{Dec}_{\Pi})$ *with a message space M and a random space R is called* admissible *if there is a pair of deterministic polynomial-time algorithms* Enc^1_{Π} *and* Enc^2_{Π} *satisfying the following properties:*

- *Dividability:* Enc^1_{Π} *takes as input a key pk and $r \in R$, and outputs a $p(k)$ bit-string.* Enc^2_{Π} *takes as input a key pk, and $m \in M$ and outputs a $p(k)$ bit-string. Here p is some polynomial in k. Then for any pk given by* Gen, *$r \in R$, and $m \in M$,* $\mathsf{Enc}^1_{\Pi}(pk, r) \oplus \mathsf{Enc}^2_{\Pi}(pk, m) = \mathsf{Enc}_{\Pi}(pk, m; r)$.
- *Pseudorandomness: Let D be a probabilistic algorithm. We define*

$$\mathtt{Adv}^{\mathrm{ind}}_{D,\mathsf{Enc}^1_{\Pi}}(k) = \Pr\left[r \xleftarrow{\$} R, (pk, sk) \leftarrow \mathsf{Gen}(1^k) \mid D(pk, \mathsf{Enc}^1_{\Pi}(pk, r)) = 1\right] - \Pr\left[s \leftarrow \mathcal{U}_{p(k)}, (pk, sk) \leftarrow \mathsf{Gen}(1^k) \mid D(pk, s) = 1\right].$$

Also we define the advantage function of the problem as follows. For any t,

$$\mathtt{Adv}^{\mathrm{ind}}_{\mathsf{Enc}^1_{\Pi}}(k,t) = \max_D \left\{\mathtt{Adv}^{\mathrm{ind}}_{D,\mathsf{Enc}^1_{\Pi}}(k)\right\},$$

where the maximum is over all D with time-complexity t. Then $\mathtt{Adv}^{\mathrm{ind}}_{\mathsf{Enc}^1_{\Pi}}(k,t)$ *is negligible for every polynomially bounded t and every sufficiently large k.*

In the following lemma, we prove that if Π is an admissible cryptosystem, then it is an IND-CPA encryption scheme.

Lemma 1. *If there exists an algorithm D which runs in time t, and such that*

$$\mathtt{Adv}^{\mathrm{sem}}_{D,\Pi}(k,t) \geq \delta,$$

then

$$\mathtt{Adv}^{\mathrm{ind}}_{\mathsf{Enc}^1_\Pi}(k,t+t') \geq \delta,$$

where t' is the worst-case time-complexity of computing Enc^1_Π.

Proof. We construct a distinguisher D' from the IND-CPA adversary D. We show that if D breaks the semantic security with non-negligible probability then D' distinguishes $s_1 = \mathsf{Enc}^1_\Pi(pk,r)$ and the same length random value s_0 with non-negligible probability.

We construct an algorithm D' as follows:

```
$\underline{D'(pk,\tilde{s})}$
Run $D_1(pk)$ to obtain $(m_0,m_1)$
$b \leftarrow \mathcal{U}_1$
Define $c = \tilde{s} \oplus \mathsf{Enc}^2_\Pi(pk,m_b)$
Run $D_2(c)$ to obtain $b'$
Output 1 if $b' = b$, and 0 otherwise
```

Let Rand be the event that $\tilde{s}(= s_0)$ was chosen from the random distribution, and let Real be the event that $\tilde{s}(= s_1)$ is $\mathsf{Enc}^1_\Pi(pk,r)$ for some random string r. We will say that D succeeds if $b' = b$ (and denote this event by Succ) under the event Real occurs, and we denote this probability as $\Pr_D[\mathsf{Succ}]$. Note that, we know

$$\Pr[D' = 1 \mid \mathsf{Real}] - \Pr[D' = 1 \mid \mathsf{Rand}] \tag{2}$$

is upper-bounded by $\mathtt{Adv}^{\mathrm{ind}}_{\mathsf{Enc}^1_\Pi}(k,t+t')$.

We claim that $\Pr[D' = 1 \mid \mathsf{Real}] = \Pr_D[\mathsf{Succ}]$. To see this, note that when Real occurs we have $\tilde{s} = s_1 = \mathsf{Enc}^1_\Pi(pk,r)$. But then s_1 is distributed exactly as they would be in a real execution. Since D' outputs 1 iff D succeeds, the claim follows.

To complete the proof, we show $\Pr[D' = 1 \mid \mathsf{Rand}] = \frac{1}{2}$. Here we know that $\tilde{s}$ is uniformly distributed in $\mathcal{U}_{p(k)}$. Therefore, $\tilde{s} \oplus \mathsf{Enc}^2_\Pi(pk,m_b)$ given to D is uniformly distributed in $\mathcal{U}_{p(k)}$ as well. This means that D obtains no information related to b. Since D' outputs 1 iff D succeeds, we can conclude that $\Pr[D' = 1 \mid \mathsf{Rand}] = \frac{1}{2}$.

By combining these results, now we can estimate (2) as follows:

$$\begin{aligned}\Pr[D' = 1 \mid \mathsf{Real}] - \Pr[D' = 1 \mid \mathsf{Rand}] &= \Pr_D[\mathsf{Succ}] - 1/2\\ &= \mathtt{Adv}^{\mathrm{sem}}_{D,\Pi}(k)\\ &\geq \delta.\end{aligned}$$

Since $\mathtt{Adv}^{\mathrm{ind}}_{\mathsf{Enc}^1_\Pi}(k,t+t') \geq \Pr[D' = 1 \mid \mathsf{Real}] - \Pr[D' = 1 \mid \mathsf{Rand}]$,

$$\mathtt{Adv}^{\mathrm{ind}}_{\mathsf{Enc}^1_\Pi}(k,t+t') \geq \delta.$$

This concludes the proof. □

Therefore, to prove Propositions 1 and 2, it is sufficient to prove that the randomized McEliece and the randomized Niederreiter cryptosystems are admissible.

4.2 Proof of Proposition 1

Let us recall the form of the randomized McEliece cryptosystem: $c = [r|m]\mathbf{G} \oplus e$. Let $\mathbf{G}_1$ and $\mathbf{G}_2$ be $k_1 \times n$ and $k_2 \times n$ sub-matrices of $\mathbf{G}$, respectively, such that $\mathbf{G}^T = [\mathbf{G}_1^T|\mathbf{G}_2^T]$. Then we can rewrite the above equation as follows:

$$c = [r|m]\mathbf{G} \oplus e = \{r\mathbf{G}_1 \oplus e\} \oplus m\mathbf{G}_2. \tag{3}$$

If we define the algorithm $\mathsf{Enc}^1_{\mathrm{RME}}(pk, [r|r']))$ by $r\mathbf{G}_1 \oplus e$, where r' is the random number utilized for generating the weight w random vector $e \in \{0,1\}^n$, and define the algorithm $\mathsf{Enc}^2_{\mathrm{RME}}(pk, m)$ by $m\mathbf{G}_2$, then the randomized McEliece cryptosystem satisfies dividability. So to prove the IND-CPA security of the randomized McEliece cryptosystem, it is sufficient to prove that $\mathsf{Enc}^1_{\mathrm{RME}}$ satisfies the pseudorandomness property.

The following lemma, which states that the hardness of the LPN problem implies pseudorandomness of the output, plays an important role to prove the pseudorandomness of $\mathsf{Enc}^1_{\mathrm{RME}}(pk, r)$.

In the following lemma, we set the length of a and r as k_1.

Lemma 2 (Lemma 1 in [15]). *If there exists an algorithm which runs in time t, makes queries q times and such that*

$$\Pr\left[r \leftarrow \{0,1\}^{k_1} \mid D^{\mathcal{Q}_{r,\theta}} = 1\right] - \Pr\left[D^{\mathcal{U}_{k_1+1}} = 1\right] \geq \delta, \ \textit{then}$$

$\mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k_1, t', q') \geq \delta/4$, *where* $t' = O(tk_1\delta^{-2}\log k_1)$, $q' = O(q\delta^{-2}\log k_1)$.

This is the key technical lemma which was rigorously proved in [15]. The next corollary easily follows from the above lemma.

Corollary 1. *Let $\mathcal{O}_1 = \mathcal{Q}_{r,\theta}$ and $\mathcal{O}_0 = \mathcal{U}_{k_1+1}$. If there exists an algorithm which runs in time t, makes queries q times and such that*

$$\left|\Pr\left[\begin{array}{c} r \leftarrow \mathcal{U}_{k_1}, b \leftarrow \mathcal{U}_1 \\ D^{\mathcal{O}_b} = b' \end{array}\middle| b = b'\right] - \frac{1}{2}\right| \geq \delta$$

then

$$2 \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k_1, t', q') \geq \delta,$$

where $t' = O(tk_1\delta^{-2}\log k_1)$, $q' = O(q\delta^{-2}\log k_1)$.

Informally, the next lemma states that choosing the errors for the LPN problem according to $\mathcal{E}_{n,w}$ (instead of the Bernoulli distribution with parameter Θ) preserves pseudorandomness of the output if $w = \lfloor\theta(n+1)\rfloor$.

Let $\mathbf{R}_1$ and $\mathbf{R}_2$ be a $k_1 \times n$ sub-matrix and a $k_2 \times n$ sub-matrix of a matrix $\mathbf{R}$, respectively, such that $\mathbf{R}^T = [\mathbf{R}_1^T|\mathbf{R}_2^T]$. Also let $q = n$.

Lemma 3. *If there exists an algorithm D which runs in time t and such that*

$$\Pr\left[\begin{array}{c|c} r \leftarrow \mathcal{U}_{k_1}, \mathbf{R} \leftarrow \mathcal{U}_{k,n}, e \leftarrow \mathcal{E}_{n,w}, & D(\mathbf{R}, w, s_b) = b \\ b \leftarrow \mathcal{U}_1, s_0 \leftarrow \mathcal{U}_n, s_1 \leftarrow r\mathbf{R}_1 \oplus e & \end{array}\right] - \frac{1}{2} \geq \delta$$

then

$$2(n+1) \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k_1, t', q') \geq \delta, \tag{4}$$

respectively, where $q' = O(n\delta'^{-2} \log k_1)$*,* $t' = O((t + n^2)k_1\delta'^{-2} \log k_1)$ *and* $\delta' = \frac{\delta}{n+1}$*.*

Sketch of Proof. Let's denote the distinguisher denoted in Corollary 1 by D'. Then we can construct the distinguisher which tells $\mathcal{O}_1$ from $\mathcal{O}_0$ using D as follows:

- D' accesses to the oracle n times. Let (a_i, b_i) be a response from the oracle at time i. If the oracle is $\mathcal{O}_1$ then we denote each error vector by ν_i and so $b_i = \langle a_i, r\rangle \oplus \nu_i$.
- D' sets $\mathbf{R}_1 = [a_1|a_2|\dots|a_n]$, where we regard each a_i as a column vector and thus $\mathbf{R}_1$ is $k_1 \times n$ random matrix.
- D' randomly generates $\mathbf{R}_2 \leftarrow \mathcal{U}_{k_2,n}$.
- D' feeds D with $\mathbf{R}^T = [\mathbf{R}_1^T|\mathbf{R}_2^T]$, w, and $[b_1|b_2|\dots|b_n]$.
- D' outputs what D outputs.

Consider the case where the oracle is $\mathcal{O}_1$. In this case, each error ν_i added by oracle $\mathcal{O}_1$ is generated according to Bernoulli distribution, but D' must feed D with $r\mathbf{G}_1 \oplus e$, where $e = [\nu_1|\nu_2|\dots|\nu_n]$ is a string of weight w. So we must estimate the probability of the weight of e being w. However, this probability is at least $\frac{1}{n+1}$ since the weight of w being $\lfloor\theta(n+1)\rfloor$ is the most likely to occur in Bernoulli distribution among $n+1$ possible weights. This introduces $(n+1)$ in the left part of (4), still leaving the advantage negligible. □

To prove the proposition, we need to replace the random matrix $\mathbf{R}$ with the (pseudorandom) public-key matrix $\mathbf{G}$. The next lemma states that exchanging a truly random matrix with a pseudorandom matrix $\mathbf{G}$ preserves the pseudorandomness of the output $s_1 = r\mathbf{G}_1 \oplus e$. That is, the randomized McEliece cryptosystem is an admissible cryptosystem.

Lemma 4. *If there exists an algorithm D which runs in time t and such that*

$$\begin{aligned} &\Pr\left[r \leftarrow \mathcal{U}_{k_1}, (\mathbf{G}, w) \leftarrow \mathsf{Gen}_{\mathrm{ME}}(1^k), e \leftarrow \mathcal{E}_{n,w} \mid D((\mathbf{G}, w), r\mathbf{G}_1 \oplus e) = 1\right] \\ &\qquad - \Pr\left[s_0 \leftarrow \mathcal{U}_n, (\mathbf{G}, w) \leftarrow \mathsf{Gen}_{\mathrm{ME}}(1^k) \mid D((\mathbf{G}, w), s_0) = 1\right] \geq \delta, \end{aligned}$$

then

$$4(n+1) \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k_1, t_1, q_1) + 2 \cdot \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t_2) \geq \delta.$$

$q_1 = O(n\delta'^{-2} \log k_1)$*,* $t_1 = O((t+n^2)k_1\delta'^{-2} \log k_1)$*,* $t_2 = O(t+n^2)$ *and* $\delta' = \frac{\delta}{n+1}$*.*

Proof. We will say that the algorithm D succeeds iff it outputs 1 when given input was of the form $r\mathbf{G}_1 \oplus e$. We denote this event by Succ. We construct an adversary D' which distinguishes the random matrix $\mathbf{R}$ from the matrix $\mathbf{G}$ as follows.

```
$\underline{D'(\mathbf{M}, w)}$
Divide $\mathbf{M}$ into $\mathbf{M}_1$ and $\mathbf{M}_2$ such that $\mathbf{M}^T = [\mathbf{M}_1^T | \mathbf{M}_2^T]$, $\mathbf{M}_1$ is $k_1 \times n$ sub-matrix
    and $\mathbf{M}_2$ is $k_2 \times n$ sub-matrix.
$b \leftarrow \mathcal{U}_1$
If $b = 1$
    $e \leftarrow \mathcal{E}_{n,w}$, $r \leftarrow \mathcal{U}_{k_1}$, run $D((\mathbf{M}, w), r\mathbf{M}_1 \oplus e)$ to obtain $b'$
Else
    $s_0 \leftarrow \mathcal{U}_n$, run $D((\mathbf{M}, w), s_0)$ to obtain $b'$
Endif
If $b = b'$ then output 1, and otherwise 0
```

Let Rand be the event that the matrix $\mathbf{M}$ was chosen randomly from uniform distribution $\mathcal{U}_{k,n}$, and let Real be the event that the matrix was generated by $\mathsf{Gen}_{\mathrm{ME}}$. Note that we want to estimate the amount of

$$\Pr\left[b = b' \mid \mathsf{Real}\right] - \Pr\left[b = b' \mid \mathsf{Rand}\right].$$

We first claim that $\Pr\left[b = b' \mid \mathsf{Real}\right] = \Pr_D[\mathsf{Succ}]$. To see this, note that when Real occurs we have $\mathbf{M} = \mathbf{G}$. But then $\mathbf{G}$ is distributed exactly as this would be in a real execution, and since D' outputs 1 iff D succeeds, the claim follows.

Next, we estimate the amount of $\Pr\left[b = b' \mid \mathsf{Rand}\right]$. From the construction of D', we can rewrite this by

$$\Pr\left[b = b' \mid \mathsf{Rand}\right] = \Pr\left[\begin{array}{c} \mathbf{M} \leftarrow \mathcal{U}_{k,n}, b \leftarrow \mathcal{U}_1, e \leftarrow \mathcal{E}_{n,w}, r \leftarrow \mathcal{U}_{k_1}, \\ s_0 \leftarrow \mathcal{U}_n, s_1 \leftarrow r\mathbf{M}_1 \oplus e, b' \leftarrow D((\mathbf{M}, w), s_b) \end{array} \middle| b = b' \right].$$

But this is already evaluated in Lemma 3. So we know that

$$2(n+1) \cdot \mathtt{Adv}_{\mathrm{LPN}_\theta}^{\mathrm{oneway}}(k_1, t_1, q_1) \geq \Pr\left[b = b' \mid \mathsf{Rand}\right] - \frac{1}{2},$$

where $q_1 = O(n\delta'^{-2} \log k_1)$, $t_1 = O((t + n^2)k_1\delta'^{-2} \log k_1)$ and $\delta' = \frac{\delta}{n+1}$.

By combining these observations, we obtain

$$\Pr\left[b = b' \mid \mathsf{Real}\right] - \Pr\left[b = b' \mid \mathsf{Rand}\right] \\ \geq \Pr_D[\mathsf{Succ}] - \frac{1}{2} - 2(n+1) \cdot \mathtt{Adv}_{\mathrm{LPN}_\theta}^{\mathrm{oneway}}(k_1, t_1, q_1)$$

and

$$\mathtt{Adv}_{\mathbf{G}}^{\mathrm{ind}}(k, t_2) \geq \Pr\left[b = b' \mid \mathsf{Real}\right] - \Pr\left[b = b' \mid \mathsf{Rand}\right],$$

where $t_2 = O(t + n^2)$. So

$$\mathtt{Adv}_{\mathbf{G}}^{\mathrm{ind}}(k, t_2) + 2(n+1) \cdot \mathtt{Adv}_{\mathrm{LPN}_\theta}^{\mathrm{oneway}}(k_1, t_1, q_1) \geq \Pr_D[\mathsf{Succ}] - \frac{1}{2}$$

and thus

$$2 \cdot \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t_2) + 4(n+1) \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k_1, t_1, q_1) \geq \delta.$$

This concludes the proof. □

Remember the form of the randomized McEliece cryptosystem, that is

$$c = [r|m]\,\mathbf{G} \oplus e = \{r\mathbf{G}_1 \oplus e\} \oplus m\mathbf{G}_2.$$

In the above lemma, we proved that $\mathsf{Enc}^1_{\mathrm{RME}}(pk, r) = r\mathbf{G}_1 \oplus e$ is pseudorandom for every PPT algorithm. Thus, the randomized McEliece cryptosystem is the admissible cryptosystem. By Lemma 1 and Lemma 4, we can conclude with the following: If there exists an IND-CPA adversary D which runs in time t, then

$$2 \cdot \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t_2) + 4(n+1) \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k_1, t_1, q_1) \geq \mathtt{Adv}^{\mathrm{sem}}_{D,\mathrm{RME}}(k, t),$$

where $q_1 = O(n\delta'^{-2} \log k_1)$, $t_1 = O((t+n^2)k_1\delta'^{-2} \log k_1)$, $t_2 = O(t+n^2)$ and $\delta' = \frac{\delta}{n+1}$. Therefore, if $\mathbf{G}$ is indistinguishable and the LPN problem is hard then the randomized McEliece cryptosystem is IND-CPA secure.

4.3 Proof of Proposition 2

In the above proof, Lemma 2 played an important role. There is a similar result in [10] which is useful for proving the semantic security of the randomized Niederreiter cryptosystem. The result stated in [10] is that, for a randomly chosen vector $r \in \{0,1\}^{n_1}$ of weight w_1 and $n_1 \times (n-k)$ binary random matrix $\mathbf{R}$, $r\mathbf{R}$ is pseudorandom. So we can prove its semantic security with the similar strategy. That is, recall the form of the randomized Niederreiter cryptosystem: $c = [r|m]\mathbf{H}$, where r is the random vector of weight w_1. Let $\mathbf{H}_1$ and $\mathbf{H}_2$ be $n_1 \times (n-k)$ and $n_2 \times (n-k)$ sub-matrices of $\mathbf{H}$, respectively, such that $\mathbf{H}^T = [\mathbf{H}_1^T|\mathbf{H}_2^T]$. Similar to the randomized McEliece cryptosystem, we can rewrite the above equation as follows:

$$c = [r|m]\mathbf{H} = \{r\mathbf{H}_1\} \oplus m\mathbf{H}_2.$$

The randomized Niederreiter cryptosystem has dividable property in nature, and thus the rest of the proof is to prove its pseudorandomness of $\mathsf{Enc}^1_{\mathrm{RNR}}(pk, r') = r\mathbf{H}_1$, where r' is the random string for generating a random string $r \in \{0,1\}^{n_1}$ of weight w_1. To prove pseudorandom property we utilize the result of [10][§].

Theorem 1 ([10, 12]). *If there exists an algorithm which runs in time t, and such that*

$$\begin{aligned}\Pr\,[r \leftarrow \mathcal{E}_{n_1,w_1}, \mathbf{R}_1 \leftarrow \mathcal{U}_{n_1,n-k} \mid D((\mathbf{R_1}, w_1), r\mathbf{R}_1) = 1]&\\ - \Pr\,[s \leftarrow \mathcal{U}_{n-k}, \mathbf{R}_1 \leftarrow \mathcal{U}_{n_1,n-k} \mid D((\mathbf{R}_1, w_1), s) = 1] &\geq \delta,\end{aligned}$$

then $\mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_{w_1}}(n_1, t') \geq \frac{\delta^3}{64 n_1}$*, where* $t' = O(n^2(t+n^2)/\delta^2)$.

[§] It utilized the Goldreich-Levin (hardcore) predicate theorem[13, 12] to prove the pseudorandomness but the authors did not estimate the reduction cost. To estimate the reduction cost, we simply combine Proposition 2.5.3 in [12] and Theorem 1 in [10].

Let $\mathbf{R}_1$ be the $n_1 \times (n-k)$ binary matrix, let $\mathbf{R}_2$ be the $n_2 \times (n-k)$ binary matrix, and let $\mathbf{R}^T = [\mathbf{R}_1^T | \mathbf{R}_2^T]$. The following corollary can be easily deduced from the above theorem.

Corollary 2. *If there exists an algorithm which runs in time t, and such that*

$$\begin{aligned}\Pr\left[r \leftarrow \mathcal{E}_{n_1,w_1}, \mathbf{R} \leftarrow \mathcal{U}_{n,n-k} \mid D((\mathbf{R}, w_1), r\mathbf{R}_1) = 1\right] \\ - \Pr\left[s \leftarrow \mathcal{U}_{n-k}, \mathbf{R} \leftarrow \mathcal{U}_{n,n-k} \mid D((\mathbf{R}, w_1), s) = 1\right] \geq \delta,\end{aligned}$$

then $\mathtt{Adv}^{\text{oneway}}_{\text{SD}_{w_1}}(n_1, t') \geq \frac{\delta^3}{64 n_1}$, *where* $t' = O(n^2(t+n^2)/\delta^2)$.

We follow the same strategy as with Proposition 1: We replace the random matrix $\mathbf{R}$ with the (pseudorandom) public-key matrix $\mathbf{H}$ and show $\mathsf{Enc}^1_{\text{RNR}}(pk, r') = r\mathbf{H}_1$ is pseudorandom, where r' is used to produce a random string $r \in \{0,1\}^{n_1}$ of weight w_1. The proof of the following lemma is very similar to that of Lemma 4, so we only provide its sketch.

Lemma 5. *If there exists an algorithm D which runs in time t and such that*

$$\begin{aligned}\Pr\left[r \leftarrow \mathcal{E}_{n_1,w_1}, (\mathbf{H}, w) \leftarrow \mathsf{Gen}_{\text{NR}}(1^k) \mid D((\mathbf{H}, w), r\mathbf{H}_1) = 1\right] \\ - \Pr\left[s \leftarrow \mathcal{U}_{n-k}, (\mathbf{H}, w) \leftarrow \mathsf{Gen}_{\text{NR}}(1^k) \mid D((\mathbf{H}, w), s) = 1\right] \geq \delta,\end{aligned}$$

then

$$8 \cdot \sqrt[3]{n_1 \cdot \mathtt{Adv}^{\text{oneway}}_{\text{SD}_{w_1}}(n_1, t')} + 2 \cdot \mathtt{Adv}^{\text{ind}}_{\mathbf{H}}(k, O(t+n^2)) \geq \delta,$$

where $t' = O(n^2(t+2n^2)/\delta^2)$.

Sketch of Proof. We will say that the algorithm D succeeds iff it outputs 1 when given input was of the form $r\mathbf{H}_1$. We denote this event by Succ. We construct an adversary D' which distinguishes the random matrix $\mathbf{R}$ from the matrix $\mathbf{H}$ as follows.

```
$D'(\mathbf{M}, w)$
Divide $\mathbf{M}$ into $\mathbf{M}_1$ and $\mathbf{M}_2$ such that $\mathbf{M}^T = [\mathbf{M}_1^T|\mathbf{M}_2^T]$, $\mathbf{M}_1$ is a $n_1 \times (n-k)$
    sub-matrix and $\mathbf{M}_2$ is a $n_2 \times (n-k)$ sub-matrix.
$b \leftarrow \mathcal{U}_1$
If $b = 1$
    $r \leftarrow \mathcal{E}_{n_1,w_1}$, set $s_1 = r\mathbf{M}_1$ and run $D((\mathbf{M}, w), s_1)$ to obtain $b'$
Else
    $s_0 \leftarrow \mathcal{U}_{n-k}$, and run $D((\mathbf{M}, w), s_0)$ to obtain $b'$
Endif
If $b = b'$ then output 1, and otherwise 0
```

Let Rand be the event that the matrix $\mathbf{M}$ was chosen randomly from uniform distribution $\mathcal{U}_{n,n-k}$, and let Real be the event that the matrix was generated by $\mathsf{Gen}_{\text{RNR}}$. Note that, from the assumption that $\mathbf{H}$ and $\mathbf{R}$ are indistinguishable, we know

$$\Pr\left[b = b' \mid \mathsf{Real}\right] - \Pr\left[b = b' \mid \mathsf{Rand}\right]$$

is upper-bounded by $\mathtt{Adv}_{\mathbf{H}}^{\mathrm{ind}}(k, O(t+n^2))$.

We are going to estimate the amount of $\Pr[b = b' \mid \mathsf{Real}]$ and $\Pr[b = b' \mid \mathsf{Rand}]$. However, by the same reason with Lemma 4, $\Pr[b = b' \mid \mathsf{Real}] = \Pr_D[\mathsf{Succ}]$ and

$$\sqrt[3]{64 n_1 \cdot \mathtt{Adv}_{\mathrm{SD}_{w_1}}^{\mathrm{oneway}}(n_1, t')} \geq \Pr[b = b' \mid \mathsf{Rand}] - \frac{1}{2},$$

where $t' = O(n^2(t + 2n^2)/\delta^2)$. Combining all these observations together, we have

$$\begin{aligned} \Pr_D[\mathsf{Succ}] - \frac{1}{2} &\leq \sqrt[3]{64 n_1 \cdot \mathtt{Adv}_{\mathrm{SD}_{w_1}}^{\mathrm{oneway}}(n_1, t')} + \mathtt{Adv}_{\mathbf{H}}^{\mathrm{ind}}(k, O(t+n^2)) \\ \delta &\leq 8 \cdot \sqrt[3]{n_1 \cdot \mathtt{Adv}_{\mathrm{SD}_{w_1}}^{\mathrm{oneway}}(n_1, t')} + 2 \cdot \mathtt{Adv}_{\mathbf{H}}^{\mathrm{ind}}(k, O(t+n^2)). \end{aligned}$$

□

The above lemma states that $\mathsf{Enc}_{\mathrm{RNR}}^{1}(pk, r') = r\mathbf{H}_1$, where r' is a random value for generating r, is pseudorandom and thus we can say that the randomized Niederreiter cryptosystem is the admissible cryptosystem. By combining Lemma 1 and Lemma 5 we can say

$$8 \cdot \sqrt[3]{n_1 \cdot \mathtt{Adv}_{\mathrm{SD}_{w_1}}^{\mathrm{oneway}}(n_1, t')} + 2 \cdot \mathtt{Adv}_{\mathbf{H}}^{\mathrm{ind}}(k, O(t+n^2)) \geq \mathtt{Adv}_{\mathrm{RNR}}^{\mathrm{sem}}(k, t),$$

where $t' = O(n^2(t+n^2)/\delta^2)$. Thus we can conclude that the randomized Niederreiter cryptosystem is IND-CPA cryptosystem if $\mathbf{H}$ is indistinguishable and syndrome decoding problem is hard.

5 Estimation of the Security Parameters

In all the cryptosystems, if the adversary has some partial information on the plaintext, the time complexity of recovering the entire plaintext is reduced. Particularly, let us consider the original McEliece cryptosystem. Let $m = [m_l | m_r]$ for $m_l \in \{0,1\}^{k_1}$ and $m_r \in \{0,1\}^{k_2}$ and let m_r be the partial information which the adversary knows in advance. Since

$$c = m\mathbf{G} \oplus e = m_l\mathbf{G}_1 \oplus m_r\mathbf{G}_2 \oplus e,$$

he can compute $m_r\mathbf{G}_2$ and

$$c' = m_l\mathbf{G}_1 \oplus m_r\mathbf{G}_2 \oplus e \oplus m_r\mathbf{G}_2 = m_l\mathbf{G}_1 \oplus e.$$

Thus, the time-complexity of recovering the entire m will be reduced to that of decrypting only c', hereby changing from 1 to the following:

$$\binom{n}{k_1+1} \cdot \binom{n-w}{k_1+1}^{-1}. \tag{5}$$

In this paper, we consider the semantically secure variant of the McEliece cryptosystem. In our scenario, the adversary knows that ciphertext is the encryption of either m_0 or m_1. Thus, we need to consider that the adversary knows the partial information of the given ciphertext and this situation is very similar to the above attack. That is, if the adversary can recover r, then he can distinguish the encryptions of m_0 and m_1. We present the estimated lower-bound of the size of the public-key in terms of this attack in Table 5. This time complexity is estimated according to (5).

Time complexity		
$(n, k, w) \Rightarrow$	(2048, 1289, 69)	(4096, 2560, 128)
$k_2 = 1$	$2^{101.7}$	$2^{186.1}$
$k_2 = 2$	$2^{101.6}$	$2^{186.0}$
$k_2 = 4$	$2^{101.3}$	$2^{185.7}$
$k_2 = 8$	$2^{101.7}$	$2^{185.2}$
$k_2 = 16$	$2^{99.7}$	$2^{184.2}$
$k_2 = 32$	$2^{97.6}$	$2^{182.2}$
$k_2 = 64$	$2^{93.4}$	$2^{178.4}$
$k_2 = 128$	$2^{85.7}$	$2^{170.8}$
$k_2 = 256$	$2^{71.72}$	$2^{156.6}$
$k_2 = 512$	$2^{48.6}$	$2^{131.05}$
$k_2 = 1024$	$2^{14.1}$	$2^{88.63}$

Table 1. Time Complexity for the “low weight codeword” Attack

6 Concluding Remarks

We formally show that random padding of the plaintext makes the McEliece and Niederreiter cryptosystems IND-CPA secure. It is worth noting that both these results do not allow tight reductions. Improving them, or, in other words, providing tightness for [15] and [10] is an open problem.

Another interesting open question, in the light of [16], is whether the security of the randomized versions of the McEliece and the Niederreiter cryptosystems is equivalent or not.

Finally, one might want to extend our result in order to achieve IND-CCA2 secure version of the McEliece as well as the Niederreiter cryptosystems without employing random oracles.

References

1. M. Bellare, P. Rogaway, “Random Oracles are Practical: a Paradigm for Designing Efficient Protocols,” Proc. CCS, pp.62–73, 1993.

2. M. Bellare, P. Rogaway, "Optimal Asymmetric Encryption - How to Encrypt with RSA," Proc. EUROCRYPT '94, LNCS 950, pp. 92–111, 1995.
3. E.R. Berlekamp, R.J. McEliece, H.C.A van Tilborg, "On the Inherent Intractability of Certain Coding Problems," IEEE Trans. Inf. Theory, vol. 24, pp.384–386, 1978.
4. A. Blum A. Kalai, H. Wasserman, "Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model, J. of ACM 50(4): pp. 506–519, 2003.
5. A. Canteaut, F. Chabaud "A new algorithm for finding minimum-weight words in a linear code: application to primitive narrow-sense BCH codes of length 511," IEEE Trans. Inf. Theory, vol. 44(1), pp.367–378, 1998.
6. P.-L. Cayrel, P. Gaborit and M. Girault, "Identity Based Identification and Signature Schemes Using Correcting Codes" Proc. WCC '07, pp. 69–78.
7. N. Courtois, M. Finiasz, N. Sendrier, "How to Achieve a McEliece-Based Digital Signature Scheme," Proc. Asiacrypt '01, LNCS 2248, pp.157–174, 2001.
8. R. Cramer, V. Shoup, "A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack," Proc. Crypto '98, LNCS 1462, pp.13–25, 1998.
9. T. El Gamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Trans. Inf. Theory, IT-31(4), pp.469–472, 1985.
10. J-B. Fischer, J. Stern, "An Efficient Pseudo-Random Generator Provably as Secure as Syndrome Decoding," Proc. Eurocrypt '96, LNCS 1070, pp.245–255, 1996.
11. E. Fujisaki and T. Okamoto, "Secure Integration of Asymmetric and Symmetric Encryption Schemes", Proc. Crypto '99, LNCS 1666, pp. 537–554, 1999.
12. O. Goldreich, "Foundation of Cryptography, Basic Tools," Cambridge University Press, 2001.
13. O. Goldreich, L.A. Levin, "A Hard-Core Predicate for all One-Way Functions," Proc. STOC '89, pp.25–32, 1989.
14. S. Goldwasser, S. Micali, "Probabilistic Encryption," J. Comp. Syst. Sci. 28:270–299, 1984.
15. J. Katz, J.S. Shin, "Parallel and Concurrent Security of the HB and HB$^+$ Protocols," Cryptology ePrint Archive: Rep.No. 461, 2005.
16. Y.X. Li, R.H. Deng, X.M. Wang, "The Equivalence of McEliece's and Niederreiter's Public-Key Cryptosystems," IEEE Trans. Inf. Theory, 40, pp.271–273, 1994.
17. G. Kabatiansky, E. Krouk and S. Semenov, "Error Correcting Codes and Security for Data Networks," Wiley, 2005.
18. K. Kobara, H. Imai, "Semantically Secure McEliece Public-Key Cryptosystems - Conversions for McEliece PKC -," Proc. PKC '01, LNCS 1992, pp.19–35, 2001.
19. J.S. Leon, "A probabilistic algorithm for computing minimum weights of large error-correcting codes" IEEE Trans. on Inf. Theory, volume 34(5), pp. 1354–1359, 2001.
20. P. Loidreau and N. Sendrier, "Weak keys in the McEliece public-key cryptosystem", IEEE Trans. on Inf. Theory, vol. 47 (3), pp. 1207 – 1211, 2001.
21. R.J. McEliece, "The Theory of Information and Coding (Vol. 3 of The Encyclopedia of Mathematics and Its Applications.), Reading, Mass., Addison-Wesley, 1977.
22. R.J. McEliece, "A Public-Key Cryptosystem Based on Algebraic Coding Theory," Deep Space Network Progress Rep., 1978.
23. H. Niederreiter, "Knapsack-type Cryptosystems and Algebraic Coding Theory," Prob. of Control and Inf. Theory, vol. 15(2), pp.159–166, 1986.
24. P. Paillier, "Public-Key Cryptosystem Based on Discrete Logarithm Residues," Proc. Eurocrypt '99, LNCS 1592, pp.223–238, 1999
25. E. Petrank, R.M. Roth, "Is Code Equivalence Easy to Decide?," IEEE Trans. Inf. Theory, Vol.43, pp.1602–1604, 1997.

26. D. Pointcheval, “Chosen-Ciphertext Security for any One-Way Cryptosystem”, Proc. PKC ’00, LNCS 1751, pp. 129–146, 2000.
27. N. Sendrier, “Finding the Permutation Between Equivalent Linear Codes: The Support Splitting Algorithm,” IEEE Trans. Inf. Theory, 46(4), pp.1193–1203, 2000.
28. V. Shoup, “OAEP reconsidered,” CRYPTO ’01, LNCS 2139, pp. 239–259, 2001.
29. J. Stern, “A Method for Finding Codewords of Small Weight”, Proc. Coding Theory and Applications, LNCS 388, pp. 106–113, Springer, 1989.
30. M. Suzuki, K. Kobara, H. Imai, “Privacy Enhanced and Light Weight RFID System without Tag Synchronization and Exhaustive Search,” IEEE SMC, Taipei, 2006.