Des. Codes Cryptogr. (2008) 49:289–305
DOI 10.1007/s10623-008-9175-9

Semantic security for the McEliece cryptosystem without random oracles

Ryo Nojima · Hideki Imai · Kazukuni Kobara · Kirill Morozov

Received: 1 June 2007 / Revised: 1 December 2007 / Accepted: 21 January 2008 /
Published online: 6 March 2008

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, Kobara and Imai in the context of RFID security. Our proof relies on the technical result by Katz and Shin from Eurocrypt '05 showing "pseudorandomness" implied by the learning parity with noise (LPN) problem. We do not need the random oracles as opposed to the known generic constructions which, on the other hand, provide a 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 some estimates for suitable key sizes together with corresponding workload required for successful attack.

Keywords Semantic security · Cryptographic standard model · McEliece cryptosystem · Niederreiter cryptosystem

AMS Classification 11T71

Ryo Nojima's work was done when he was at the University of Tokyo, Japan.

R. Nojima
Information Security Research Center, National Institute of Information and Communications Technology (NICT), Tokyo, Japan
e-mail: ryo-no@nict.go.jp

H. Imai
Department of Electrical, Electronic and Communication Engineering, Chuo University, Tokyo, Japan
e-mail: h-imai@aist.go.jp

H. Imai · K. Kobara · K. Morozov (✉)
Research Center for Information Security (RCIS), National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
e-mail: kirill.morozov@aist.go.jp

K. Kobara
e-mail: k-kobara@aist.go.jp

1 Introduction

The *semantic security* (also called *indistinguishability*) defined by Goldwasser and Micali [15] is the security notion for a public-key cryptosystem (PKC). Its 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 [1,8,9,25].

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* [23] and the *Niederreiter* [24] cryptosystems whose security is ensured under the following two assumptions: (a) hardness of the bounded distance decoding of random binary linear codes[1] 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.[2] From the security point of view, these cryptosystems has 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 [23], 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 public-key encryption but also 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. Additionally, we estimate the time-complexity of breaking this version of the McEliece PKC (which we call the *randomized McEliece cryptosystem*) and show that it remains practical.

A bit more formally, let $E_{pk}(\cdot)$ be an encryption algorithm of the McEliece (or the Niederreiter) cryptosystem with message space $\{0,1\}^k$, where $m \in \{0,1\}^{k_2}$ an actual 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.

We show that this padding yields the encryption secure against 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. Our paper was inspired by the work of Suzuki, Kobara and Imai [31] where such padding was suggested (without a formal proof) for increasing the security of encryption.

[1] 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].

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

The technical tool which we use to prove the security of our scheme is the lemma by Katz and Shin [16] 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 in the LPN problem, substituting the random matrix by a pseudorandom one preserves 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 the Goldreich-Levin (hardcore) predicate theorem [14] 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) [2,12,18,27]. 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 [29].

Fujisaki and Okamoto [12] and Pointcheval [27] independently suggested a conversion from any one-way PKC to a PKC semantically secure against chosen ciphertext attack (IND-CCA). 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 semantic security against adaptive 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 Sect. 2, we provide some basic notation and definitions, and describe the original versions of the PKC's in question. In Sect. 3, their randomized versions are introduced along with related security definitions and the main result is stated, while its proof is presented in Sect. 4. In Sect. 5, the security parameters for the randomized McEliece cryptosystem are estimated. In Sect. 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. Specifically, the code we concentrate on is the irreducible binary Goppa code and the relationships between these parameters are $n = 2^{m'}$ and $k \geq n - m'w$ 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. Let $\mathcal{D}$ be a probability distribution over S, and let O be an algorithm which, on input an empty string, outputs an element $s \in S$ according to the distribution $\mathcal{D}$. Then, A^O is the oracle O embedded probabilistic polynomial-time algorithm which can obtain the element of S according to the distribution $\mathcal{D}$ through O. That is, when A queries to O with an empty string, O chooses an element according to $\mathcal{D}$ and returns it back. We usually denote this $A^{\mathcal{D}}$ for short.

A public-key encryption scheme is composed of a triplet of algorithms Π=(Gen_Π, Enc_Π, Dec_Π). 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 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

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

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

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

where the maximum is over all D 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 [23] consists of a triplet of PPT algorithms ME=($\mathsf{Gen}_{\mathrm{ME}}$, $\mathsf{Enc}_{\mathrm{ME}}$, $\mathsf{Dec}_{\mathrm{ME}}$) and $M = \{0,1\}^k$.

- Key generation algorithm $\mathsf{Gen}_{\mathrm{ME}}$ works as follows:
  1. Generate a $k \times n$ generator matrix $\mathbf{G}'$ of an irreducible 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 $sk = (\mathbf{S}, \mathbf{G}', \mathbf{P})$.
- 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 $e \leftarrow \mathcal{E}_{n,w}$.
- Decryption algorithm $\mathsf{Dec}_{\mathrm{ME}}$, given ciphertext c and secret key sk as input, 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 [24] 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 a triplet of PPT algorithms $\mathrm{NR} = (\mathsf{Gen}_{\mathrm{NR}}, \mathsf{Enc}_{\mathrm{NR}}, \mathsf{Dec}_{\mathrm{NR}})$ and $M \subset \{0,1\}^n$ is a set of all strings of weight w.

- Key generation algorithm $\mathsf{Gen}_{\mathrm{NR}}$ works as follows:
  1. Generate a $(n-k) \times n$ parity check matrix $\mathbf{H}''$ of an irreducible 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 $sk = (\mathbf{S}, \mathbf{H}'', \mathbf{P})$.
- Encryption algorithm $\mathsf{Enc}_{\mathrm{NR}}$ takes a plaintext $m \in \{0,1\}^n$ of weight w and pk as input and outputs ciphertext $c = m\mathbf{H}$.
- Decryption algorithm $\mathsf{Dec}_{\mathrm{NR}}$, given ciphertext c and secret key sk, works as follows:
  1. Compute $\mathbf{S}^{-1}c^T = \mathbf{H}''(\mathbf{P}m^T)$, where $\mathbf{S}^{-1}$ 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 [23] 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 [31], 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 string 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 the irreducible binary Goppa codes 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 as the weak keys [21] 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 sub-exponential. 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* [28], 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 [26] which is conjectured to be in $\mathcal{NP} \setminus \mathcal{P}$. 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 Subsects. 3.4 and 3.5.

For the excellent surveys on security of both PKC's, we refer the reader to [10,17].

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

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

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

$$\mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t) = \max_{D}\left\{\mathtt{Adv}^{\mathrm{ind}}_{D,\mathbf{G}}(k)\right\}, \tag{1}$$

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 [6,7].

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

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

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

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

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

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

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

We assume that the LPN_θ problem is hard for some θ satisfying $w = \lfloor \theta(n+1) \rfloor$. In fact, all known algorithms for solving this problem are still sub-exponential 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 [30], Canteaut and Chabaud [5], and its time complexity is roughly

$$\binom{n}{k} \cdot \binom{n-w}{k}^{-1}, \tag{2}$$

where w is the weight of the noise.

With the above two assumptions, we can prove the first part of our main result:

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

The proof is given in Sect. 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

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

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

$$\mathrm{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k,t) = \max_D \left\{ \mathrm{Adv}^{\mathrm{ind}}_{D,\mathbf{H}}(k) \right\}, \tag{3}$$

where the maximum is over all D with time-complexity t. We say $\mathbf{H}$ is *indistinguishable* if $\mathrm{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k,t)$ 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.

Assume that the SD_w problem is hard. Together with the previous assumption, it allows us to prove the second part of our main result:

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 Sect. 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 property:

- Partible: 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_Π, $r \in R$, 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 and let

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

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

$$\mathtt{Adv}^{\mathrm{ind}}_{\mathsf{Enc}^1_\Pi}(k, t) = \max_D \left\{\mathtt{Adv}^{\mathrm{ind}}_{D,\mathsf{Enc}^1_\Pi}(k)\right\}, \tag{4}$$

where the maximum is over all D with time-complexity t. Then, the function $\mathtt{Adv}^{\mathrm{ind}}_{\mathsf{Enc}^1_\Pi}(k, t)$ 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*

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

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^2_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:

$D'(pk, \tilde{s})$
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 uniform 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 by $\Pr_D\left[\mathsf{Succ}\right]$.

Note that, from (4), we know

$$\Pr\left[D' = 1 \mid \mathsf{Real}\right] - \Pr\left[D' = 1 \mid \mathsf{Rand}\right] \leq \mathtt{Adv}^{\mathrm{ind}}_{\mathsf{Enc}^1_\Pi}(k, t + t'), \tag{5}$$

where t' is the worst-case time-complexity of computing Enc^2_Π.

We claim that $\Pr\left[D' = 1 \mid \mathsf{Real}\right] = \Pr_D[\mathsf{Succ}]$. 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 $\Pr\left[D' = 1 \mid \mathsf{Rand}\right] = \frac{1}{2}$. Here we know that $\tilde{s}$ is distributed according to the uniform distribution $\mathcal{U}_{p(k)}$. Therefore, $\tilde{s} \oplus \mathsf{Enc}^2_\Pi(pk, m_b)$ given to D is uniformly distributed as well. This means that D obtains no information related to b. Since D' outputs 1 iff D succeeds, we can conclude that $\Pr\left[D' = 1 \mid \mathsf{Rand}\right] = \frac{1}{2}$.

By combining these results, now we can estimate (5) as follows:

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

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

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

This concludes the proof. □

Above lemma implies that, 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 re-write the above equation as follows:

$$c = [r|m]\mathbf{G} \oplus e = \{r\mathbf{G}_1 \oplus e\} \oplus m\mathbf{G}_2. \tag{6}$$

Let us define the algorithm $\mathsf{Enc}^1_{\mathrm{RME}}(pk, [r|r'])$ as $r\mathbf{G}_1 \oplus e$, where r' is the random number utilized for generating the weight w random vector $e \in \{0,1\}^n$, and define the algorithm $\mathsf{Enc}^2_{\mathrm{RME}}(pk, m)$ as $m\mathbf{G}_2$. Then, clearly, the randomized McEliece cryptosystem is partible. Hence, in order 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 $\mathsf{Enc}^1_{\mathrm{RME}}(pk, r)$. Note that we set the length of a and r as k_1. So each response from the oracle $\mathcal{Q}_{r,\theta}$ becomes $(a, \langle r, a\rangle \oplus \nu)$ of length $k_1 + 1$.

Lemma 2 (Lemma 1 in [16]) *If there exists an algorithm D which runs in time t, makes queries q times and such that*

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

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

This is the key technical lemma which was rigorously proved in [16]. We re-write the above lemma as follows:

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

$$\Pr\left[r \leftarrow \mathcal{U}_{k_1}, b \leftarrow \mathcal{U}_1 \mid D^{\mathcal{O}_b} = b\right] - \frac{1}{2} \geq \delta$$

then

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

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

To prove the semantic security of the randomized McEliece cryptosystem, we slightly change the statement of the LPN problem. More precisely, we represent the sequence of responses from the oracle by the linear algebraic notation and re-define the LPN problem with this. Let $(a_i, \langle a_i, r\rangle \oplus \nu_i)$ be a response from the oracle at time i for $1 \leq i \leq n$, let us regard each a_i as a column vector and set $\mathbf{R}_1 = [a_1|a_2|\ldots|a_n]$. Then, once fixing the number of the queries to n which is polynomially bounded, the hardness of the LPN problem is equivalent to saying that, given $r\mathbf{R}_1 \oplus \nu$, and $\mathbf{R}_1$, it is hard for any PPT algorithm to output r, where $\nu^T = [\nu_1|\nu_2|\ldots|\nu_n]$. With this notation, the previous lemma is equivalent to saying that

$$r\mathbf{R}_1 \oplus \nu$$

is pseudorandom. Remind that our target here was to prove

$$r\mathbf{G}_1 \oplus e$$

being pseudorandom, where e is an error vector of weight w. So what we want to show here is that replacing $\mathbf{R}_1$ and ν with $\mathbf{G}_1$ and e, respectively, preserves the pseudorandomness.

To do so, we first replace ν with e and prove that

$$r\mathbf{R}_1 \oplus e$$

is pseudorandom, where $e \leftarrow \mathcal{E}_{n,w}$ with $w = \lfloor \theta(n+1) \rfloor$.

Lemma 3 *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 $q = n$.*

Then if there exists an algorithm D which runs in time t and such that

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

then

$$2(n+1) \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k_1, t', q') \geq \delta, \tag{7}$$

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

Proof Sketch Let us denote the distinguisher described 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 $b_i = \langle r, a_i \rangle \oplus \nu_i$.
- D' sets $\mathbf{R}_1 = [a_1 | a_2 | \ldots | a_n]$, where we regard each a_i as a column vector and thus $\mathbf{R}_1$ is a $k_1 \times n$ 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 $[b_1 | b_2 | \ldots | b_n]$.
- 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 $e = [\nu_1 | \nu_2 | \ldots | \nu_n]$ 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 cases of weights. This introduces $(n+1)$ in the left part of (7), still leaving the advantage negligible. □

We showed that $r\mathbf{R}_1 \oplus e$ is pseudorandom. Therefore, the rest of the proof is to replace the random matrix $\mathbf{R}$ with the (pseudorandom) public key matrix $\mathbf{G}$ and to show that $r\mathbf{G}_1 \oplus e$ is pseudorandom. Here note that this means that the randomized McEliece cryptosystem is in fact an admissible cryptosystem.

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

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

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 $\mathsf{Gen}_{\mathrm{ME}}$. Then from (1) we can describe the inequality

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

where $t_2 = O(t + n^2)$. 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. Since D' outputs 1 iff D succeeds, $\Pr\left[b = b' \mid \mathsf{Real}\right] = \Pr_D[\mathsf{Succ}]$.

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

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

But we know from Lemma 3 that

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

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, we obtain

$$\begin{aligned} \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t_2) &\geq \Pr\left[b = b' \mid \mathsf{Real}\right] - \Pr\left[b = b' \mid \mathsf{Rand}\right] \\ &\geq \Pr_D[\mathsf{Succ}] - \frac{1}{2} - 2(n+1) \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{LPN}_\theta}(k_1, t_1, q_1). \end{aligned}$$

A simple modification yields

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

and therefore

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

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

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

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 [11] which is useful for proving the semantic security of the randomized Niederreiter cryptosystem. The result stated in [11] 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 $\mathbf{R}_1$, $r\mathbf{R}_1$ 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 show that the randomized Niederreiter cryptosystem is partible by re-writing the above equation as follows:

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

Thus it only remains to prove 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. We utilize the result of [11][3].

Theorem 1 **([11,13])** *If there exists an algorithm D which runs in time t, and such that*

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

then $4\sqrt[3]{n_1 \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_{w_1}}(n_1, t')} \geq \delta$, *where* $t' = O(n^2(t+n^2)/\delta^2)$.

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 D which runs in time t, and such that*

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

then $4\sqrt[3]{n_1 \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_{w_1}}(n_1, t')} \geq \delta$, *where* $t' = O(n^2(t+n^2)/\delta^2)$.

[3] It uses the Goldreich-Levin (hardcore) predicate theorem [13,14] 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 [13] and Theorem 1 in [11].

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 $\mathsf{Enc}^1_{\mathrm{RNR}}(pk, r') = r\mathbf{H}_1$ 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*

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

then

$$4 \cdot \sqrt[3]{n_1 \cdot \mathrm{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_{w_1}}(n_1, t')} + 2 \cdot \mathrm{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k, O(t + n^2)) \geq \delta,$$

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

Proof Sketch 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.

$\underline{D'(\mathbf{M}, w)}$
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 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 $\mathsf{Gen}_{\mathrm{RNR}}$. Then from (3), we can deduce the inequality

$$\Pr\left[b = b' \mid \mathsf{Real}\right] - \Pr\left[b = b' \mid \mathsf{Rand}\right] \leq \mathrm{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k, O(t + n^2)).$$

Now, we are going to estimate $\Pr\left[b = b' \mid \mathsf{Real}\right]$ and $\Pr\left[b = b' \mid \mathsf{Rand}\right]$. First, we note that, similarly to the proof of Lemma 4, we have $\Pr\left[b = b' \mid \mathsf{Real}\right] = \Pr_D[\mathsf{Succ}]$. And we can bound $\Pr\left[b = b' \mid \mathsf{Rand}\right] - 1/2$ by Corollary 2:

$$\Pr\left[b = b' \mid \mathsf{Rand}\right] - 1/2 \leq 2 \cdot \sqrt[3]{n_1 \cdot \mathrm{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_{w_1}}(n_1, t')},$$

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

Combining all these together, we have

$$\begin{aligned}\Pr_D[\mathsf{Succ}] - \frac{1}{2} &= \Pr\left[b = b' \mid \mathsf{Real}\right] - \frac{1}{2} \\ &\leq \Pr\left[b = b' \mid \mathsf{Rand}\right] + \mathrm{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k, O(t + n^2)) - \frac{1}{2} \\ &\leq 2 \cdot \sqrt[3]{n_1 \cdot \mathrm{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_{w_1}}(n_1, t')} + \mathrm{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k, O(t + n^2)).\end{aligned}$$

Therefore, $\delta \leq 4 \cdot \sqrt[3]{n_1 \cdot \mathrm{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_{w_1}}(n_1, t')} + 2 \cdot \mathrm{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k, O(t + n^2))$. □

The above lemma states that $\mathsf{Enc}^1_{\mathrm{RNR}}(pk, r') = r\mathbf{H}_1$, 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

$$4 \cdot \sqrt[3]{n_1 \cdot \mathtt{Adv}^{\mathrm{oneway}}_{\mathrm{SD}_{w_1}}(n_1, t')} + 2 \cdot \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{H}}(k, O(t + n^2)) \geq \mathtt{Adv}^{\mathrm{sem}}_{\mathrm{RNR}}(k, t),$$

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 (2) to the following:

$$\binom{n}{k_1 + 1} \cdot \binom{n - w}{k_1 + 1}^{-1}. \tag{8}$$

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

Table 1 Time complexity for the "low weight codeword" attack

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

on the size of the public key in terms of time complexity of this attack in Table 1. This time complexity is estimated according to (8).

6 Concluding remarks

We formally show that random padding of the plaintext makes the McEliece and the 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 [16] and [11] is an open problem.

Another interesting open question, in the light of [20], 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. Bellare M., Rogaway P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of CCS, pp. 62–73 (1993).
2. Bellare M., Rogaway P.: Optimal asymmetric encryption – how to encrypt with RSA. In: EUROCRYPT '94, LNCS vol. 950, pp. 92–111 (1995).
3. Berlekamp E., McEliece R.J., van Tilborg H.C.A.: On the inherent intractability of certain coding problems. IEEE Trans. Inform. Theory **24**, 384–386 (1978).
4. Blum A., Kalai A., Wasserman H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM **50**(4), 506–519 (2003).
5. Canteaut A., Chabaud F.: A new algorithm for finding minimum-weight words in a linear code: application to primitive narrow-sense BCH codes of length 511. IEEE Trans. Inform. Theory **44**(1), 367–378 (1998).
6. Cayrel P.-L., Gaborit P., Girault M.: Identity based identification and signature schemes using correcting codes. In: WCC '07, pp. 69–78 (2007).
7. Courtois N., Finiasz M., Sendrier N.: How to achieve a McEliece-based digital signature scheme. In: Asiacrypt '01, LNCS vol. 2248, pp. 157–174 (2001).
8. Cramer R., Shoup V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Crypto '98, LNCS vol. 1462, pp. 13–25 (1998).
9. El Gamal T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inform. Theory **31**(4), 469–472 (1985).
10. Engelbert D., Overbeck R., Schmidt A.: A summary of McEliece-type cryptosystems and their security. J. Math. Cryptol. **1**(2), 151–199 (2007).
11. Fischer J.-B., Stern J.: An efficient pseudo-random generator provably as secure as syndrome decoding. In: Eurocrypt '96, LNCS vol. 1070, pp. 245–255 (1996).
12. Fujisaki E., Okamoto T.: Secure integration of asymmetric and symmetric encryption schemes. In: Crypto '99, LNCS vol. 1666, pp. 537–554 (1999).
13. Goldreich O.: Foundation of Cryptography, Basic Tools. Cambridge University Press (2001).
14. Goldreich O., Levin L.A.: A hard-core predicate for all one-way functions. In: STOC '89, pp. 25–32 (1989).
15. Goldwasser S., Micali S.: Probabilistic encryption. J. Comp. Syst. Sci. **28**, 270–299 (1984).
16. Katz J., Shin J.S.: Parallel and concurrent security of the HB and HB+ protocols. In: Eurocrypt '06, LNCS vol. 4004, pp. 73–87 (2006).
17. Kabatiansky G., Krouk E., Semenov S.: Error Correcting Codes and Security for Data Networks. Wiley (2005).
18. Kobara K., Imai H.: Semantically secure McEliece public-key cryptosystems – conversions for McEliece PKC. In: PKC '01, LNCS vol. 1992, pp. 19–35 (2001).
19. Leon J.S.: A probabilistic algorithm for computing minimum weights of large error-correcting codes. IEEE Trans. Inform. Theory **34**(5), 1354–1359 (2001).

20. Li Y.X., Deng R.H., Wang X.M.: The equivalence of McEliece's and Niederreiter's public-key cryptosystems. IEEE Trans. Inform. Theory **40**, 271–273 (1994).
21. Loidreau P., Sendrier N.: Weak keys in the McEliece public-key cryptosystem. IEEE Trans. Inform. Theory **47**(3), 1207–1211 (2001).
22. McEliece R.J.: The theory of information and coding. In: The Encyclopedia of Mathematics and Its Applications, vol. 3. Addison-Wesley (1977).
23. McEliece R.J.: A public-key cryptosystem based on algebraic coding theory. Deep Space Network Prog. Rep. (1978).
24. Niederreiter H.: Knapsack-type cryptosystems and algebraic coding theory. Prob. Control Inform. Theory **15**(2), 159–166 (1986).
25. Paillier P.: Public-key cryptosystem based on discrete logarithm residues. In: Eurocrypt '99, LNCS vol. 1592, pp. 223–238 (1999).
26. Petrank E., Roth R.M.: Is code equivalence easy to decide? IEEE Trans. Inform. Theory **43**, 1602–1604 (1997).
27. Pointcheval D.: Chosen-ciphertext security for any one-way cryptosystem. In: PKC '00, LNCS vol. 1751, pp. 129–146 (2000).
28. Sendrier N.: Finding the permutation between equivalent linear codes: the support splitting algorithm. IEEE Trans. Inform. Theory **46**(4), 1193–1203 (2000).
29. Shoup V.: OAEP reconsidered. In: Crypto '01, LNCS vol. 2139, pp. 239–259 (2001).
30. Stern J.: A method for finding codewords of small weight. In: Coding Theory and Applications, LNCS vol. 388, pp. 106–113 (1989).
31. Suzuki M., Kobara K., Imai H.: Privacy enhanced and light weight RFID system without tag synchronization and exhaustive search. In: IEEE SMC (2006).