Semantic Security for the McEliece Cryptosystem without Random Oracles

Ryo Nojima[1], Hideki Imai[2][3], 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's) 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 conversions, while the recent ones provide stronger protection (as compared to our scheme) – against adaptive chosen ciphertext attack (IND-CCA2).

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 [12] 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, 3, 4, 8, 18].

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* [16] and the *Niederreiter* [17] 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.

syndrome decoding problem or, equivalently [20], the *learning parity with noise (LPN)*[†] and b) indistinguishability of the scrambled generating matrix of the Goppa code from random.[‡] In these cryptosystems, efficiently decodable codes such as Goppa codes [15] must be used in order to make them efficient and secure. 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 cryptosystem is the following: a) As it was pointed out in the original paper [16], the hardware implementation of this 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., digital signatures [7]) 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 yet been 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 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 $E_{pk} : \{0,1\}^k \rightarrow \{0,1\}^n$ be an encryption algorithm of the McEliece cryptosystem, $m \in \{0,1\}^{k_2}$ 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 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 [22] where it was suggested (without a formal proof) for increasing the security of encryption.[¶]

[†] This problem is known to be NP-complete [1].

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

[§] The same result follows for the Niederreiter cryptosystem due to their equivalence [23], its formal proof appears in the full version of this paper.

[¶] In fact, the Niederreiter cryptosystem was employed in this work.

The technical tool which we use to prove the security of our scheme is the technical lemma by Katz and Shin [13] which established a pseudorandomness of the queries to the oracle for LPN problem. The key difference from their setting is that we have a generating 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 2, states that substituting a random matrix by a pseudorandom one preserves the pseudorandomness of the output. Then, under the above assumptions the proof of Theorem 1 stating semantic security of the McEliece cryptosystem with randomized plaintext follows as well as the similar result for the Niederreiter cryptosystem (the latter will appear in the full version of this work).

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 [11] 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 employing random oracles include (but is not limited to) [4, 24, 25, 14]. The optimal asymmetric encryption padding (OAEP) by Bellare and Rogaway [4] is the first result of such kind but it dealt with one-way trapdoor permutations (while the cryptosystem we consider is only a trapdoor function) and needed some fixing in the general case [26].

Fujisaki and Okamoto [24] and Pointcheval [25] independently suggested a conversion from any one-way PKC to a PKC semantically secure against chosen ciphertext attack (IND-CCA). Finally, Kobara and Imai [14] 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 note 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. In Section 3, the randomized McEliece cryptosystem is formally described along with its underlying assumptions. The proof of IND-CPA security of the randomized version is presented in Section 4. In Section 5, the security parameters for the presented scheme are estimated. Section 6 concludes our work and discusses the open questions.

2 Preliminaries

Throughout this paper, we consider k and n as security parameters. We denote the probabilistic polynomial-time as PPT and we call the algorithm *efficient* if its running time is polynomial. Let $s \overset{\$}{\leftarrow} S$ 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 $\mathit{\Pi} = (\mathsf{Gen}_{\mathit{\Pi}}, \mathsf{Enc}_{\mathit{\Pi}}, \mathsf{Dec}_{\mathit{\Pi}})$. The key generation algorithm $\mathsf{Gen}_{\mathit{\Pi}}$ 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 $\mathsf{Enc}_{\mathit{\Pi}}$ is a PPT algorithm which, on input pk and a plaintext $m \in M$, outputs a ciphertext $c \in \{0,1\}^*$. The decryption algorithm $\mathsf{Dec}_{\mathit{\Pi}}$ 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 $\mathsf{Gen}_{\mathit{\Pi}}$, and any plaintext $m \in M$, $\mathsf{Dec}_{\mathit{\Pi}}(sk, \mathsf{Enc}_{\mathit{\Pi}}(pk, m)) = m$.

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 $\mathit{\Pi} = (\mathsf{Gen}_{\mathit{\Pi}}, \mathsf{Enc}_{\mathit{\Pi}}, \mathsf{Dec}_{\mathit{\Pi}})$ be a public-key encryption scheme and let $A = (A_1, A_2)$ be a PPT algorithm. For every $k \in \mathbb{N}$, we define

$$\mathtt{Adv}^{\mathrm{sem}}_{A,\mathit{\Pi}}(k) = \Pr\left[\begin{array}{c|c} (pk, sk) \leftarrow \mathsf{Gen}_{\mathit{\Pi}}(1^k), & \\ (m_0, m_1) \leftarrow A_1(pk), & A_2(y) = b \\ b \xleftarrow{\$} \{0,1\}, & \\ y \leftarrow \mathsf{Enc}_{\mathit{\Pi}}(pk, m_b) & \end{array}\right] - \frac{1}{2}.$$

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

$$\mathtt{Adv}^{\mathrm{sem}}_{\mathit{\Pi}}(k, t) = \max\nolimits_A \left\{\mathtt{Adv}^{\mathrm{sem}}_{A,\mathit{\Pi}}(k)\right\},$$

where the maximum is over all A with time-complexity t. We say that $\mathit{\Pi}$ is semantically secure if the function $\mathrm{Adv}^{\mathrm{sem}}_{\mathit{\Pi}}(\cdot)$ is negligible for every algorithm A which time-complexity is polynomial in k.

3 Randomized McEliece Cryptosystem

In the McEliece cryptosystem [16] (see Appendix A.1 for its description), if the adversary obtains a ciphertext, say 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 weight being w or not. To avoid such the situation, concatenating a random sequence r to a message m and encrypting $[r|m]$ has been often employed. But there has been no formal proof of this padding appeared in the literatures. In this paper, we prove the security formally.

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 just encrypts $[r|m]$ instead of m itself. The decryption algorithm 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.

3.1 Security Assumptions

In order to prove the security of this scheme, we use the same assumptions as for the original PKC. The short discussion to background their actuality is presented in Appendix A.2. Let us establish here some relevant notation and facts.

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 conjecture that for any polynomially bounded t, $\mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t)$ is negligible.

This conjecture was also utilized in [7] to construct a digital signature scheme. But we also need the following assumption known as the "learning parity with the noise" problem. We employ the definition given in [13].

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

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

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

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

the maximum is over all A with time-complexity t and query-complexity q.

All known algorithms for solving this problem are still super-polynomial time [2]. Especially, for fixed q and small amount of noise, the best one is due to Canteaut and Chabaud [5], its time complexity is roughly $n^{w(1/2+o(1))}$.

Also we can denote this problem in another way. Let $q = n$. Then we can rewrite this as follows: for any A,

$$\begin{aligned}\Pr\left[\mathbf{R} \leftarrow \mathcal{U}_{k,n}, s \leftarrow \{0,1\}^k, e \leftarrow \mathcal{E}_{n,w} \mid A(\mathbf{R}, w, s\mathbf{R} \oplus e) = s\right]& \\ \leq (n+1) \cdot \mathtt{Adv}^{\mathrm{one-way}}_{A,\mathrm{LPN}_\theta}(k, t, n),&\end{aligned}$$

where we set $w = \lfloor \theta(n+1) \rfloor$. Note that, in the above inequation, a random noise e of weight w is added instead of the noise generated by the Bernoulli distribution and this results in multiplication of $(n+1)$ in the right-hand side.

This upper bound easily follows from the fact that, in the Bernoulli distribution, the probability of the weight of e being $\lfloor \theta(n+1) \rfloor$ is at least $\frac{1}{n+1}$.

It is easy to see that if $\mathbf{G}$ looks like $\mathbf{R}$, i.e. pseudorandom, and the LPN problem is hard, then the McEliece cryptosystem has one-wayness.[‖] But, with these two – conjectured to be hard – problems, we can prove more useful result, that is the semantic security of the randomized McEliece cryptosystem. The proof is presented in the next section.

4 Semantic Security of the Randomized Version

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:

$$c = [r|m]\mathbf{G} \oplus e = \{r\mathbf{G}_1 \oplus e\} \oplus m\mathbf{G}_2. \tag{1}$$

Intuitively, the semantic security of the randomized McEliece cryptosystem is ensured by the *pseudorandomness* of $r\mathbf{G}_1 \oplus e$. In the formal proof of this fact, the following lemma, which states that the hardness of the LPN problem implies pseudorandomness of the output, plays an important role.

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

Lemma 1 (Lemma 1 in [13]). *If there exists an algorithm which runs in time t, makes queries q times and such that*

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

$\mathtt{Adv}_{\mathrm{LPN}_\theta}^{\mathrm{one-way}}(k_1, t_L, q_L) \geq \delta/4$, *where* $t_L = O(tk_1\delta^{-2}\log k_1)$, $q_L = O(q\delta^{-2}\log k_1)$

This is the key technical lemma which was rigorously proved in [13]. 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*

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

then

$$\mathtt{Adv}_{\mathrm{LPN}_\theta}^{\mathrm{one-way}}(k_1, t_L, q_L) \geq \delta/2. \tag{3}$$

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, for the same reason as noted in the previous section, we can rewrite the above inequations (2) and (3) as

$$\Pr\left[\begin{matrix} r \leftarrow \mathcal{U}_{k_1}, \mathbf{R}_1 \leftarrow \mathcal{U}_{k_1,n}, e \leftarrow \mathcal{E}_{n,w}, \\ b \leftarrow \mathcal{U}_1, s_0 \leftarrow \mathcal{U}_n, s_1 \leftarrow r\mathbf{R}_1 \oplus e \end{matrix}\middle| D(\mathbf{R}_1, w, s_b) = b\right] - \frac{1}{2} \geq \delta \tag{4}$$

[‖] Actually, there is a possibility that $\mathbf{G}$ is not pseudorandom but the McEliece PKC has still has one-wayness.

and

$$2(n+1)\cdot \mathtt{Adv}^{\text{one-way}}_{\mathrm{LPN}_\theta}(k_1, t'_L, q'_L) \geq \delta, \tag{5}$$

respectively, where $q'_L = O(n\delta^{-2}\log k_1)$ and t'_L is essentially the same as t_L. Also note that this holds even if we modify the above inequation as follows: replace $D(\mathbf{R}_1, w, s_b) = b$ with $D(\mathbf{R}, w, s_b) = b$, and $\mathbf{R}_1 \leftarrow \mathcal{U}_{k_1,n}$ with $\mathbf{R} \leftarrow \mathcal{U}_{k,n}$.

To prove the theorem, 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.

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

$$\begin{aligned}\Pr\left[r \leftarrow \mathcal{U}_{k_1}, (\mathbf{G}, w) \leftarrow \mathsf{Gen}_{\mathrm{ME}}(1^k), e \leftarrow \mathcal{E}_{n,w} \mid D((\mathbf{G}, w), r\mathbf{G}_1 \oplus e) = 1\right]\\ - \Pr\left[s_0 \leftarrow \mathcal{U}_n, (\mathbf{G}, w) \leftarrow \mathsf{Gen}_{\mathrm{ME}}(1^k) \mid D((\mathbf{G}, w), s_0) = 1\right] \geq \delta,\end{aligned} \tag{6}$$

then

$$4(n+1)\cdot \mathtt{Adv}^{\text{one-way}}_{\mathrm{LPN}_\theta}(k_1, t''_L, q''_L) + 2\cdot \mathtt{Adv}^{\text{ind}}_{\mathbf{G}}(k, t_G) \geq \delta.$$

Here t''_L, q''_L and t_G are essentially the same as t'_L, q'_L and t, respectively.

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.

```
$D'(\mathbf{M}, w)$
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 from the assumption that $\mathbf{G}$ and $\mathbf{R}$ are indistinguishable, $|\Pr[b = b' \mid \mathsf{Real}] - \Pr[b = b' \mid \mathsf{Rand}]|$ is negligible.

We first claim that $\Pr[b = b' \mid \mathsf{Real}] = \Pr_D[\mathsf{Succ}]$. 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 claim that $\left|\Pr[b = b' \mid \mathsf{Rand}] - \frac{1}{2}\right|$ is negligible. From the construction of D',

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

Thus, from the inequations (4) and (5), and their following modification, we can obtain that

$$\left| \Pr\left[b = b' \mid \mathsf{Rand}\right] - \frac{1}{2} \right|$$

is negligible. Combining all these observations together, we have

$$\Pr_D[\mathsf{Succ}] - \frac{1}{2} \leq \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t_G) + 2(n+1) \cdot \mathtt{Adv}^{\mathrm{one-way}}_{\mathrm{LPN}_\theta}(k_1, t''_L, q''_L)$$
$$\delta \leq 2 \cdot \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t_G) + 4(n+1) \cdot \mathtt{Adv}^{\mathrm{one-way}}_{\mathrm{LPN}_\theta}(k_1, t''_L, q''_L).$$

Also it is easy to see that t_G, t''_L, and q''_L are essentially the same as t, t'_L, and q'_L, respectively. This concludes the proof. □

Remember Equation (1), 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 $\{r\mathbf{G}_1 \oplus e\}$ is pseudorandom for every PPT algorithm. Thus, even if $m\mathbf{G}_2$ is not pseudorandom, c looks random for every PPT algorithms. The proof of the following theorem utilized this fact straight forwardly. This proof is similar in nature to the proof of Lemma 2, we provide it in Appendix A.3.

Theorem 1. *If there exists a probabilistic algorithm A which runs in time t and such that $\mathtt{Adv}^{\mathrm{sem}}_{A,\mathrm{ME}}(k) \geq \delta$ then*

$$2 \cdot \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t'_G) + 4(n+1) \cdot \mathtt{Adv}^{\mathrm{one-way}}_{\mathrm{LPN}_\theta}(k_1, t'''_L, q'''_L) \geq \delta.$$

Here t'''_L, q'''_L and t'_G are essentially the same as t_L, q_L, and t, respectively.

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 c'. One of the fastest attacks which computes m_l from c' is "finding-low-weight-codeword" attack [5, 6], and its time-complexity is estimated as [14]

$$\binom{n}{k_1+1} \cdot \binom{n-w}{k_1+1}^{-1}. \qquad (7)$$

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 (7). The details on this attack with exact time complexity estimations can be found in [6].

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 cryptosystem IND-CPA secure. To prove the security of the Niederreiter cryptosystem with the similar padding, we can utilize the result of [10] instead of Lemma 1 in [13]. It is worth noting that both of these works do not allow tight reductions. Improving the results of [13] and [10] is an open problem.

Another interesting open question, in the light of [23], 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 cryptosystem without employing random oracles.

References

1. 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.

2. 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.
3. M. Bellare, P. Rogaway, “Random Oracles are Practical: a Paradigm for Designing Efficient Protocols,” Proc. CCS, pp.62–73, 1993.
4. M. Bellare, P. Rogaway, “Optimal Asymmetric Encryption - How to Encrypt with RSA,” Proc. EUROCRYPT ’94, LNCS 950, pp. 92–111, 1995.
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. A. Canteaut, N. Sendrier, “Cryptanalysis of the Original McEliece Cryptosystem,” Proc. Asiacrypt ’98, LNCS 1514, pp.187–199, 1998.
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. O. Goldreich, L.A. Levin, “ A Hard-Core Predicate for all One-Way Functions,” Proc. STOC ’89, pp.25–32, 1989.
12. S. Goldwasser, S. Micali, “Probabilistic Encryption,” J. Comp. Syst. Sci. 28:270–299, 1984.
13. J. Katz, J.S. Shin, “Parallel and Concurrent Security of the HB and HB$^+$ Protocols,” Cryptology ePrint Archive: Rep.No. 461, 2005.
14. K. Kobara, H. Imai, “Semantically Secure McEliece Public-Key Cryptosystems - Conversions for McEliece PKC -,” Proc. PKC ’01, LNCS 1992, pp.19–35, 2001.
15. R.J. McEliece, “The Theory of Information and Coding (Vol. 3 of The Encyclopedia of Mathematics and Its Applications.), Reading, Mass., Addison-Wesley, 1977.
16. R.J. McEliece, “A Public-Key Cryptosystem Based on Algebraic Coding Theory,” Deep Space Network Progress Rep., 1978.
17. H. Niederreiter, “Knapsack-type Cryptosystems and Algebraic Coding Theory,” Prob. of Control and Inf. Theory, vol. 15(2), pp.159–166, 1986.
18. P. Paillier, “Public-Key Cryptosystem Based on Discrete Logarithm Residues,” Proc. Eurocrypt ’99, LNCS 1592, pp.223–238, 1999
19. E. Petrank, R.M. Roth, “Is Code Equivalence Easy to Decide?,” IEEE Trans. Inf. Theory, Vol.43, pp.1602–1604, 1997.
20. O. Regev, “On Lattices, Learning with Errors, Random Linear Codes, and Cryptography”, Proc. 37th STOC, pp. 84-93, 2005.
21. N. Sendrier, “Finding the Permutation Between Equivalent Linear Codes: The Support Splitting Algorithm,” IEEE Trans. Inf. Theory, 46(4), pp.1193–1203, 2000.
22. M. Suzuki, K. Kobara, H. Imai, “Privacy Enhanced and Light Weight RFID System without Tag Synchronization and Exhaustive Search,” IEEE SMC, Taipei, 2006.
23. 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.
24. E. Fujisaki and T. Okamoto, “Secure Integration of Asymmetric and Symmetric Encryption Schemes”, Proc. Crypto ’99, LNCS 1666, pp. 537–554, 1999.
25. D. Pointcheval, “Chosen-Ciphertext Security for any One-Way Cryptosystem”, Proc. PKC ’00, LNCS 1751, pp. 129–146, 2000.
26. V. Shoup, “OAEP reconsidered,” CRYPTO ’01, LNCS 2139, pp. 239–259, 2001.

Appendix A

A.1 McEliece Public-Key Cryptosystem

The McEliece cryptosystem [16] 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 outputs $pk = (\mathbf{G}, w)$ 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 $e \in \{0,1\}^n$ is a random vector of weight w.
- The decryption algorithm: The polynomial-time 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 $m = (m\mathbf{S})\mathbf{S}^{-1}$.

A.2 Security of the McEliece Cryptosystem

Generally, we can categorize the attacks to the McEliece cryptosystem into the following two cases:

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

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

If we employ Goppa codes on $\mathbb{F}_2$ from codes on $\mathbb{F}_{2^m}$ then an efficient algorithm which can extract the secret-key from the public-key in the McEliece cryptosystem has not been founded. Moreover, there is no algorithm which can efficiently distinguish the matrix defined by the public-key of the McEliece cryptosystem and the same size random matrix. 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 $\mathbf{G}$ are "permutation equivalent" or not by using the *support splitting algorithm* [21], 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 [19]. From this observation, we utilize the conjecture that these two matrices are indistinguishable for any PPT algorithm. We define this formally in Section 3.1.

A.3 Proof of Theorem 1

We construct a distinguisher D from the adversary A. We show that if A breaks the semantic security of the padded McEliece with non-negligible probability then D distinguishes $s_1 = r\mathbf{G}_1 \oplus e$ and s_0 as defined in Lemma 2 with non-negligible probability.

We construct an algorithm D as follows:

```
$D(pk, \tilde{s})$
Run $A_1(pk)$ to obtain $(m_0, m_1)$
$b \leftarrow \mathcal{U}_1$
Define $c = \tilde{s} \oplus m_b\mathbf{G}_2$
Run $A_2(c)$ 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 $r\mathbf{G}_1 \oplus e$. We will say that A succeeds if $b' = b$ (and denote this event by Succ) under the event Real occurs, and we denote this probability as $\Pr_A[\mathsf{Succ}]$. Note that, we know from Lemma 2 that

$$|\Pr[D = 1 \mid \mathsf{Real}] - \Pr[D = 1 \mid \mathsf{Rand}]|$$

is negligible.

We claim that $\Pr[D = 1 \mid \mathsf{Real}] = \Pr_A[\mathsf{Succ}]$. To see this, note that when Real occurs we have $\tilde{s} = s_1 = r\mathbf{G_1} \oplus e$. But then s_1 is distributed exactly as they would be in a real execution. Since D outputs 1 iff A 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 $\mathcal{U}_n$. Therefore, $\tilde{s} \oplus m_b\mathbf{G_2}$ given to A is uniformly distributed in $\mathcal{U}_n$ as well. This means that A obtains no information related to b. Since D outputs 1 iff A succeeds, we can conclude that $\Pr[B = 1 \mid \mathsf{Rand}] = \frac{1}{2}$.

By combining these results, we obtain

$$\Pr_A[\mathsf{Succ}] - 1/2 \leq 2 \cdot \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t'_G) + 4(n+1) \cdot \mathtt{Adv}^{\mathrm{one-way}}_{\mathrm{LPN}_\theta}(k_1, t'''_L, q'''_L)$$
$$\delta \leq 2 \cdot \mathtt{Adv}^{\mathrm{ind}}_{\mathbf{G}}(k, t'_G) + 4(n+1) \cdot \mathtt{Adv}^{\mathrm{one-way}}_{\mathrm{LPN}_\theta}(k_1, t'''_L, q'''_L).$$

The remaining part is estimating the amount of t'''_L, q'''_L, and t'_G but it is easy to see that these are essentially the same as t''_L, q''_L, and t_G, respectively. Combining all of them together, these are essentially the same as t_L, q_L and t, respectively. This concludes the proof.