An Efficient CCA2-Secure Variant of the McEliece Cryptosystem in the Standard Model

Roohallah Rastaghi

Advanced Intelligent Signal Processing Center, Tehran, Iran

r.rastaghi59@gamail.com

Abstract

Recently, a few chosen-ciphertext secure (CCA2-secure) variants of the McEliece public-key encryption (PKE) scheme in the standard model were introduced. All the proposed schemes are based on encryption repetition paradigm and use general transformation from CPA-secure scheme to a CCA2-secure one. Therefore, the resulting encryption scheme needs *separate* encryption and has *large* key size compared to the original scheme, which complex public key size problem in the code-based PKE schemes. Thus, the proposed schemes are not sufficiently efficient to be used in practice.

In this work, we propose an efficient CCA2-secure variant of the McEliece PKE scheme in the standard model. The main novelty is that, unlike previous approaches, our approach is a generic conversion and can be applied to *any* one-way trapdoor function (OW-TDF), the lowest-level security notion in the context of public-key cryptography, resolving a big fundamental and central problem that has remained unsolved in the past two decades.

Keywords: Post-quantum cryptography, McEliece cryptosystem, Permutation algorithm, CCA2 security, Standard model.

1 Introduction

Post-quantum cryptography has obtained great attention in recent years. Code-based cryptography holds a great promise for the post-quantum cryptography, as it enjoys very strong security proofs based on average-case hardness [22], relatively fast and efficient encryption/decryption nature, as well as great simplicity. In the context of code-based cryptography, there are two well-known public-key encryption (PKE) schemes, namely McEliece [13] and Niederreiter [15] PKE schemes. The McEliece encryption scheme was the first PKE scheme based on linear error-correcting codes. It has a very fast and efficient encryption procedure, but it has one big flaw: the size of the public key. Recently, how to reduce the public key size and how to secure the parameter choice in the code-based cryptography are deeply explored [1, 2, 3, 9, 14].

Semantic security (a.k.a indistinguishability) against adaptive chosen ciphertext attacks (CCA2 security) is one of the strongest known notions of security for the PKE schemes was introduced by Rackoff and Simon [20]. It is possible to produce CCA2-secure variants of the code-based PKE schemes in the random oracle model [4, 11, 12], however, CCA2 security in the standard model has not been widely discussed. To the best of our knowledge, only a few papers have touched this research issue.

1.1 Related work

There are mainly two class of CCA2-secure code-based PKE schemes in the standard model.

- *CCA-secure schemes based on syndrome decoding problem.* Freeman *et al.* [10] used Rosen-Segev approach [21] to introduce a correlation-secure trapdoor function related to the hardness of syndrome decoding. Their construction is based on the Niederreiter PKE scheme. Very recently, Preetha Mathew *et al.* [19] proposed a somewhat efficient variant of the Niederreiter scheme based on lossy trapdoor functions [17], which avoids encryption repetition paradigm.

- *CCA-secure schemes based on general decoding problem.* The first CCA2-secure variants of the McEliece cryptosystem was introduced by Dowsley *et al.* [5]. They proposed a scheme that resembles the Rosen-Segev approach trying to apply it to the McEliece PKE scheme. Their construction has some ambiguity. The scheme does not rely on a collection of functions but instead defines a structure called k-*repetition* PKE scheme. This is essentially an application of k-samples of the PKE to the *same* input, in which the decryption algorithm also includes a verification step on the k outputs. The encryption algorithm produces a signature directly on the McEliece ciphertexts instead of introducing a random vector x as in the original Rosen-Segev scheme; therefore a CPA-secure variant of the McEliece cryptosystem is necessary to achieve CCA2 security [18]. Very recently, Döttling *et al.* [6] showed that Nojima *et al.* [16] randomized version of the McEliece cryptosystem is k-repetitions CPA-secure and, as we mentioned earlier, it can obtain CCA2 security by using a strongly unforgeable one-time signature scheme. In a subsequent work, Persichetti [18] proposed a CCA2-secure PKE scheme based on the McEliece assumptions using the original Rosen-Segev approach.

1.2 Motivation

To date, as we stated above, all the proposed CCA2-secure code-based PKE schemes in the standard model are based on either lossy and correlation-secure trapdoor functions or k-repetitions encryption paradigm. Therefore, the resulting encryption schemes are not efficient as they need to run encryption/decryption algorithms several times and use a strongly unforgeable one-time signature scheme to handle CCA2 security related issues. Moreover, in these schemes, excluding the keys of the signature scheme, the public/secret keys are $2k$-times larger than the public/secret keys of the original scheme, which complex the public key length problem in the code-based PKE schemes. Although the Preetha Mathew *et al.*'s scheme [19] avoids k-repetitions paradigm, it yet needs to run encryption/decryption algorithms 2-times and the public/secret keys are larger than the original Niederreiter scheme. Further, it also uses a strongly unforgeable one-time signature scheme to achieve CCA2 security, and so needs separate encryption. Hence, how to design an efficient CCA2-secure code-based encryption scheme in the standard model is still worth of investigation. This motivates us to investigate new approach for construction efficient such schemes in the standard model without using encryption repetition and generic transformation from CPA-secure schemes to a CCA2-secure one.

1.3 Our Contributions

To tackle the above challenging issues, we introduce a randomized encoding algorithm called PCA and use it along with the McEliece PKE scheme to construct a CCA2-secure PKE scheme in the standard model. Our contributions in this paper are:

- The main novelty is that our construction is a generic conversion and can be applied to any low-level primitive. To further demonstrate the usefulness of our approach, in Section 4 we also introduce *direct* "black-box" construction of a CCA2-secure PKE scheme from any TDF in the standard model, resolving a big fundamental and central problem in the context of public-key cryptography that has remained unsolved in the past two decades.

- Our proposed scheme is more efficient, the publick/secret keys are as in the original scheme and the encryption/decryption complexity are comparable to the original scheme.
- This novel approach leads to the elimination of the encryption repetition and using strongly unforgeable one-time signature scheme.
- This scheme can be used for encryption of long length messages without employing the hybrid encryption method and symmetric encryption.

Organisation. In the next section, we briefly explain some mathematical background and definitions. Then, in Section 3, we introduce our proposed scheme. Finally, a generalized construction based on OW-TDFs will be given in Section 4.

2 Preliminary

2.1 Notation

We represent a binary string in general by bold face letter such as $\mathbf{x} = (x_1, \dots x_n)$. Regular small font letter x denotes its corresponding *decimal* value, that is $x = \sum_{i=1}^{n} x_i 2^{(n-i)}$ and $|\mathbf{x}|$ denotes its binary length. If $k \in \mathbb{N}$ then $\{0,\, 1\}^k$ denote the set of k-bit strings, 1^k denote a string of k ones and $\{0,\, 1\}^*$ denote the set of bit strings of finite length. $y \leftarrow x$ denotes the assignment to y of the value x. For a set S, $s \leftarrow S$ denote the assignment to s of a uniformly random element of S. For a deterministic algorithm $\mathcal{A}$, we write $x \leftarrow \mathcal{A}^{\mathcal{O}}(y,\, z)$ to mean that x is assigned the output of running $\mathcal{A}$ on inputs y and z, with access to oracle $\mathcal{O}$. If $\mathcal{A}$ is a probabilistic algorithm, we may write $x \leftarrow \mathcal{A}^{\mathcal{O}}(y,\, z,\, R)$ to mean the output of $\mathcal{A}$ when run on inputs y and z with oracle access to $\mathcal{O}$ and using the random coins R. We denote by $\Pr[E]$ the probability that the event E occurs. If a and b are two strings of bits, we denote by $a \| b$ their concatenation. $\mathsf{Lsb}_{x_1}(a)$ means the right x_1 bits of a and $\mathsf{Msb}_{x_2}(a)$ means the left x_2 bits of a.

Since the proposed cryptosystem is code-based, a few notations regarding coding theory are introduced. Let $\mathbb{F}_2$ be the finite field with 2 elements $\{0, 1\}$, $k \in \mathbb{N}$ be a security parameter. A binary linear-error correcting code $\mathcal{C}$ of length n and dimension k or an $[n, k]$-code is a k-dimensional subspace of $\mathbb{F}_2^n$. Elements of $\mathbb{F}_2^n$ are called words, and elements of $\mathcal{C}$ are called codewords. If the minimum hamming distance between any two codewords is d, then the code is a $[n, k, d]$ code. The Hamming weight of a codeword $\mathbf{x}$, $\mathsf{wt}(\mathbf{x})$, is the number of non-zero bits in the codeword. For $t \leq \lfloor \frac{d-1}{2} \rfloor$, the code is said to be t-error correcting if it detects and corrects errors of weight at most t. Hence, the code can also be represented as a $[n, k, 2t + 1]$ code. The generator matrix $\mathsf{G} \in \mathbb{F}_2^{k \times n}$ of a $[n, k]$ linear code $\mathcal{C}$ is a matrix of rank k whose rows span the code $\mathcal{C}$.

2.2 Definitions

Definition 1 (Trapdoor functions). A trapdoor function family is a triple of algorithms $\mathsf{TDF} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$, where Tdg is probabilistic and on input 1^k generates an evaluation/trapdoor key-pair $(ek, td) \leftarrow \mathsf{Tdg}(1^k)$. $\mathsf{F}(ek, \cdot)$ implements a function $f_{ek}(\cdot)$ over $\{0, 1\}^k$ and $\mathsf{F}^{-1}(td, \cdot)$ implements its inverse $f^{-1}(\cdot)$.

Definition 2 (One-wayness). Let $\mathcal{A}$ be an inverter and define its $\mathsf{OW\text{-}advantage}$ against TDF as

$$\mathrm{Adv}^{ow}_{\mathsf{TDF},\mathcal{A}}(k) = \Pr\left[x = x' : \begin{array}{l} (ek, td) \leftarrow \mathsf{Tdg}(1^k); x \leftarrow \{0,1\}^k \\ y \leftarrow \mathsf{F}(ek, x); x' \leftarrow \mathcal{A}(ek, y) \end{array} \right].$$

Trapdoor function TDF is one-way if $\mathrm{Adv}^{ow}_{\mathsf{TDF},\mathcal{A}}(k)$ is negligible for every PPT inverter $\mathcal{A}$.

Definition 3 (Circular Shift). A circular (cyclic) shift is the operation of rearranging the components in a string circularly with a prescribed number of positions. Thus, a q-position circular shift (or circular q-shift) defines as the operation in which the i-th sample, s_i, replace with the $(i + q \mod n)$-th sample in a n sample ensemble. We denote this operation by $\mathsf{CS}_{q,n}(s_i) = s_{(i+q \mod n)}$, $1 \le i \le n$.

Definition 4 (General Decoding Problem). Given a generator matrix $\mathsf{G} \in \mathbb{F}_2^{k\times n}$ and a word $\mathbf{m} \in \mathbb{F}_2^n$, find a codeword $\mathbf{c} \in \mathbb{F}_2^k$ such that $\mathbf{e} = \mathbf{m} - \mathbf{c}\mathsf{G}$ has Hamming weight $\mathsf{wt}(\mathbf{e}) \le t$.

Definition 5 (General Decoding Assumption). Let $\mathcal{C}$ be an $[n, k, d]$-binary linear code defined by a $k \times n$ generator matrix G with the minimal distance d, and $t \le \lfloor \frac{d-1}{2} \rfloor$. An adversary $\mathcal{A}$ that takes an input of a word $\mathbf{m} \in \mathbb{F}_2^n$, returns a codeword $\mathbf{c} \in \mathbb{F}_2^k$. We consider the following random experiment on GDP problem.

$$\begin{aligned}
&\mathbf{Exp}^{\mathrm{GDP}}_{\mathcal{A}} :\\
&\quad \mathbf{c} \in \mathbb{F}_2^k \ \leftarrow \ \mathcal{A}(\mathsf{G}, \mathbf{m} \in \mathbb{F}_2^n)\\
&\quad \text{if } \ \mathbf{x} = \mathbf{m} - \mathbf{c}\mathsf{G} \text{ and } \mathsf{wt}(\mathbf{x}) \le t\\
&\quad \text{then } b \leftarrow 1, \text{ else } b \leftarrow 0\\
&\text{return } b.
\end{aligned}$$

We define the corresponding success probability of $\mathcal{A}$ in solving the GDP problem via

$$\mathbf{Succ}^{\mathrm{GDP}}_{\mathcal{A}} = \Pr[\mathbf{Exp}^{\mathrm{GDP}}_{\mathcal{A}} = 1].$$

Let $\tau \in \mathbb{N}$ and $\varepsilon \in [0, 1]$. We call GDP to be (τ, ε)-secure if no polynomial algorithm $\mathcal{A}$ running in time τ has success $\mathbf{Succ}^{\mathrm{GDP}}_{\mathcal{A}} \ge \varepsilon$.

Definition 6 (Public-key encryption). A public-key encryption (PKE) scheme is a triple of probabilistic polynomial time (PPT) algorithms (Gen, Enc, Dec) such that:

- Gen is a probabilistic polynomial time key generation algorithm which takes a security parameter 1^n as input and outputs a public key pk and a secret-key sk. We write $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$. The public key specifies the message space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$.

- Enc is a (possibly) probabilistic polynomial time encryption algorithm which takes as input a public key *pk,* a $m \in \mathcal{M}$ and random coins r, and outputs a ciphertext $C \in \mathcal{C}$. We write $\mathsf{Enc}(pk, m; r)$ to indicate explicitly that the random coins *r* is used and $\mathsf{Enc}(pk, m)$ if fresh random coins are used.

- Dec is a deterministic polynomial time decryption algorithm which takes as input a secret-key *sk* and a ciphertext $C \in \mathcal{C}$, and outputs either a message $m \in \mathcal{M}$ or an error symbol $\perp$. We write $m \leftarrow \mathsf{Dec}(C,\ sk)$.

- (Completeness) For any pair of public and secret-keys generated by Gen and any message $m \in \mathcal{M}$ it holds that $\mathsf{Dec}(sk,\ \mathsf{Enc}(pk, m; r)) = m$ with overwhelming probability over the randomness used by Gen and the random coins *r* used by Enc.

Definition 7 (CCA2 security). A public-key encryption scheme is secure against adaptive chosen-ciphertext attacks (i.e. CCA2-secure) if the advantage of any two-stage PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the following experiment is negligible in the security parameter k:

$$\begin{aligned}
&\mathbf{Exp}^{\mathsf{cca2}}_{\mathrm{PKE},\mathcal{A}}(k):\\
&\quad (pk, sk) \leftarrow \mathsf{Gen}(1^k)\\
&\quad (m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(pk) \quad \text{s.t.} \quad |m_0| = |m_1|\\
&\quad b \leftarrow \{0,1\}\\
&\quad C^* \leftarrow \mathsf{Enc}(pk, m_b)\\
&\quad b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(sk,\cdot)}(C^*, \mathsf{state})\\
&\quad if\, b\, =\, b^{'}\, return\, 1,\; else\, return\, 0.
\end{aligned}$$

The attacker may query a decryption oracle with a ciphertext C at any point during its execution, with the exception that $\mathcal{A}_2$ is not allowed to query $\mathrm{Dec}(sk,\ .)$ with "challenge" ciphertext C^*. The decryption oracle returns $b^{'} \leftarrow \mathcal{A}_2^{\mathrm{Dec}(sk,\cdot)}(C^*,\ state)$. The attacker wins the game if $b = b'$ and the probability of this event is defined as $\Pr[\mathrm{Exp}\,{}^{cca2}_{\mathrm{PKE},\,\mathcal{A}}\,(k)]$. We define the advantage of $\mathcal{A}$ in the experiment as

$$\mathsf{Adv}^{\mathsf{Ind-cca2}}_{\mathrm{PKE},\mathcal{A}}\,(k) = \left|\Pr[\mathrm{Exp}^{\mathsf{cca2}}_{\mathrm{PKE},\mathcal{A}}\,(k) = 1] - \frac{1}{2}\right|. \tag{1}$$

2.3 The McEliece PKE scheme

The McEliece PKE consists of a triplet of probabilistic polynomial time algorithms $(\mathsf{Gen}_{\mathrm{McE}}, \mathsf{Enc}_{\mathrm{McE}}, \mathsf{Dec}_{\mathrm{McE}})$.

- **System parameters.** $q, n, t \in \mathbb{N}$, where $t \ll n$.
- **Key Generation.** $\mathsf{Gen}_{\mathrm{McE}}$ take as input security parameter 1^k and generate the following matrices:
  1. A $k \times n$ generator matrix G of a code $\mathcal{G}$ over $\mathbb{F}_q$ of dimension k and minimum distance $d \geq 2t+1$. (A binary irreducible Goppa code in the original proposal).
  2. A $k \times k$ random binary non-singular matrix S
  3. A $n \times n$ random permutation matrix P.

  Then, Gen compute the $k \times n$ matrix $\mathsf{G}^{pub} = \mathsf{SGP}$ and outputs a public key pk and a secret key sk, where
  $$pk = (\mathsf{G}^{\mathrm{pub}}, t) \quad \text{and} \quad pk = (\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$$
  where $D_{\mathcal{G}}$ is an efficient decoding algorithm for $\mathcal{G}$.
- **Encryption.** $\mathsf{Enc}_{\mathrm{McE}}(pk)$ takes plaintext $\mathbf{m} \in \mathbb{F}_2^k$ as input and randomly choose a vector $\mathbf{e} \in \mathbb{F}_2^n$ wit Hamming weight $\mathsf{wt}(\mathbf{e}) = t$ and computes the ciphertext **c** as follows.
  $$\mathbf{c} = \mathbf{m}\mathsf{G}^{\mathrm{pub}} \oplus \mathbf{e}.$$
- **Decryption.** To decrypt a ciphertext **c**, $\mathsf{Dec}_{\mathrm{McE}}(sk, \mathbf{c})$ first calculates
  $$\mathbf{c}\mathsf{P}^{-1} = (\mathbf{m}\mathsf{S})\mathsf{G} \oplus \mathbf{e}\mathsf{P}^{-1}$$

and then apply the decoding algorithm $D_{\mathcal{G}}$ to it. If the decoding succeeds, output

$$\mathbf{m} = (\mathbf{m}\mathsf{S})\mathsf{S}^{-1}.$$

Otherwise, output $\perp$.

There are two computational assumptions underlying the security of the McEliece scheme.

Assumption 1 (Indistinguishability[1]). *The matrix* G *output by* Gen *is computationally indistinguishable from a uniformly chosen matrix of the same size.*

Assumption 2 (Decoding hardness). *Decoding a random linear code with parameters* n, k, w *is hard.*

Note that Assumption 2 is in fact equivalent to assuming the hardness of GDP. It is immediately clear that the following corollary is true.

Corollary 1. *Given that both the above assumptions hold, the McEliece cryptosystem is one-way secure under passive attacks.*

3 The proposed cryptosystem

In this section, we introduce our conversion. Our construction consists of two parts: 1) Encryption of random coins r using the original McEliece PKE scheme; 2) Randomized encoding of the plaintext, where randomization is done using r (that used for consistency check) based on a heuristic encoding algorithm. Encoding includes a permutation and combination on the message bits that performs using an algorithm called *permutation combination algorithm* (PCA).

3.1 PCA encoding algorithm

To encode message $\mathbf{m} \in \{0,1\}^n$ with $n \gg k$, we firstly pick coins $\mathbf{r} \in \{0,1\}^k, \mathbf{r} \neq 0^k, 1^k$ uniformly at random, where k is the security parameter. Let $\mathsf{wt}(\mathbf{r}) = h$ be its Hamming weight. We divide $\mathbf{m}$ into l blocks $(b_1\|\ldots\|b_l)$ with equal binary length $\lceil n/l \rceil$, where $l = h$ if $h \geq k-h$, else $l = k-h$. If $l \nmid n$, then we should pad $\mathbf{m}$. In such cases, we can sample a random binary string (RBS) from $\mathbf{r}$, say $\mathsf{RBS} = \mathsf{Msb}_{l\lceil n/l \rceil - n}(\mathbf{r})$, and pad it on the right of $\mathbf{m}$[2]. Therefore, if $l \mid n$ then $v = n/l$, $\mathsf{RBS} = \varphi$ (the empty set) and $b_l = \mathsf{Lsb}_v(\mathbf{m})$, else $v = \lceil n/l \rceil$, RBS is a random string with length $l\lceil n/l \rceil - n$ which sampled from $\mathbf{r}$ and $d_l = \mathsf{Lsb}_{(n-(l-1)\lceil n/l \rceil)}(\mathbf{m})\|\mathsf{RBS}$. Now, we perform a secure permutation on the message blocks $b_i, 1 \leq i \leq l$ with the following algorithm.

First, note that any positive integer $s, 1 \leq s \leq l! - 1$ uniquely can be shown as

$$s = u_1 \times (l-1)! + u_2 \times (l-2)! + \cdots + u_l \times 0, \quad 0 \leq u_i \leq l - i.$$

Note that based on this definition we have $u_l = 0$. The sequence $U_s = (u_1, \ldots, u_l)$ is called *factorial carry value* of s. We define original sequence $\mathbf{m}^0$ as $\mathbf{m}^0 = (b_1\|\ldots\|b_l)$. Recombine all elements of the original sequence $\mathbf{m}^0$ obtain $l! - 1$ new sequences $\mathbf{m}^1, \ldots, \mathbf{m}^{(l!-1)}$, which any sequence owns a corresponding factorial carry value. Using the factorial carry value of s, we can efficiently obtain any sequence $\mathbf{m}^s, 1 \leq s \leq l! - 1$ with the following algorithm.

Algorithm 3.1 (PCA encoding algorithm).
Input: Message $\mathbf{m} \in \{0,1\}^n$, coins $\mathbf{r} \in \{0,1\}^k$ and integer $s, 1 \leq s \leq l! - 1$.

[1] This statement is not true in general. See [7, 8] for instance.

[2] Note that since $l \in [\lceil k/2 \rceil, k-1]$, the length of sampled RBS, i.e. $l\lceil n/l \rceil - n$, is smaller than k, i.e. the length of $\mathbf{r}$. Therefor, for all cases we do not have any problem for sampling RBS from $\mathbf{r}$.

Output: Encoded message $\mathbf{y}' = \mathbf{m}^s = (b'_1\| \dots \|b'_l)$.

SETUP:

1. $h \leftarrow \mathsf{wt}(\mathbf{r})$. If $2h \geq k$ then $l \leftarrow h$, else $l \leftarrow k - h$.
2. If $l \mid n$ then set $\mathsf{RBS} = \varphi$; otherwise, $\mathsf{RBS} \leftarrow \mathsf{MSb}_{(l\lceil n/l\rceil - n)}(\mathbf{r})$.
3. $\mathbf{m}' \leftarrow \mathbf{m}\|\mathsf{RBS}$ and divide $\mathbf{m}'$ into l blocks $(b_1\| \dots \|b_l)$ with equal length $v = \lceil n/l \rceil$.

PERMUTATION:

1. Write s as $s = \sum_{i=1}^{l-1} u_i\,(l-i)! + u_l \times 0, \quad 0 \leq u_i \leq l - i$.
2. For $1 \leq i \leq l$:
   If $u_i = 0$, then $b'_i \leftarrow b_i$;
   Else, $b'_i \leftarrow b_{i+u_i}$, and for $1 \leq j \leq u_i$:
   $b'_{i+j} \leftarrow b_i$;
3. Return $\mathbf{y}' = \mathbf{m}^s = (b'_1\| \dots \|b'_l)$.

Note that the *number* and the *length* of the message blocks are variable and changed by $\mathbf{r}$.
It is clear that the above encoding algorithm satisfies correctness. Namely, for any $(\mathbf{m}, \mathbf{r})$ we have

$$\forall \mathbf{m} \in \{0,1\}^n, \mathbf{r} \in \{0,1\}^k \text{ and } s \in \mathbb{N} : \mathsf{PCA}^{-1}(\mathsf{PCA}(\mathbf{m}, \mathbf{r}, s), \mathbf{r}, s) = \mathbf{m}.$$

We illustrate PCA encoding algorithm with a small example. Suppose $\mathbf{m} = (m_1, \dots, m_{512})$ and $\mathbf{r} = (r_1, \dots, r_{25})$ with $h = \sum_{i=1}^{25} r_i = 12$. Since $2h < k$, thus $l = k - h = 13$. Therefore, the algorithm divides $\mathbf{m}$ into 13 blocks with equal length $v = \lceil n/l \rceil = \lceil 512/13 \rceil = 40$. In this case, we have to sample a string with length $l\lceil n/l\rceil - n = 8$ from $\mathbf{r}$ and pad it on the right of $\mathbf{m}$. Therefore, we have $\mathbf{m}' = (\underbrace{m_1, \dots, m_{40}}_{b_1}\|\underbrace{m_{41}, \dots, m_{80}}_{b_2}\| \dots \|\underbrace{m_{481}, \dots, m_{512}\|r_1, \dots r_8}_{b_{13}})$.
We choose integer $s, 1 \leq s \leq 13! - 1$, say $s = 4819995015$. We have

$$\begin{aligned} 4819995015 = {} & 10 \times 12! + 0 \times 11! + 8 \times 10! + 2 \times 9! + 5 \times 8! + 4 \times 7! + 1 \times 6! \\ & +3 \times 5! + 0 \times 4! + 2 \times 3! + 1 \times 2! + 1 \times 1! + 0 \end{aligned}$$

Thus, the factorial carry value of $\mathbf{m}^{4819995015}$ is $\{10, 0, 8, 2, 5, 4, 1, 3, 0, 2, 1, 1, 0\}$. Compute sequence $D^{4819995015}$ with its factorial carry value $\{10, 0, 8, 2, 5, 4, 1, 3, 0, 2, 1, 1, 0\}$. We have

$$\begin{aligned} & 10 - -\{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}\} \ \rightarrow b_{11} \\ & 0 - -\{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{12}, b_{13}\} \ \rightarrow b_1 \\ & 8 - -\{b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{12}, b_{13}\} \ \rightarrow b_{10} \\ & 2 - -\{b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{12}, b_{13}\} \ \rightarrow b_4 \\ & \quad \vdots \\ & 1 - -\{b_5, b_6, b_{13}\} \ \rightarrow b_6 \\ & 1 - -\{b_5, b_{13}\} \ \rightarrow b_{13} \\ & 0 - -\{b_5\} \ \rightarrow b_5 \end{aligned}$$

Therefore, the permutation of sequence $\mathbf{m}^{4819995015}$ is $(b_{11}\|b_1\|b_{10}\|b_4\|b_8\|b_7\|b_3\|b_9\|b_2\|b_{12}\|b_6\|b_{13}\|b_5)$.

3.2 The proposed scheme

Now, we are ready to define our conversion. Given a McEliece PKE scheme $\Pi = (\mathsf{Gen}_{\mathrm{McE}}, \mathsf{Enc}_{\mathrm{McE}}, \mathsf{Dec}_{\mathrm{McE}})$, we transform it into CCA2-secure PKE scheme $\Pi' = (\mathsf{Gen}_{cca2}, \mathsf{Enc}_{cca2}, \mathsf{Dec}_{cca2})$.

Key Generation. On security parameter k, $\mathsf{Gen}_{cca2}(1^k)$ run $(sk_{\mathrm{McE}}, pk_{\mathrm{McE}}) \leftarrow \mathsf{Gen}_{\mathrm{McE}}(1^k)$ to obtain $sk_{\mathrm{McE}} = (\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$ and $pk_{\mathrm{McE}} = (\mathsf{G}^{pub}, t)$ as in subsection 2.3. It also choose target collision resistant (aka universal one-way) hash function $\mathsf{T} : \{0,1\}^k \to \{0,1\}^k$ and pseudorandom generator $G : \{0,1\}^k \to \{0,1\}^n$. $pk = \{(\mathsf{S}, D_{\mathcal{G}}, \mathsf{P}), G, \mathsf{T}\}$ is the public key and $sk = \{(\mathsf{G}^{pub}, t), G, \mathsf{T}\}$ is the secret key.

Encryption. To create ciphertext, encryption algorithm in some cases performs operations in decimal, and in other cases it performs operations in binary representation of the components. When we do operations in decimal, we show components by regular small fonts, and when we perform operations in binary, we show components by bold face fonts. To encrypt message $\mathbf{m} \in \{0,\, 1\}^n$ with $n \gg k$, Enc_{cca2}:

1. Randomly choose error vector $\mathbf{e} \in \{0,1\}^k$ wit Hamming weight $\mathsf{wt}(\mathbf{e}) = t$.
2. Compute $\mathbf{r} = \mathsf{T}(\mathbf{e}) \in \{0,1\}^k, \mathbf{r} \neq 0^k, 1^k$. Let $\mathsf{wt}(\mathbf{r}) = h$. If $2h > k$ then $l \leftarrow h$, else $l \leftarrow k - h$.
3. Compute $\tilde{\mathbf{m}} = \mathbf{m} \oplus G(\mathbf{r})$.
4. Set $s = \sum_{i=1}^{l-1} u_i(l-i)! + u_l \times 0$, where $u_i = (r+i) \mod (l-i), 1 \leq i \leq l-1$ and $u_l = 0$, and run PCA encoding algorithm (Algorithm 3.1) on inputs $(\tilde{\mathbf{m}}, \mathbf{r}, s)$ to generate encoded message $\mathbf{y}' = \mathsf{PCA}(\tilde{\mathbf{m}}, \mathbf{r}, s)$. Note that we have $U_s = (u_1, \ldots, u_{l-1}, 0)$.
5. Perform a circular q-shift on the encoded message $\mathbf{y}'$ and compute sequence $\mathbf{y} = \mathsf{CS}_{q,|\mathbf{y}'|}(\mathbf{y}')$, where $q = r \mod n$. Note n and so $q < |\mathbf{y}'|$.
6. Compute

$$C_1 = (hy + \bar{r})r + z, \quad C_2 = \mathsf{Enc}_{\mathrm{McE}}(pk, \mathbf{r}; \mathbf{e}) = \mathbf{r}\mathsf{G}^{pub} \oplus \mathbf{e},$$

where r, y are the corresponding decimal value of $\mathbf{r}$ and $\mathbf{y}$. $\bar{r}$ is the decimal value of the complement of $\mathbf{r}$ and $z = \sum_{i=1}^{l-1} u_i$.

As we know, in hybrid PKE schemes XOR alone cannot perfectly hide challenge bit to the CCA2 adversary. To handle CCA2 security related issues, we increase *obfuscation* of the XORed message by a) perform a randomized encoding on its bits and b) perform a secure circular shift on the bits of encoded message, whose shift step depends on the value of r. Moreover, we disguise encoded message and conceal its bits by setting $C_1 = (hy + \bar{r})r + z$ in order to *decrease malleability* of the ciphertext. (See Proposition 1). Therefore, the CCA2 adversary to extract any useful information about challenge bit from C_1 must first recover the *same* coins $\mathbf{r}$ that was used to create the ciphertext from the McEliece PKE scheme, which is impossible if the McEliece PKE scheme be secure.

Decryption. To recover message $\mathbf{m}$ from $C = (C_1, C_2)$, Dec_{cca2} perform the following steps.

1. Compute coins $\mathbf{r}$ as $\mathbf{r} = \mathsf{Dec}_{\mathrm{McE}}(sk_{\mathrm{McE}}, C_2)$ and retrieve error vector $\mathbf{e} = C_2 \oplus \mathbf{r}\mathsf{G}^{pub}$
2. Check whether

$$\mathbf{r} \stackrel{?}{=} \mathsf{T}(\mathbf{e}) \tag{2}$$

holds[3] and reject if not (*consistency* check). If it holds compute $\mathsf{wt}(r) = h$. If $2h \geq k$ then $l \leftarrow h$, else $l \leftarrow k - h$.

3. Compute $s = \sum_{i=1}^{l-1} u_i(l-i)!$, $U_s = (u_1, \ldots, u_{l-1}, 0)$ and $z = \sum_{i=1}^{l-1} u_i$, where $u_i = (r+i) \mod (l-i)$.

4. Compute
$$y = \frac{(C_1 - z)/r - \bar{r}}{h}, \tag{3}$$
and reject if y is not a $(l\lceil n/l \rceil)$-bit integer (consistency check). Note that if $l \mid n$ then $|y| = n$.

5. Compute $\mathbf{y}' = \mathsf{CS}^{-1}_{q,|\mathbf{y}|}(\mathbf{y})$, where $\mathbf{y}$ is the binary representation of y. $q = r \mod n$ and $|\mathbf{y}'| = |\mathbf{y}|$.

6. Compute $\tilde{\mathbf{m}} = \mathsf{PCA}^{-1}(\mathbf{y}', \mathbf{r}, s)$.

7. Compute $\mathbf{m}' = \tilde{\mathbf{m}} \oplus G(\mathbf{r})$ and $\mathsf{RBS} = \mathsf{Msb}_{l\lceil n/l \rceil - n}(\mathbf{r})$. Check wether
$$\mathsf{Lsb}_{l\lceil n/l \rceil - n}(\mathbf{m}') \stackrel{?}{=} \mathsf{RBS} \tag{4}$$
holds and reject if not (consistency check). If it holds output
$$\mathbf{m} = \mathsf{Msb}_{l\lceil n/l \rceil - n}(\mathbf{m}'), \tag{5}$$
else output $\perp$.

Remark 1. Note that if either $\mathbf{y}$ or $\mathbf{r}$ be illegal values, then the decryption algorithm outputs a *random* string.

Proposition 1. The ciphertext C_1 is non-malleable.

Proof. We say that $(\hat{C}_1, C_2)$ is a valid forgery on (C_1, C_2) if $\hat{\mathbf{y}}$ differ from $\mathbf{y}$ in some bits, where $\hat{\mathbf{y}}$ and $\mathbf{y}$ are the corresponding binary representation of $\hat{y} = ((\hat{C}_1 - z)/r - \bar{r})/h$ and $y = ((C_1 - z)/r - \bar{r})/h$ respectively. If the adversary can produce such $\hat{C}_1$, then he can guess challenge bit from $\hat{\mathbf{m}} = \mathsf{Msb}_{l\lceil n/l \rceil - n}(\hat{\tilde{\mathbf{m}}} \oplus G(\mathbf{r}))$. Without loss of generality, we assume $\hat{\mathbf{y}}$ and $\mathbf{y}$ differ in i-th bit. Namely $\hat{\mathbf{y}} = \mathbf{y} \oplus e_i$, where e_i is the i-th unit vector. Thus, we should have $\hat{y} = y \pm 2^i$. That is, $\hat{C}_1 = (h\hat{y} + \bar{r})r + z = C_1 \pm 2^i hr$. Since secret coins $r \in \{0,1\}^k$ is not known to the adversary, thus the probability that the adversary produces a valid forgery on C_1 is negligible and it is 2^{-k}. This is because in the encryption algorithm we set $C_1 = (hy + \bar{r})r + z$. □

Theorem 1. Suppose $\Pi = (\mathsf{Gen}_{\mathrm{McE}}, \mathsf{Enc}_{\mathrm{McE}}, \mathsf{Dec}_{\mathrm{McE}})$ be the McEliece PKE scheme. Then, the proposed scheme $\Pi' = (\mathsf{Gen}_{cca2}, \mathsf{Enc}_{cca2}, \mathsf{Dec}_{cca2})$ is CCA2-secure in the standard model.

Proof. The encryption algorithm uses coins r to encrypt challenge message. In the encryption algorithm, we do not use any cryptographic primitives to be able to reduce CCA security of the proposed PKE scheme to the hardness or security of them. Note that we only use the McEliece PKE scheme to encrypt the coins r and encryption of the challenge message independent of its output.
In the proof of security, we exploit the fact that for a given ciphertext, we can recover the message if we know the *same* encoded message y and randomness r that was used to create the ciphertext. We stress that if

[3] In deterministic code-based PKE schemes such as Niederreiter PKE scheme, we don't need to perform this checking. In these schemes since encryption algorithm is deterministic, each message has *one* pre-image. Therefore if $C_2 \neq C_2^*$, then $\mathbf{r} = \mathsf{Dec}(C_2, sk) \neq \mathsf{Dec}(C_2^*, sk) = \mathbf{r}^*$. But in the McEliece encryption scheme, for i-th and j-th unit vectors e_i and e_j (with $i \neq j$) if $\mathsf{wt}(e, e_i) = 1$ and $\mathsf{wt}(e, e_j) = 0$, then $C_2' = (\mathbf{r}\mathsf{G}^{pub} \oplus \mathbf{e}) \oplus e_i \oplus e_j$ is a correct ciphertext, since the Hamming weight of $\mathbf{e} \oplus e_i \oplus e_j$ is t. Therefore, queried ciphertext of the form $(C_1, C_2 \oplus e_i \oplus e_j)$ may leaks information of the original message. Thus, we need to check well-formedness of the ciphertext and reject such maliciously-formed one.

either y or r is not legal values, then the output of the decryption algorithm is *random*. Thus, the challenge bit is information-theoretically hidden to the CCA2 adversary, and so, his advantage in guessing challenge bit is 0.

Let $C^* = (C_1^*, C_2^*)$ be the challenge ciphertext, where $C_1^* = (h^* y^* + \bar{r^*}) r^* + z^*$ and $\mathbf{y}^* = \mathsf{PCA}(\mathbf{m}_b \oplus G(\mathbf{r}^*), \mathbf{r}^*)$. Denote the secret randomness used to encrypt $\mathbf{m}_b$ by $\mathbf{r}^*$. Assume towards contradiction that there is an efficient adversary $\mathcal{A}$ breaking CCA2 security of the proposed PKE scheme with non-negligible probability. That is, the adversary $\mathcal{A}$ can guess challenge bit with non-negligible probability at least from one of the below cases. Since a decryption query on the challenge ciphertext is forbidden by the CCA2-experiment, thus if $C_1 = C_1^*$, then $C_2 \neq C_2^*$ and vice versa. Therefore, there are three possible cases:

Case1. $C = (C_1, C_2) \neq (C_1^*, C_2^*)$. In this case, the decryption oracle takes as input (C_1, C_2) and compute $\mathbf{r} = \mathsf{Dec}_{\mathrm{McE}}(C_2) \in \{0,1\}^k$. If $\mathbf{r} = \mathbf{r}^*$ while $C_2 \neq C_2^*$, then the decryption oracle will reject in (2). It also computes l, $u_i = (r + i) \mod (l - i), 1 \leq i \leq l - 1$, $s = \sum_{i=1}^{l-1} u_i (l - i)!$ and $z = \sum_{i=1}^{l-1} u_i$. In the worst case, we assume $y = ((C_1 - z)/r - \bar{r})/h$ is a $(n + l\lceil n/l \rceil - n)$-bit integer. That is, there is an integer y such that $C_1 = (hy + \bar{r})r + z$. The decryption oracle computes $\mathbf{y}' = \mathsf{CS}^{-1}_{r \mod n, |\mathbf{y}|}(\mathbf{y})$ and decodes $\mathbf{y}'$ based on recovered coins $\mathbf{r}$ and computed value s. We have $\tilde{\mathbf{m}} = \mathsf{PCA}^{-1}(\mathbf{y}', \mathbf{r}, s) \neq \mathsf{PCA}^{-1}(\mathbf{y}'^*, \mathbf{r}^*, s^*) = \tilde{\mathbf{m}}^*$, even we assume $\mathbf{y}' = \mathbf{y}'^*$. If we also assume condition (6) holds (i.e., $\mathsf{Lsb}_{l\lceil n/l \rceil - n}(\tilde{\mathbf{m}} \oplus G(\mathbf{r})) = \mathsf{Msb}_{l\lceil n/l \rceil - n}(\mathbf{r})$), then the decryption oracle outputs random string $\mathbf{m} = \mathsf{Msb}_{l\lceil n/l \rceil - n}(\tilde{\mathbf{m}} \oplus G(\mathbf{r}))$ in (7). Since $\mathbf{m}$ is a random string, thus challenge bit is information-theoretically hidden to the CCA2 adversary, and so, his advantage to guess challenge bit is 0.

Case2. $C = (C_1^*, C_2 \neq C_2^*)$. In this case, the decryption oracle takes as input (C_1^*, C_2) and compute $\mathbf{r} = \mathsf{Dec}_{\mathrm{McE}}(C_2)$. If $\mathbf{r} = \mathbf{r}^*$ while $C_2 \neq C_2^*$, then the decryption oracle will reject in (2). In the worst case, we assume $y = ((C_1^* - z)/r - \bar{r})/h$ is a $(n + l\lceil n/l \rceil - n)$-bit integer and $\mathbf{y}' = \mathsf{CS}^{-1}_{r \mod n, |\mathbf{y}|}(\mathbf{y}) = \mathbf{y}'^*$. Since $\mathbf{r}$ is illegal, i.e. $\mathbf{r} \neq \mathbf{r}^*$ (and so $s \neq s^*$), thus $\tilde{\mathbf{m}} = \mathsf{PCA}^{-1}(\mathbf{y}', \mathbf{r}, s) \neq \mathsf{PCA}^{-1}(\mathbf{y}'^*, \mathbf{r}^*, s^*) = \tilde{\mathbf{m}}^*$. If we also assume condition (6) is hold, then the decryption algorithm outputs random string $\mathbf{m} = \mathsf{Msb}_{l\lceil n/l \rceil - n}(\tilde{\mathbf{m}} \oplus G(\mathbf{r}))$ in (7). Therefore, challenge bit is information-theoretically hidden to the CCA2 adversary, and so, his advantage to guess challenge bit is 0.

Case3. $C = (C_1 \neq C_1^*, C_2^*)$. In this case, the decryption oracle takes as input (C_1, C_2^*) and compute $\mathbf{r} = \mathsf{Dec}_{\mathrm{McE}}(C_2^*) = \mathbf{r}^*$. It also computes $y = ((C_1 - z^*)/r^* - \bar{r^*})/h^* \neq y^*$ [4]. In the worst case, we assume y is an integer. We consider tree possible cases for y:

a) y is a multiple of y^*. That is, for any $k \in \mathbb{N}, k \neq 1$ we have $y = ky^*$. In this case $|y| = |k||y^*| \neq |y^*| = n + l\lceil n/l \rceil - n$ and the decryption oracle reject in (3).

b) $|\mathbf{y}| = |\mathbf{y}^*|$, and, $\mathbf{y}$ and $\mathbf{y}^*$ are differ from each other only in some bits. In this case, $\mathbf{y}' = \mathsf{CS}^{-1}_{q^*, |\mathbf{y}|}(\mathbf{y})$ and $\mathbf{y}'^* = \mathsf{CS}^{-1}_{q^*, |\mathbf{y}|}(\mathbf{y})$ are also differ from each other only in some bits. Therefore, the CCA2 adversary can guess challenge bit from $\mathbf{m}' = \mathsf{PCA}^{-1}(\mathbf{y}', \mathbf{r}^*, s^*) \oplus G(\mathbf{r}^*)$. Without loss of generality, we can assume $\mathbf{y} = \mathbf{y}^* \oplus e_i$, where e_i is the i-th unit vector. Thus we have $y = y^* \pm 2^i$, where y and y^* are the corresponding decimal value of $\mathbf{y}$ and $\mathbf{y}^*$. Therefore, we should have $C_1 = (y^* h^* + \bar{r^*}) r^* + z^* \pm 2^i h^* r^* = C_1^* \pm 2^i h^* r^*$. Since secret coins $r^* \in \{0,1\}^k$ is not known to the CCA2 adversary, thus the probability that CCA2 adversary produce a forgery on C_1 is negligible and it is 2^{-k}. Therefore, the CCA2 adversary's advantage in this case is negligible (see

[4]In this case we have $y \neq y^*$. If $y = y^*$, then we have $C_1 = (h^* y + \bar{r^*}) r^* + z^* = (h^* y^* + \bar{r^*}) r^* + z^* = C_1^*$, which is a contradiction since a decryption query on the challenge ciphertext is forbidden by the CCA2-experiment.

also Proposition1).

c) $|\mathbf{y}| = |\mathbf{y}^*|$, and, $\mathbf{y} \neq \mathbf{y}^*$ (and so $\mathbf{y}' = \mathsf{CS}^{-1}_{q^*,|\mathbf{y}|}(\mathbf{y})$) is a random string. In this case $\mathbf{m}' = \mathsf{PCA}^{-1}(\mathbf{y}', \mathbf{r}^*, s^*) \oplus G(\mathbf{r}^*)$ also is a random string and the decryption oracle outputs random string $\mathsf{Msb}_{l\lceil n/l\rceil - n}(\mathbf{m}')$. Thus, challenge bit is information-theoretically hidden to the CCA2 adversary, and so, his advantage to guess challenge bit is 0.

From **Case1**, **Case2** and **Case3**, the CCA2 adversary advantage to guess challenge bit is negligible. This contradicts the assumption that the CCA2 adversary can break CCA2 security of the proposed PKE scheme with non-negligible probability. □

3.3 Performance analysis

The performance-related issues can be discussed with respect to the computational complexity of key generation, key sizes, encryption and decryption speed.

The resulting encryption scheme is very efficient. The public/secret keys are roughly as in the original scheme. The time for computing $\mathsf{T}(\cdot), G(\cdot)$ and the time for encoding and decoding is negligible compared to the time for computing $\mathsf{Enc}_{\text{McE}}$ and $\mathsf{Dec}_{\text{McE}}$. Encryption roughly needs one application of $\mathsf{Enc}_{\text{McE}}$, and decryption roughly needs one application of $\mathsf{Dec}_{\text{McE}}$.

As we previously stated, the Niederreiter-based proposed scheme does not need to perform well-formedness checking. Therefore, compared to Freeman *et al.*[10] and Mathew *et al.*[19] schemes, our scheme is more efficient. The comparison of the proposed schemes with existing schemes are presented in Table 1.

Table 1. Comparison with other proposed CCA2-secure code-based cryptosystems

Scheme	**Public-key**	**Secret key**	**Ciphertext** Size	**Encryption Complexity**	**Decryption complexity**
Dowsley and Döttling *et al.*[5, 6]	$2k \times pk_{\text{McE}}$	$2k \times sk_{\text{McE}}$	$k \times \mathsf{Ciph}_{\text{McE}}$	$k \times \mathsf{Enc}_{\text{McE}}+$ $1\,\mathcal{OT}-\mathcal{SS}$	$1\,\mathsf{Ver}_{\mathcal{OT}-\mathcal{SS}}+$ $1 \times \mathsf{Dec}_{\text{McE}}+$ $t \times \mathsf{Enc}_{\text{McE}}$
Freeman *et al.*[10]	$2k \times pk_{\text{Nie}}$	$2k \times sk_{\text{Nie}}$	$k \times \mathsf{Ciph}_{\text{Nie}}$	$k \times \mathsf{Enc}_{\text{Nie}}+$ $1\,\mathcal{OT}-\mathcal{SS}$	$1\,\mathsf{Ver}_{\mathcal{OT}-\mathcal{SS}}+$ $1 \times \mathsf{Dec}_{\text{Nie}}+$ $t \times \mathsf{Enc}_{\text{Nie}}$
Mathew *et al.*[19]	$1\,pk_{\text{Nie}}+$ $1\,(n \times n)$ Matrix	$2 \times sk_{\text{Nie}}$	$2 \times \mathsf{Ciph}_{\text{Nie}}$	$2 \times \mathsf{Enc}_{\text{Nie}}+$ 1 MM+ $1\,\mathcal{OT}-\mathcal{SS}$	$1\,\mathsf{Ver}_{\mathcal{OT}-\mathcal{SS}}+$ $1 \times \mathsf{Dec}_{\text{Nie}}+$ $2 \times \mathsf{Enc}_{\text{Nie}}+$ 1 MM
Proposed Scheme	$\approx 1\,pk_{\text{McE}}$	$\approx 1\,sk_{\text{McE}}$	$\approx 2\,\mathsf{Ciph}_{\text{McE}}$ $+n$	$\approx 1\,\mathsf{Enc}_{\text{McE}}$	$\approx 1\,\mathsf{Dec}_{\text{McE}}$

McE: McEliece cryptosystem, Nie: Niederreiter cryptosystem, Ciph: Ciphertext, Ver: Verification, $\mathcal{OT}-\mathcal{SS}$: Strongly unforgeable one-time signature scheme, P: Product, D: Division, MM: Matrix Multiplication, PCA: Permutation Combination Algorithm (algorithm 3.1), PCA^{-1}: Reverse Permutation Combination Algorithm and $t \leq k$.

4 General construction from TDFs

Devising public-key encryption schemes which are secure against chosen ciphertext attack from low-level primitives has been the subject of investigation by many researchers. Currently, the minimal security assumption on trapdoor functions need to obtain CCA2-secure PKE schemes, in terms of "black-box" implications, is that of *adaptivity* was proposed by Kiltz, Mohassel and O'Neill in Eurocrypt 2010 [11]. They proposed a black-box *one*-bit CCA2-secure encryption scheme and then apply a transform of Myers and shelat [16] from one-bit to multi-bit CCA-secure encryption scheme. The Myers-shelat conversion is not efficient; it uses encryption reputation paradigm along with a strongly unforgeable one-time signature scheme to handle CCA2 security related issues. Therefore, the resulting encryption scheme needs *separate* encryption and it is not sufficiently efficient to be used in practice.

Here, we give *direct* black-box construction of a CCA2-Secure PKE scheme from TDFs. Our construction is similar to the construction of Section 3. We only need to replace the underlying code-based PKE scheme with a OW-TDF. Let $\mathsf{TDF} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$ be an *injective* TDF. We construct multi-bit PKE scheme $\mathsf{PKE}[\mathsf{TDF}] = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

Key Generation. On security parameter k, the generator Gen runs Tdg to obtain $(ek, td) \leftarrow \mathsf{Tdg}(1^k)$ and return (ek, td). It also chooses PRG $G : \{0,1\}^k \to \{0,1\}^n$. $pk = (ek, G)$ is the public key and $sk = (td, G)$ is the secret key.

Encryption. On inputs $(\mathbf{m}, ek)$, where $\mathbf{m} \in \{0,\ 1\}^n$, Enc perform as follows:

1. Choose coins $\mathbf{r} \in \{0,1\}^k, \mathbf{r} \neq 0, 1^k$ uniformly at random and let $\mathsf{wt}(\mathbf{r}) = h$. If $2h > k$ then $l \leftarrow h$, else $l \leftarrow k - h$.

2. Compute $\tilde{\mathbf{m}} = \mathbf{m} \oplus G(\mathbf{r})$.

3. Set $s = \sum_{i=1}^{l-1} u_i(l-i)! + u_l \times 0$, where $u_i = (r+i) \mod (l-i), 1 \leq i \leq l-1$ and $u_l = 0$, and run PCA encodding algorithm (Algorithm 3.1) on inputs $(\tilde{\mathbf{m}}, \mathbf{r}, s)$ to generate encoded message $\mathbf{y}' = \mathsf{PCA}(\tilde{\mathbf{m}}, \mathbf{r}, s)$.

4. Perform a circular q-shift on the encoded message $\mathbf{y}'$ and compute $\mathbf{y} = \mathsf{CS}_{q,|\mathbf{y}'|}(\mathbf{y}')$, where $q = r \mod n$.

5. Compute
$$C_1 = (hy + \bar{r})r + z, \quad C_2 = \mathsf{F}(ek, \mathbf{r}),$$

where $\bar{r}$ is the decimal value of the complement of $\mathbf{r}$ and $z = \sum_{i=1}^{l-1} u_i$.

Decryption. On inputs (C, td), Dec perform as follows:

1. Compute coins $\mathbf{r}$ as $\mathbf{r} = \mathsf{F}^{-1}(C_2, td)$. Compute $\bar{r}$ and $h = \mathsf{wt}(r)$. If $h \geq k - h$ then $l \leftarrow h$, else $l \leftarrow k - h$.

2. Compute $U_s = (u_1, \ldots, u_{l-1}, 0)$ and $z = \sum_{i=1}^{l-1} u_i$, where $u_i = (r+i) \mod (l-i)$ and $s = \sum_{i=1}^{l-1} u_i(l-i)!$.

3. Compute
$$y = \frac{(C_1 - z)/r - \bar{r}}{h},$$
and reject the ciphertext if y is not a $(l\lceil n/l \rceil)$-bit integer.

4. Compute $\mathbf{y}' = \mathsf{CS}^{-1}_{q,|\mathbf{y}|}(\mathbf{y})$. $q = r \mod n$ and $|\mathbf{y}'| = |\mathbf{y}|$.

5. Compute $\tilde{\mathbf{m}} = \mathsf{PCA}^{-1}(\mathbf{y}', \mathbf{r}, s)$.

6. Compute $\mathbf{m}' = \tilde{\mathbf{m}} \oplus G(\mathbf{r})$ and $\mathsf{RBS} = \mathsf{Msb}_{l\lceil n/l\rceil - n}(\mathbf{r})$. Check wether

$$\mathsf{Lsb}_{l\lceil n/l\rceil - n}(\mathbf{m}') \stackrel{?}{=} \mathsf{RBS} \tag{6}$$

holds and reject if not. If it holds output

$$\mathbf{m} = \mathsf{Msb}_{l\lceil n/l\rceil - n}(\mathbf{m}'), \tag{7}$$

else output $\perp$.

Theorem 2. Let TDF be a one-way trapdoor function, then the $\mathsf{PKE}[\mathsf{TDF}]$ defined above is CCA2-secure. The proof of Theorem2 is similar to the proof of Theorem1 which is omitted.

References

[1] T. Berger, P. Cayrel, P. Gaborit and A. Otmani. Reducing key length of the mceliece cryptosystem. In *AFRICACRYPT 2009*, LNCS, Vol. 5580. pp.77-97, 2009.

[2] D. Bernstein, T. Lange and C. Peters. Attacking and defending the mceliece cryptosystem. In *PQCrypto 2008*, LNCS, Vol.5299. pp.31-46, 2008.

[3] D. Bernstein, T. Lange, C. Peters and H. van Tilborg. Explicit bounds for generic decoding algorithms for code-based cryptography. In *WCC 2009*, pp.168-180, 2009.

[4] P. L. Cayrel, G. Hoffmann, E. Persichetti. Efficient Implementation of a CCA2-Secure Variant of McEliece Using Generalized Srivastava Codes. In *PKC 2012*, LNCS, Vol. 7293, pp 138-155, 2012.

[5] R. Dowsley, J. Müller-Quade, A. C. A. Nascimento. A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model. In *CT-RSA 2009*, LNCS, Vol. 5473, pp. 240251.

[6] N. Döttling, R. Dowsley, J. M. Quade and A. C. A. Nascimento. A CCA2 Secure Variant of the McEliece Cryptosystem. *IEEE, Transactions on Information Theory*, Vol. 58(10), pp.6672-6680, 2012.

[7] J.-C. Faugère, A. Otmani, L. Perret, J.-P. Tillich. Algebraic Cryptanalysis of McEliece Variants with Compact Keys. In *EUROCRYPT 2010*, pp. 279-298, 2010.

[8] J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, J.-P. Tillich. A Distinguisher for High Rate McEliece Cryptosystems. *IEEE Information Theory Workshop (ITW)*, pp. 282286, 2011.

[9] M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In ASIACRYPT 2009, LNCS, Vol.5912, pp. 88-105, 2009.

[10] D.-M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, G. Segev, More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In *PKC 2010*, LNCS, Vol.6056, pp.279295, 2010.

[11] E. Kiltz, P. Mohassel, and A. O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *EUROCRYPT 2010*, LNCS, Vol. 6110 pp. 673692, 2010.

[12] K. Kobara and H. Imai. Semantically Secure McEliece Public-Key Cryptosystems Conversions for McEliece PKC. In *PKC 2001*, LNCS, Vol.1992, pp. 19-35, 2001.

[13] R. Lu, X. Lin, X. Liang and X. Shen. An efficient and provably secure public key encryption scheme based on coding theory. In *Security Comm. Networks*,Vol.4 (19), pp. 1440-1447, 2011.

[14] R. McEliece. A public-key cryptosystem based on algebraic number theory. *Technical report, Jet Propulsion Laboratory*. DSN Progress Report pp. 42-44, 1978.

[15] R. Misoczki and P. Barreto. Compact mceliece keys from goppa codes. In *SAC'2009*, LNCS, Vol.5867. pp.376-392, 2009.

[16] S. Myers and A. Shelat. Bit encryption is complete. In *FOCS 2009*, pp. 607616. IEEE Computer Society Press, 2009.

[17] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Probl. Control and Inform. Theory*, Vol.15, pp.1934, 1986.

[18] R. Nojima, H. Imai, K. Kobara and K. Morozov. Semantic Security for the McEliece Cryptosystem without Random Oracles. *Designs, Codes and Cryptography*, Vol. 49, No. 1-3, pp. 289-305, 2008.

[19] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC 2008*, pp. 187-196, 2008.

[20] E. Persichetti. On a CCA2-secure variant of McEliece in the standard model. Cryptology ePrint Archive: Report 2012/268. http://eprint.iacr.org/2012/268.pdf

[21] K. Preetha Mathew, S. Vasant, S. Venkatesan and C. Pandu Rangan. An Efficient IND-CCA2 Secure Variant of the Niederreiter Encryption Scheme in the Standard Model. In *ACISP 2012*, LNCS, Vol.7372, pp. 166179, 2012.

[22] C. Rackoff and D. Simon. Noninteractive Zero-knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *CRYPTO 91*, LNCS, Vol. 576, pp. 433-444, 1992.

[23] A. Rosen and G. Segev. Chosen-Ciphertext Security via Correlated Products. In *TCC 2009*, LNCS, Vol. 5444, pp. 419-436, 2009.

[24] N. Sendrier. The tightness of security reductions in code-based cryptography. In *IEEE, Information Theory Workshop (ITW)*, pp.415-419, 2011.