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

Roohallah Rastaghi

July, 5, 2013

Abstract. Recently, a few chosen-ciphertext secure (CCA2-secure) variants of the McEliece cryptosystem 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 cryptosystems. 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 cryptosystem in the standard model. The main novelty is that, unlike previous approaches, our approach is a generic conversion and can be applied to *any* code-based one-way trapdoor cryptosystem. To further demonstrate the usefulness of our approach, we introduce a direct black-box construction of CCA2-secure scheme from one-way trapdoor functions (OW-TDFs) in the standard model, the lowest-level security notion in the context of public-key cryptography, resolving a problem that has remained unsolved in the past two decades.

Keywords: Post-quantum cryptography, McEliece cryptosystem, Permutation combination 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 [24], 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 [14] and Niederreiter [17] 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 [1,2,3,9,15].

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 [22]. It is possible to produce CCA2-secure variants of the code-based cryptosystems in the random oracle model [4,12,13], 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 [23] to introduce a correlation-secure trapdoor function related to the hardness of syndrome decoding. Their construction is based on the Niederreiter cryptosystem. Very recently, Preetha Mathew et al. [21] proposed an efficient variant of the Niederreiter scheme based on lossy trapdoor functions [19], 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 cryptosystem. Their scheme 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 [20]. Very recently, Döttling et al. [6] showed that Nojima et al. [18] 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 [20] proposed a CCA2-secure cryptosystem based on the McEliece assumptions using the original Rosen-Segev approach.

1.2 Motivation

To the date, as we state 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 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 such 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 [21] 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. In our scheme, the decryption algorithm is witness-recovering, i.e., along with the message it also recovers the randomness that was used to create the ciphertext. It then checks well-formedness simply by re-encrypting the message under the retrieved randomness, and comparing the result to the original ciphertext.

1.3 Our Contributions

To tackle the challenging above issues, we introduce a randomized variant of the McEliece cryptosystem and prove its security in the standard model based on the McEliece assumptions. Our contributions in this paper are:

- The main novelty is that our construction is a generic conversion and can be applied to *any* one-way trapdoor functions, resolving a problem 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 *arbitrary-length* long messages without employing the hybrid encryption method and symmetric encryption (i.e., KEM/DEM paradigm).

Organisation. In the next section, we briefly explain some mathematical background and definitions. Then, in Section 3, we introduce our proposed scheme. Security and performance of the proposed scheme will be discussed in section 4. We introduce a generalized scheme based on OW-TDF in Section **??**.

2 Preliminary

2.1 Notation

We represent a binary string in general by bold face letter such as $\mathbf{x} = (x_1, \ldots x_n)$. 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 (**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 $\mathsf{CS}_{q,n}(s_i) = s_{(i+q \mod n)}$, $1 \leq i \leq n$.*

Definition 2 (**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 3 (**One-wayness**). *Let $\mathcal{A}$ be an inverter and define its* OW-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 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 $\mathsf{wt}(\mathbf{e}) \leq t$.*

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 $t \leq \lfloor \frac{d-1}{2} \rfloor$. 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.*

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

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 $\mathbf{Succ}^{\mathrm{GDP}}_{\mathcal{A}} \geq \varepsilon$.

Definition 6 (**Public-key encryption**). *A public-key encryption scheme (PKE) 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 PKE 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 McEliece cryptosystem

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

$$pk = (\mathsf{G}^{\text{pub}}, t) \quad \text{and} \quad pk = (\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$$

where $D_{\mathcal{G}}$ is an efficient decoding algorithm for $\mathcal{G}$.

– **Encryption.** $\mathsf{Enc}_{\text{McE}}(pk)$ 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 $\mathbf{c}$ as follows.

$$\mathbf{c} = \mathbf{m}\mathsf{G}^{\text{pub}} \oplus \mathbf{e}.$$

– **Decryption.** To decrypt a ciphertext $\mathbf{c}$, $\mathsf{Dec}_{\text{McE}}(sk, \mathbf{c})$ 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 is based on a heuristic randomized encoding algorithm which can be applied to *any* code-based one-way trapdoor cryptosystem such as the McEliece, the Niederreiter and etc. Encoding includes a permutation and combination on the message bits and performs with an algorithm called *permutation combination algorithm* (PCA).

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

To encrypt a message with arbitrary long length n, in security parameter k, we firstly choose coins $r \in \{0,1\}^k, r \neq 0,1^k$ uniformly at random and use it to encrypt the message. The proposed encoding algorithm uses coins r as an *ephemeral* key to encrypt the message each time it wants to encode a message. Therefore, encoding algorithm can be regarded as a one-time pad encryption scheme, and so, providing high security level. Coins r encrypts using underlying code-based PKE scheme and sent to the receiver along with the encrypted message. Receiver after receives the ciphertext, firstly recovers coins r from the underlying code-based PKE scheme and then uses it to decrypt the message.

3.1 Permutation combination algorithm

To encode message $\mathbf{m} = (m_1, \ldots, m_n)$, we firstly pick coins $\mathbf{r}$ from $\{0,1\}^k, \mathbf{r} \neq 0,1^k$ uniformly at random. Let $\mathsf{wt}(\mathbf{r}) = h$ be its Hamming weight. If $h \geq k - h$, then $l \leftarrow h$, else $l \leftarrow k - h$. We can divide $\mathbf{m}$ into l blocks $\mathbf{m} = (b_1 \| \ldots \| b_l)$ with equal binary length $\lceil n/l \rceil$. If $\mathbf{m}$ is short, then we have to pad it. In these cases, we 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 $b_l = \mathsf{Lsb}_v(m)$, else, $v = \lceil n/l \rceil$ and RBS is a random string with length $l\lceil n/l \rceil - n$ which sampled from $\mathbf{r}$. 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

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

The sequence $U_s = (u_1, \ldots, u_l)$ is called *factorial carry value* of s. We define original sequence $\mathbf{m}^0$ as $\mathbf{m}^0 = (b_1 \| b_2 \| \ldots \| b_l)$. Recombine all the 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 1 (**Permutation Combination Algorithm** (PCA))**.**

Input: Message $\mathbf{m} = (m_1, \ldots, m_n)$, coins $\mathbf{r} \in \{0,1\}^k$ and integer $s, 1 \leq s \leq l!-1$.
Output: Encoded message $\mathbf{y}' = \mathbf{m}^s = (b'_1 \| \ldots \| b'_l)$.

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, $\mathsf{RBS} \leftarrow \mathsf{MSb}_{(l\lceil n/l \rceil - n)}(\mathbf{r})$.

[2] Note 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 $\mathbf{r}$.

3. $\mathbf{m}' \leftarrow \mathbf{m} \| \mathsf{RBS}$ and divide $\mathbf{m}'$ into l blocks $(b_1 \| \ldots \| b_l)$.

PERMUTATION:

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

Note that the *number* and the *length* of the message blocks are variable and change by $\mathbf{r}$.

We illustrate PCA encoding algorithm with a small example. Suppose $\mathbf{m} = (m_1, \ldots, m_{512})$ and $\mathbf{r} = (r_1, \ldots, r_{25})$ 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 $\mathbf{m}' = (\underbrace{m_1, \ldots, m_{40}}_{b_1} \| \underbrace{m_{41}, \ldots, m_{80}}_{b_2} \| \ldots \| \underbrace{m_{481}, \ldots, m_{512} \| r_1, \ldots r_8}_{b_{13}})$.
We choose integer $s, 1 \le s \le 13! - 1$, 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 encryption scheme $\Pi = (\mathsf{Gen}_{\mathrm{McE}}, \mathsf{Enc}_{\mathrm{McE}}, \mathsf{Dec}_{\mathrm{McE}})$, we transform it into CCA2-secure scheme $\Pi' = (\mathsf{Gen}_{cca2}, \mathsf{Enc}_{cca2}, \mathsf{Dec}_{cca2})$. In our construction, the decryption algorithm is randomness-recovering, i.e., along with the message it recovers the randomness that was used to create the ciphertext. It then checks well-formedness by re-encrypting the message under the retrieved randomness, and comparing the result to the original ciphertext.

Key Generation. Suppose $\mathsf{Gen}_{\mathrm{McE}}$ be the McEliece system generator. On security parameter k, the generator Gen_{cca2} runs $\mathsf{Gen}_{\mathrm{McE}}(1^k)$ to obtain $sk = sk_{\mathrm{McE}} = (\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$ and $pk = pk_{\mathrm{McE}} = (\mathsf{G}^{pub}, t)$ as in subsection 2.3.

Encryption. Because of the block-based structure of the PCA encoding algorithm, it cannot perfectly hide the message bits. Therefore, we firstly disguise message $\mathbf{m}$ and conceal its bits by computing $\tilde{m} = m + r$ and then encode the disguised message $\tilde{m}$, where m, r are the corresponding decimal value of $\mathbf{m}$ and $\mathbf{r}$.

To create ciphertext, encryption algorithm in some cases performs operations in decimal, and in some 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 values by bold face fonts.
To encrypt message $\mathbf{m} \in \{0,1\}^n$, $\mathsf{Enc}(pk)$:

1. Choose coins $\mathbf{r} \in \{0,1\}^k, \mathbf{r} \neq 0, 1^k$ uniformly at random. Let $\mathsf{wt}(\mathbf{r}) = h$ and r be its corresponding decimal value. If $2h > k$ then $l \leftarrow h$, else $l \leftarrow k - h$.
2. $\tilde{m} \leftarrow m + r$.
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 algorithm (algorithm 1) on inputs $(\tilde{\mathbf{m}}, \mathbf{r}, s)$ to generate encoded message $\mathbf{y}' = \mathsf{PCA}(\tilde{\mathbf{m}}, \mathbf{r}, s)$. In this case we have $U_s = (u_1, \ldots, u_{l-1}, 0)$.
4. Perform a circular q-shift on the encoded message $\mathbf{y}'$ and compute $\mathbf{y} = \mathsf{CS}_{q,|\mathbf{y}'|}(\mathbf{y}')$, where $q = r \mod n$. Note n and so $q < |\mathbf{y}'|$.
5. $\mathsf{Enc}_{\mathrm{McE}}$ randomly choose a vector $\mathbf{e}$ wit Hamming weight $\mathsf{wt}(\mathbf{e}) = t$.
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 $\bar{r}$ is the decimal value of the complement of $\mathbf{r}$ and $z = \sum_{i=1}^{l-1} u_i$.

To handle CCA2 security related issues, we increase obfuscation of the encoded message by setting $C_1 = (hy + \bar{r})r + z$ in order to decrease malleability of

the ciphertext. We correlate the message bits m_i to the coins $\mathbf{r}$ via encoding algorithm. We also correlate encoded message $\mathbf{y}'$ to the coins $\mathbf{r}$ by perform a secure circular shift, whose shift step depends on the decimal value of coins $\mathbf{r}$, i.e. r. Therefore, the CCA2 adversary to extract message $\mathbf{m}$ from C_1 must first recover the *same* coins $\mathbf{r}$ that was used to create the ciphertext from the code-based PKE scheme, which is impossible, if the underlying code-based PKE scheme be secure.

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

1. Compute coins $\mathbf{r}$ as $\mathbf{r} = \mathsf{Dec}_{\mathrm{McE}}(C_2, sk)$ and retrieve error vector $\mathbf{e} = C_2 \oplus \mathbf{r}\mathsf{G}^{pub}$
2. Check wether
$$C_2 \stackrel{?}{=} \mathsf{Enc}_{\mathrm{McE}}(pk, \mathbf{r}; \mathbf{e}) \tag{2}$$
holds[3] and reject the ciphertext if not (*consistency* checking). If it holds compute $\hat{r}$ and $h = \mathsf{wt}(r)$. If $h \geq k - h$ then $l \leftarrow h$, else $l \leftarrow k - h$.
3. 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 $s = \sum_{i=1}^{l-1} u_i (l - i)!$.
4. Compute
$$y = \frac{(C_1 - z)/r - \bar{r}}{h},$$
and reject the ciphertext if y is not an integer (consistency checking).
5. Compute $\mathbf{y}' = \mathsf{CS}^{-1}_{q,|\mathbf{y}|}(\mathbf{y})$, where $\mathbf{y}$ is the binary representation of y. Note $q = r \mod n$ and $|\mathbf{y}'| = |\mathbf{y}|$.
6. Compute $\tilde{\mathbf{m}} = \mathsf{PCA}^{-1}(\mathbf{y}', \mathbf{r}, s)$.
7. Run algorithm 1 on inputs $(\tilde{\mathbf{m}}, \mathbf{r}, s)$ and compute $\hat{\mathbf{y}}'$, and $\hat{\mathbf{y}} = \mathsf{CS}_{q,|\hat{\mathbf{y}}'|}(\hat{\mathbf{y}}')$.
8. Check whether
$$C_1 \stackrel{?}{=} (h\hat{y} + \bar{r})r + z. \tag{3}$$
holds (consistency checking). If it holds output $m = \tilde{m} - r$, else output $\perp$.

[3] In deterministic code-based PKE schemes such as Niederreiter cryptosystem, we don't need to perform re-encryption checking. In such schemes since encryption algorithm is deterministic, each message has *one* pre-image. Therefore, if $C_2 \neq C_2^*$ then $\mathbf{r} = \mathsf{Dec}(C_2, pk) \neq \mathsf{Dec}(C_2^*, pk) = \mathbf{r}^*$. But in the McEliece encryption scheme, for 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 perform re-encryption checking to check well-formedness of the ciphertext and reject such maliciously-formed ciphertexts.

4 Security and performance analysis

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

In the proof of security, we exploit the fact that for a ciphertext, we can recover the message if we know the *same* randomness r that was used to create the ciphertext. Our proof is constructed as Games. The first game, Game 0, is basically identical to the CCA2 game, and the adversary's advantage is defined accordingly. In each game, the adversary will still has to guess a given bit (as in the CCA2 game), but in each game the simulator will reject inconsistent ciphertexts by well-formedness checking of the ciphertext. Therefore, his advantage in guessing challenge bit in each game is 0. Let S_i be the event that the adversary $\mathcal{A}$ wins in Game i. Here is the games.

Game 0. We define Game 0 which is an interactive computation between an *adversary* $\mathcal{A}$ and a *challenger* $\mathcal{D}$. This game is usual CCA2 game used to define CCA2-security, in which the challenger provides the adversary's environment.
Initially, the challenger runs the key generation algorithm and gives the public key pk to the adversary.
Challenge Ciphertext. The adversary chooses two messages m_0, m_1 with $|m_0| = |m_1|$ and gives it to the challenger. The challenger chooses $b \in \{0,1\}$, error vector e^* and coins $r^* \in \{0,1\}^k, r \neq 0, 1^k$ at random and encrypts m_b, obtaining the challenge ciphertext $C^* = (C_1^*, C_2^*) = ((h^* y^* + \overline{r^*})r^* + z^*, \mathsf{Enc}_{\mathrm{MCE}}(pk, r^*; e^*))$ and gives it to the adversary, where

$$\mathbf{y}^* = \mathsf{CS}_{q^*,|\mathbf{y}'^*|}(\mathbf{y}'^*) \quad \text{and} \quad \mathbf{y}'^* = \mathsf{PCA}(\tilde{m^*}, \mathbf{r}^*, s^*), \tag{4}$$

where $\tilde{m^*} = m_b + r^*$ and $h^* = \mathsf{wt}(\mathbf{r}^*)$. We denote by $\mathbf{r}^*$ and $\mathbf{e}^*$ the corresponding intermediate quantities chosen by the challenger, and l^*, $u_i^* = (r^* + i) \mod (l^* - i), 1 \leq i \leq l^* - 1$, $z^* = \sum_{i=1}^{l^*-1} u_i^*$, $q^* = r^* \mod n$ and $s^* = \sum_{i=1}^{l^*-1} u_i^*(l^* - i)!$ the corresponding intermediate quantities computed by the challenger.

Decryption Oracle. The challenger has to simulate the decryption oracle. In processing a decryption request $C = (C_1, C_2)$, the simulator proceeds as follows:

Input: The ciphertext $C = (C_1, C_2)$
Output: The message m.

D1: $\mathbf{r} \leftarrow \mathsf{Dec}_{\mathrm{McE}}(C_2, sk)$ and $\mathbf{e} = C_2 \oplus \mathbf{r}\mathsf{G}^{pub}$. If $C_2 \neq \mathsf{Enc}_{\mathrm{McE}}(pk, \mathbf{r}; \mathbf{e})$, return $\perp$. \\Ciphertext inconsistent

D2: $h \leftarrow \mathsf{wt}(\mathbf{r})$. If $2h \geq k$ $l \leftarrow h$, else $l \leftarrow k - h$.

D3: Compute $\bar{r}$, $q \leftarrow r \mod n$, $u_i \leftarrow (r + i) \mod (l - i)$ for $1 \leq i \leq l - 1$, $z \leftarrow \sum_{i=1}^{l-1} u_i$ and $s \leftarrow \sum_{i=1}^{l-1} u_i(l - i)!$.

D4: $y \leftarrow ((C_1 - z)/r - \bar{r})/h$. If y is not a positive integers, return $\perp$. \\Ciphertext inconsistent

D5: $\mathbf{y}' \leftarrow \mathsf{CS}^{-1}_{q,|\mathbf{y}|}(\mathbf{y})$ and $\tilde{\mathbf{m}} \leftarrow \mathsf{PCA}^{-1}(\mathbf{y}', \mathbf{r}, s)$.

D6: Run algorithm 1 on inputs $(\tilde{\mathbf{m}}, \mathbf{r}, s)$ and output $\hat{\mathbf{y}'} = \mathsf{PCA}(\tilde{\mathbf{m}}, \mathbf{r}, s)$, and compute $\hat{\mathbf{y}} = \mathsf{CS}_{q,|\hat{\mathbf{y}'}|}(\hat{\mathbf{y}'})$. \\Re-encoding checking

D7: If $C_1 \neq (h\hat{y} + \bar{r})r + z$, return $\perp$. \\Ciphertext inconsistent

Return $m \leftarrow \tilde{m} - r$.

The only restriction on the adversary's requests is that after it makes a challenge request, the subsequent decryption requests must not be the same as the challenge ciphertext. At the end of the game, the adversary $\mathcal{A}$ outputs $\tilde{b} \in \{0, 1\}$. Let S_0 be the event that $\tilde{b} = b$. Since Game 0 is identical to the CCA2 game we have that

$$|\Pr[S_0]| = \frac{1}{2} + \mathsf{Adv}^{\mathsf{Ind-cca2}}_{\Pi', \mathcal{A}}(k),$$

and, our goal is to prove $|\Pr[S_0]| \leq 1/2$.

Since a decryption query on the challenge ciphertext is forbidden by the CCA2-experiment, so if $C_2 = C_2^*$, then $C_1 \neq C_1^*$ and vice versa. Therefore, it is sufficient that we consider two below games.

Game 1. Game 1 is the same as Game 0, except that we assume $C_2 = C_2^*$. (In this case $C_1 \neq C_1^*$).

Game 2. Game 2 is the same as Game 0, except that we assume $C_1 = C_1^*$. (In this case $C_2 \neq C_2^*$).

Claim 1. *We claim that*

$$|\Pr[S_1]| = |\Pr[S_0]|. \tag{5}$$

Proof: In Game 1, the CCA2 adversary $\mathcal{A}$ queries on input $(C_1 \neq C_1^*, C_2^*)$. The challenger takes as input (C_1, C_2^*) and computes $r = r^*$, $u_i = u_i^*, 1 \leq i \leq l-1$, $s = s^*$ and $z = z^*$ in **D1, D2, D3**. Then it computes $y = ((C_1 - z^*)/r^* - \overline{r^*})/h^* \neq ((C_1^* - z^*)/r^* - \overline{r^*})/h^* = y^*$ in D4. If y is not a positive integer, then the simulator rejects the ciphertext in **D4**. Since $y \neq y^*$ (and so $\mathbf{y} \neq \mathbf{y}^*$), thus $\mathbf{y}' = \mathsf{CS}^{-1}_{q^*,|\mathbf{y}|}(\mathbf{y}) \neq \mathsf{CS}^{-1}_{q^*,|\mathbf{y}^*|}(\mathbf{y}^*) = \mathbf{y}'^*$ and so $\tilde{\mathbf{m}} \neq \tilde{\mathbf{m}}^*$ in **D5**. (Note in this case $q = r \mod n = q^*$). Since $\tilde{\mathbf{m}} \neq \tilde{\mathbf{m}}^*$, thus $\hat{\mathbf{y}'} = \mathsf{PCA}(\tilde{\mathbf{m}}, \mathbf{r}^*, s^*) \neq \mathsf{PCA}(\tilde{\mathbf{m}}^*, \mathbf{r}^*, s^*) = \mathbf{y}'^*$ and so $\hat{\mathbf{y}} \neq \mathbf{y}^*$ in **D6**. Therefore, $C_1 = (h^*\hat{y} + \overline{r^*})r^* + z^* \neq (h^*y^* + \overline{r^*})r^* + z^* = C_1^*$ and the challenger rejects the ciphertext in **D7**. Thus, the $\mathcal{A}$'s advantage in this game is 0 and we have $|\Pr[S_1]| = |\Pr[S_0]|$.

Claim 2. *We claim that:*

$$|\Pr[S_2]| = |\Pr[S_0]|. \tag{6}$$

Proof: Let $\mathcal{A}$ be an adversary against CCA2 security of the proposed scheme in Game 2. In this game, the adversary $\mathcal{A}$ queries on ciphertexts of the form $C = (C_1^*, C_2 \neq C_2^*)$. We should consider two cases:

Case1: $\mathbf{r} = \mathbf{r}^*$ while $C_2 \neq C_2^*$. Since $C_2 \neq C_2^*$, thus $\mathbf{e} = C_2 \oplus \mathbf{r}^* \mathsf{G}^{pub} \neq C_2^* \oplus \mathbf{r}^* \mathsf{G}^{pub} = \mathbf{e}^*$, and so the simulator will reject the ciphertext by re-encryption checking in **D1**.

Case2: $\mathbf{r} \neq \mathbf{r}^*$. In the worst case, we assume $h = \mathsf{wt}(\mathbf{r}) = \mathsf{wt}(\mathbf{r}^*) = h^*$, and so $l = l^*$. In these cases, the simulator computes $\overline{r} \neq \overline{r^*}$, $u_i \neq u_i^*, 1 \leq i \leq l-1$, $s \neq s^*$ and $z \neq z^*$ in **D3** and $y = ((C_1^* - z)/r - \hat{r})/h^* \neq ((C_1^* - z^*)/r^* - \overline{r^*})h^* = y^*$ in **D4**. If y is not an integer, then the simulator rejects the ciphertext. In these cases, since $s \neq s^*$, $q = r \mod n \neq r^* \mod n = q^*$ and $y \neq y^*$ (and so $\mathbf{y} \neq \mathbf{y}^*$), thus $\mathbf{y}' \neq \mathbf{y}'^*$ and $\tilde{\mathbf{m}} \neq \tilde{\mathbf{m}^*}$ in **D5**. Since $\tilde{\mathbf{m}} \neq \tilde{\mathbf{m}^*}$, $s \neq s^*$ and $\mathbf{r} \neq \mathbf{r}^*$, thus $\hat{\mathbf{y}'} = \mathsf{PCA}(\tilde{\mathbf{m}}, \mathbf{r}, s) \neq \mathsf{PCA}(\tilde{\mathbf{m}^*}, \mathbf{r}^*, s^*) = \mathbf{y}'^*$. So, $\hat{\mathbf{y}} \neq \mathbf{y}^*$ and $\hat{y} \neq y^*$) in **D6**. Therefore, $C_1 = (h^*\hat{y} + \hat{r})r + z \neq (h^* y^* + \overline{r^*})r^* + z^* = C_1^*$ and the challenger rejects the ciphertext in **D7**. Therefore, the $\mathcal{A}$'s advantage in this game is 0 and we have $|\Pr[S_2] = \Pr[S_0]|$.

Claim 3. *We claim that*

$$|\Pr[S_1]| = 1/2. \tag{7}$$

Proof: In Game 1, the adversary queries on the ciphertext of the form (C_1, C_2^*). We consider two conceivable scenarios:

Scenario 1. The adversary modifies C_1 based on C_1^* and then queries on it. In this case since $C_1 \neq C_1^*$, as we proved in claim 1, the simulator will output $\perp$.

Scenario 2. The adversary uniformly chooses C_1 at random and then queries on it. In this case, since the component C_1 of the queried ciphertext (C_1, C_2^*) is not computed by equation (4) but rather chosen uniformly at random, so it is statistically independent from the challenge bit b. Thus, the $\mathcal{A}$'s advantage in this case is obviously 0, however, as we state in claim 1, the simulator will reject ciphertexts of the form (C_1, C_2^*). Therefore we have

$$|\Pr[S_1]| = \frac{1}{2}$$

Similarly, since the advantage of the CCA2 adversary in Game 2 is 0, therefore we have $|\Pr[S_2]| = 1/2$.

Completing the Proof:

From equations 5,6 and 7 we have:

$$|\Pr[S_0]| = |\Pr[S_1]| = |\Pr[S_2]| = \frac{1}{2},$$

4.1 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 time for encoding and decoding is negligible compared to the time for computing $\mathsf{Enc}_{\mathrm{McE}}$ and $\mathsf{Dec}_{\mathrm{McE}}$. The public/secret keys are as in the original scheme. Encryption roughly needs one application of $\mathsf{Enc}_{\mathrm{McE}}$, and decryption roughly needs one application of $\mathsf{Dec}_{\mathrm{McE}}$ together one application of $\mathsf{Enc}_{\mathrm{McE}}$.

As we previously stated, the Niederreiter-based proposed scheme does not need re-encryption checking. Therefore, compared to Freeman et al.[10] and Mathew et al.[21] schemes, our scheme is more efficient.

The comparison of the proposed schemes with existing schemes are presented in table 2.

Table 2. 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]	$2k \times pk_{\mathrm{McE}}$	$2k \times sk_{\mathrm{McE}}$	$k \times \mathsf{Ciph}_{\mathrm{McE}}$	$k \times \mathsf{Enc}_{\mathrm{McE}}+$ $1\,\mathsf{Ver}_{\mathcal{OT}-\mathcal{SS}}+$ $1\,\mathcal{OT}-\mathcal{SS}$	$1 \times \mathsf{Dec}_{\mathrm{McE}}+$ $t \times \mathsf{Enc}_{\mathrm{McE}}$
Freeman et al.[10]	$2k \times pk_{\mathrm{Nie}}$	$2k \times sk_{\mathrm{Nie}}$	$k \times \mathsf{Ciph}_{\mathrm{Nie}}$	$k \times \mathsf{Enc}_{\mathrm{Nie}}+$ $1\,\mathcal{OT}-\mathcal{SS}$	$1\,\mathsf{Ver}_{\mathcal{OT}-\mathcal{SS}}+$ $1 \times \mathsf{Dec}_{\mathrm{Nie}}+$ $t \times \mathsf{Enc}_{\mathrm{Nie}}$
Mathew et al.[21]	$1\,pk_{\mathrm{Nie}}+$ $1\,(n \times n)$ Matrix	$2 \times sk_{\mathrm{Nie}}$	$2 \times \mathsf{Ciph}_{\mathrm{Nie}}$	$2 \times \mathsf{Enc}_{\mathrm{Nie}}+$ 1 MM$+$ $1\,\mathcal{OT}-\mathcal{SS}$	$1\,\mathsf{Ver}_{\mathcal{OT}-\mathcal{SS}}+$ $1 \times \mathsf{Dec}_{\mathrm{Nie}}+$ $2 \times \mathsf{Enc}_{\mathrm{Nie}}+$ 1 MM
Proposed Scheme	$1\,pk_{\mathrm{McE}}$	$1\,sk_{\mathrm{McE}}$	$1\,\mathsf{Ciph}_{\mathrm{McE}}$ $+l\lceil n/l \rceil$	$1\,\mathsf{Enc}_{\mathrm{McE}}+$ PCA $+$ 1P	$1\,\mathsf{Dec}_{\mathrm{McE}}$ $+1\,\mathsf{Enc}_{\mathrm{McE}}$ $+1\,\mathsf{PCA}^{-1}$

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 (**??**), PCA^{-1}: Reverse Permutation Combination Algorithm and $t \leq k$.

5 General construction from TDF

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 [12]. Kiltz et al. proposed a black-box *one*-bit CCA2-secure encryption scheme and then apply a transform of Myers and shelat [18] from one-bit to many-bit CCA-secure encryption scheme. The Myers-shelat conversion is not efficient; it uses encryption reputation paradigm and 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 use in practice.

Here, we give a direct black-box construction of CCA2-Secure PKE scheme from TDFs. Our construction is similar to the construction of section 4. We only need to replace the 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 $\mathsf{PKE[TDF]} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

Key Generation. On security parameter k, the generator Gen runs Tdg to obtain $(ek, pk) \leftarrow \mathsf{Tdg}(1^k)$ and return (ek, pk).

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. $\tilde{m} \leftarrow m + r$.
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 algorithm (algorithm 1) on inputs $(\tilde{\mathbf{m}}, \mathbf{r}, s)$ to generate encoded message $\mathbf{y}' = \mathsf{PCA}(\tilde{\mathbf{m}}, \mathbf{r}, s)$. In this case we have $U_s = (u_1, \ldots, u_{l-1}, 0)$.
4. Perform a circular q-shift on the encoded message $\mathbf{y}'$ and compute $\mathbf{y} = \mathsf{CS}_{q,|\mathbf{y}'|}(\mathbf{y}')$, where $q = r \mod n$. Note n and so $q < |\mathbf{y}'|$.
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 td, C, 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 an integer (consistency checking).
4. Compute $\mathbf{y}' = \mathsf{CS}^{-1}_{q,|\mathbf{y}|}(\mathbf{y})$. Note $q = r \mod n$ and $|\mathbf{y}'| = |\mathbf{y}|$.
5. Compute $\tilde{\mathbf{m}} = \mathsf{PCA}^{-1}(\mathbf{y}', \mathbf{r}, s)$.
6. Run algorithm 1 on inputs $(\tilde{\mathbf{m}}, \mathbf{r}, s)$ and compute $\hat{\mathbf{y}}'$, and $\hat{\mathbf{y}} = \mathsf{CS}_{q,|\hat{\mathbf{y}}'|}(\hat{\mathbf{y}}')$.
7. Check whether
$$C_1 \stackrel{?}{=} (h\hat{y} + \bar{r})r + z. \tag{8}$$
holds (consistency checking). If it holds output $m = \tilde{m} - r$, else output $\perp$.

Theorem 2. *If* TDF *is one-way, then the* PKE[TDF] *defined above is CCA2-secure.*

Sketch of proof. Since a decryption query on the challenge ciphertext is forbidden by the CCA2-experiment, therefore it is sufficient that we consider two following cases.

Case 1. $C_1 \neq C_1^*$ and $C_2 = C_2^*$. In this case we have $\mathbf{r} = \mathsf{F}^{-1}(C_2, td) = \mathsf{F}^{-1}(C_2^*, td) = \mathbf{r}^*$, and so $\bar{r} = \overline{r^*}$, $s = s^*$, $z = z^*$. But since $C_1 \neq C_1^*$, $y = ((C_1 - z^*)/r^* - \overline{r^*})/h^* \neq ((C_1^* - z^*)/r^* - \overline{r^*})/h^* = y^*$ (and so $\mathbf{y} \neq \mathbf{y}^*$). Therefore, $\mathbf{y}' \neq \mathbf{y}'^*$ and $\tilde{m} \neq \tilde{m^*} = m_b + r^*$. So, such ciphertexts will be reject by re-encoding checking in 8.

Case 2. $C_2 \neq C_2^*$ and $C_1 = C_1^*$. In this case we have $\mathbf{r} = \mathsf{F}^{-1}(C_2, td) \neq \mathsf{F}^{-1}(C_2^*, td) = \mathbf{r}^*$, and so $\bar{r} \neq \overline{r^*}$, $s \neq s^*$, $z \neq z^*$. Therefore $y = ((C_1^* - z)/r - \bar{r})/h \neq ((C_1^* - z^*)/r^* - \overline{r^*})/h^* = y^*$ (and so $\mathbf{y} \neq \mathbf{y}^*$). Therefore $\mathbf{y}' \neq \mathbf{y}'^*$ and $\tilde{m} \neq \tilde{m^*}$. So, the simulator will reject such ciphertexts by re-encoding checking in 8.

Remark 1. *Note that the CCA2 adversary chooses ciphertext $C \neq C^*$ and submit it to the decryption oracle, and receives its decryption, i.e. message $\boldsymbol{m}$. The decryption oracle dose not output the decryption of C_2, i.e. randomness $\boldsymbol{r}$. Thus, the CCA2 adversary does not seen and does not access to the decryption of C_2. Therefore, the proposed scheme does not need the underlying* PKE[TDF] *satisfies non-malleability related issues. The decryption algorithm uses the decryption of C_2 to decrypt C_1, and rejects the ciphertext if it is not* same *as the randomness that was used by encryption algorithm to encrypt the message. Since* F *is injective, therefore each message has* one *pre-image. Thus if $C_2 \neq C_2^*$, then $\boldsymbol{r} = \mathsf{F}^{-1}(C_2, td) \neq \mathsf{F}^{-1}(C_2^*, td) = \boldsymbol{r}^*$, and decryption algorithm rejects the ciphertext by re-encoding checking.*

References

1. T. Berger, P. Cayrel, P. Gaborit and A. Otmani. Reducing key length of the mceliece cryptosystem. In *AFRICACRYPT2009*, LNCS, Vol. 5580. pp.77-97, 2009.
2. D. Bernstein, T. Lange and C. Peters. Attacking and defending the mceliece cryptosystem. In *Post-Quantum Cryptography*, 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'09, 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. 14401447, 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.