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

Roohallah Rastaghi

Department of Electrical Engineering, Aeronautical University of Science and Technology, Tehran, Iran.

r.rastaghi59@gmail.com

**Abstract.** Recently, a few CCA2-secure (IND-CCA2) variant of the McEliece cryptosystem in the standard model were introduced. All these schemes are based on Rosrn-Segev approach and lossy trapdoor function and utilize $k$-repetition paradigm. The main drawback of these schemes is that they are need *additional* encryption and have *large* key size compared to the original scheme, which intricate the public-key size problem in the code-based cryptosystem. Furthermore, full CCA2-security of these schemes achieved by using a strongly unforgeable one-time signature scheme, and so, the resulting scheme need *separate* encryption. Therefore, the proposed schemes are not efficient.

In this manuscript, we propose a new and efficient IND-CCA2 variant of the McEliece cryptosystem in the standard model. The main novelty is that, unlike previous approaches, our approach is a generic transformation and can be applied to *any* code-based one-way cryptosystem (both the McEliece and the Niederreiter cryptosystems). Our approach also leads to the elimination of the encryption repetition and using strongly unforgeable one-time signature scheme. This novel approach is more efficient, the publick/secret keys are as in the original scheme and the encryption/decryption complexity are comparable to the original scheme. CCA2-security of the proposed scheme can be reduced in the standard model to the McEliece assumptions. To the best of our knowledge, this is the first variant of the code-based cryptosystem that is IND-CCA2 in the standard model without using $k$-repetition paradigm and strongly unforgeable one-time signature scheme.

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

## 1 Introduction

Post-quantum cryptography (PQC) has obtained great attention in recent years. Code-based cryptography hold a great promise for post-quantum cryptography, as they enjoy very strong security proofs based on average-case hardness [22],

relatively fast and efficient encryption/decryption nature, as well as great simplicity. In code-based cryptography, there are two well-known public key encryption schemes, namely McEliece [13] and Niederreiter [15] cryptosystems. McEliece cryptosystem was the first public key encryption 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 code-based cryptography are deeply explored [2,3,7,9,14].

The semantic security (a.k.a indistinguishability) against adaptive chosen ciphertext attacks (IND-CCA2) is the strongest known notion of security for the public key encryption schemes was introduced by Rackoff and Simon [20]. It is possible to produce IND-CCA2 variants of the code-based cryptosystem 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 two approach for constructing code-based cryptosystems in the standard model.

- *Syndrome decoding.* This construction was presented by Freeman et al. [10], and used Rosen-Segev approach [21] to introduce a correlation-secure trapdoor function related to the hardness of syndrome decoding. Their construction is based on Niederreiter cryptosystem. Because McEliece cryptosystem has some special structure, some general IND-CCA2 conversions such as Rosen-Segev approach cannot be applied to the McEliece cryptosystem and *it is not correlation-secure*. Recently, Preetha Mathew et al. [19] proposed an efficient variant of the Niederreiter scheme based on lossy trapdoor function [17], which avoids the $k$-repetition paradigm. Their idea is similar to Agrawal et. al. [1] approach for simulation of the key-extraction phase in their proof of CPA-security of a (H)IBE in the standard model. But the details and computations are entirely different from [1] .
- *$k$-repetition PKE.* The first IND-CCA2 variant of the McEliece cryptosystem was introduced by Dowsley et al. [5]. They propose a scheme that resembles the Rosen-Segev protocol trying to apply it to the McEliece cryptosystem. This scheme has some ambiguity. The scheme does not rely on a collection of functions but instead defines a structure called *$k$-repetition* public-key encryption ($\text{PKE}_k$) 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 step produces a signature directly on the McEliece ciphertexts instead of introducing a random vector $x$ as in the original Rosen-Segev scheme; therefore an IND-CPA secure variant of McEliece's cryptosystem is necessary to achieve CCA2-security [18]. Recently,

inspired by the Rosen-Segev approach, Döttling et al. [6] showed that Nojima et al. [16] randomized version of the McEliece cryptosystem is $k$-repetition CPA-secure, so, it can obtain CCA2-security in the standard model by using a strongly unforgeable one-time signature scheme.

Cryptosystems based on Rosen-Segev approach are less efficient. These schemes for encrypt *one* bit message[1] need to execute the original encryption algorithm *k-times* in the encryption phase, and *t-times* ($t < k$) in the decryption phase. However, the Döttling et al.'s scheme encrypts many bits as opposed to the single-bit PKE obtained from Rosen-Segev approach. The public/secret keys are *2k-times* larger than the public/secret keys of the original scheme. All the above schemes also use generic transformation such as strongly unforgeable one-time signature scheme to handle CCA2-security related issues. So, the proposed scheme needs *separate* encryptions. On the other hand, all the concrete constructions of lossy trapdoor and correlated inputs functions are based on decisional assumptions. It is widely believed that computational assumptions are more standard than their decisional versions.

### 1.2 Motivation

To date, the existing variants of the code-based cryptosystems (either McEliece or Niederreiter) which are IND-CCA2 in the standard model are based on decisional assumptions such as correlated inputs and lossy trapdoor functions, and utilize $k$-repetition paradigm. In such cryptosystems, the keys are $2k$-times larger than the keys of the original scheme, which intricate the public-key size problem, and a message must be encrypted $k$-times, so, these schemes lead to extremely large key size, ciphertext size, and thus incurring a huge encrypting cost. Although the Preetha Mathew et al. scheme [19] avoids $k$-repetitions, but the encryption/decryption algorithms must be executed 2-times and the public/secret keys are larger than the original Niederreiter scheme. In addition, it yet uses a strongly unforgeable one-time signature scheme to achieve CCA2-security and needs separate encryption. Therefore, how to design an efficient IND-CCA2 code-based cryptosystem in the standard model is still worth of investigation. Less efficiency and impracticality of the proposed IND-CCA2 code-based schemes in the standard model motivate us to investigate new approach for constructing efficient such schemes in the standard model based on computational assumptions.

### 1.3 Our Contributions

To tackle the challenging issues were mentioned in the previous subsection, we introduce a randomized variant of the McEliece cryptosystem and proof

[1] As in [21] we can assume $m$ to be a single bit message, in which case that the scheme describe a hard-core predicate for the McEliece scheme, the protocol easily can be extend to multiple bits plaintexts.

its security in the standard model based on the McEliece assumptions. Our contributions in this paper are:

- Our approach is a generic pre-coding based algorithm. The main novelty is that our approach can be applied to *any* cod-based trapdoor one-way cryptosystems.
- This novel approach, for the first time, leads to the elimination of the encryption repetition and using strongly unforgeable one-time signature schemes in the IND-CCA2 variant of the code-based cryptosystems.
- 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.
- Our CCA2-security proof is based on the assumption that the underlying primitive is a trapdoor one-way function. So, the scheme's *consistency* check can be directly implemented by the *simulator* without having access to some external gap-oracle as in previous schemes [4,5,6,10,12,19]. Thus, our proof technique is fundamentally different from all known approaches to obtain CCA2-security in the code-based cryptosystems.
- Unlike previous schemes, our scheme is based on computational assumptions (i.e. the McEliece assumptions) that is widely believed more standard than their decisional versions.

The paper is organized as follows: in the next section, we briefly explain some mathematical background and definitions. Then, in Section 3, we introduce our proposed scheme. Security and performance analysis of this cryptosystem will be discussed in Section 4. We conclude in Section 5.

## 2 Preliminary

### 2.1 Notation

We will use standard notation. If $\mathbf{x}$ is a string, then $|\mathbf{x}|$ denotes its length and $\mathsf{Lsb}_a(\mathbf{x})$ means the right $a$ bits of $\mathbf{x}$. 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 $R$. If we do not specify $R$ then we implicitly assume that the coins are selected uniformly at random from $\{0,1\}^\infty$. This is denoted $x \leftarrow \mathcal{A}^{\mathcal{O}}(y, z)$. 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.

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** (**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* $w(\mathbf{e}) \leq t$ *.*

**Definition 2** (**General Decoding Assumption**). *Let* $\mathcal{C}$ *be an* $[n,k,d]$*-binary linear code defined by a* $k \times n$ *generator matrix* $\mathsf{G}$ *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.*

*Experiment* $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{GDP}}$

$$\begin{array}{c} \mathbf{c} \in \mathbb{F}_2^k \ \leftarrow \ \mathcal{A}(\mathsf{G}, \mathbf{m} \in \mathbb{F}_2^n) \\ \text{if } \ \mathbf{x} = \mathbf{m} - \mathbf{c}\mathsf{G} \text{ and } \mathsf{wt}(\mathbf{x}) \leq t \end{array}$$

*then* $b \leftarrow 1$ *else* $b \leftarrow 0$

*return* $b$*.*

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

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

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

A public-key can be defined as follows.

**Definition 3** (**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* $\mathcal{C}$.

- $\mathsf{Enc}$ *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.*
- $\mathsf{Dec}$ *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* $\mathsf{Gen}$ *and the random coins* $r$ *used by* $\mathsf{Enc}$.

**Definition 4 (CCA2-security).** *A public-key encryption scheme PKE is secure against adaptive chosen-ciphertext attacks (i.e. IND-CCA2) 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* $k$:

$\mathbf{Exp}^{cca2}_{\mathrm{PKE},\mathcal{A}}(k)$:

$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$

$(m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,.)}(pk) \quad \text{s.t.} \quad |m_0| = |m_1|$

$b \leftarrow \{0, 1\}$

$C^* \leftarrow \mathsf{Enc}(pk, m_b)$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(sk,.)}(C^*, \mathsf{state})$

if $b = b'$ return 1, else return 0.

*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* $b' \leftarrow \mathcal{A}_2^{\mathrm{Dec}(sk,.)}(C^*, state)$. *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*

$$\mathsf{Adv}^{\mathrm{IND-CCA2}}_{\mathrm{PKE},\mathcal{A}}(k) = \left| \Pr[\mathrm{Exp}^{cca2}_{\mathrm{PKE},\mathcal{A}}(k) = 1] - \frac{1}{2} \right|. \tag{1}$$

### 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:

- A $k \times n$ generator matrix $\mathsf{G}$ 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).
- A $k \times k$ random binary non-singular matrix $\mathsf{S}$
- A $n \times n$ random permutation matrix $\mathsf{P}$.

Then, $\mathsf{Gen}$ 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 $\mathbf{c}$ as follows.

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

**Decryption.** To decrypt a ciphertext $\mathbf{c}$, $\mathsf{Dec}_{\mathrm{McE}}(sk, \mathbf{c})$ 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).**[2] *The matrix* $\mathsf{G}$ *output by* $\mathsf{Gen}$ *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.*

[2] Faugère et al. showed that this assumption is not always true for Goppa Code. for more detail see [8]

## 3 The proposed cryptosystem

In this section, we introduce our proposed encryption scheme. Our scheme is an efficient heuristic randomized pre-coding based algorithm and can be applied to *any* code-based trapdoor one-way cryptosystem such as McEliece, Niederreiter and so on. This algorithm uses a random binary string (RBS) for encoding the message to be sent. Encoding includes a permutation and combination on the message bits and performs with an algorithm called *permutation combination algorithm* (PCA). Here, we illustrate our approach based on the McEliece cryptosystem.

### 3.1 Permutation combination algorithm

Suppose we decide to encrypt message $m \in \{0,1\}^n$. For perform a random encoding to the message bits, we uniformly choose a random binary vector $\mathbf{x} = (x_1, \ldots, x_n)$ with Hamming weight $\mathsf{wt}(\mathbf{x}) = h$ such that $n/h$ is an integer. We can divide $m$ into $h$ blocks $m = (b_1 \| b_2 \| \ldots \| b_h)$ with equal binary length $v = n/h$. Then, we perform a random permutation on the message blocks $b_i, 1 \leq i \leq h$ with the following algorithm.

Notice that for any integer $s, 1 \leq s \leq h! - 1$, $s$ can be written as

$$s = \sum_{i=1}^{h} u_i \, (h-i)! \quad 0 \leq u_i \leq h-1.$$

The sequence $\{u_1, \ldots, u_h\}$ is called *factorial carry value* of $s$. Define original sequence $m_0$ as $m_0 = (b_1, b_2, \ldots, b_h)$. Recombine all the elements of the original sequence $m_0$ obtain $h! - 1$ sequences $m_1, \ldots, m_{(h!-1)}$ , which any sequence owns a corresponding factorial carry value. Using the factorial carry value, we can efficiently obtain any sequence $m_s, 1 \leq s \leq h! - 1$ using the following algorithm.

**Algorithm 2: Permutation Combination Algorithm** (PCA).
**Input:** Message $m_0 = (b_1, \ldots, b_h)$ and a random integer $s, 1 \leq s \leq h! - 1$.
**Output:** Encoded message $m' = m_s = (b_1^{'}, \ldots, b_h^{'})$.

1. Write $s$ as $s = \sum_{i=1}^{h} u_i \, (h-i)! \quad 0 \leq u_i \leq h-1$.
2. For $1 \leq i \leq h$
   - if $u_i = 0$,
     $d_i^{'} \leftarrow d_i$;
   - else
     $d_i^{'} \leftarrow d_{i+u_i}$,
     for $1 \leq j \leq u_i$,
     $d_{i+j}^{'} \leftarrow d_i$;

3. Return $m_s = (b_1^{'}, \ldots, b_h^{'})$.

We remark that based on random binary string $\mathbf{x}$, the *number* of the message blocks and the *length* of them can be variable and changed by $\mathbf{x}$.

We illustrate the $\mathsf{PCA}$ algorithm with a small example. Suppose $m = (m_1, \ldots, m_{112})$ and $\mathbf{x} = (x_1, \ldots, x_{112})$ with $\mathsf{wt}(\mathbf{x}) = h = \sum_{i=1}^{112} x_i = 8$. Since $h = 8$, the algorithm divides $m$ into 8 blocks with equal length $v = n/h = 112/8 = 14$. So, we have $m_0 = (\underbrace{m_1, ..., m_{14}}_{b_1} \| \underbrace{m_{15}, \ldots, m_{28}}_{b_2} \| ... \| \underbrace{m_{98}, \ldots, m_{112}}_{b_8})$.
We choose random integer $s, 1 \leq s \leq 8! - 1$, say $s = 2000$. We have

$$2000 = 0 \times 8! + 0 \times 7! + 2 \times 6! + 4 \times 5! + 3 \times 4! + 1 \times 3! + 1 \times 2! + 0 \times 1!$$

Thus, the factorial carry value of $D_{2000}$ is $\{0, 0, 2, 4, 3, 1, 1, 0\}$. Compute sequence $D_{2000}$ with its factorial carry value $\{0, 0, 2, 4, 3, 1, 1, 0\}$.
$0 - -\{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\} \rightarrow b_1$
$0 - -\{b_2, b_3, b_4, b_5, b_6, b_7, b_8\} \rightarrow b_2$
$2 - -\{b_3, b_4, b_5, b_6, b_7, b_8\} \rightarrow b_5$
$4 - -\{b_3, b_4, b_6, b_7, b_8\} \rightarrow b_8$
$3 - -\{b_3, b_4, b_6, b_7\} \rightarrow b_7$
$1 - -\{b_3, b_4, b_6\} \rightarrow b_4$
$1 - -\{b_3, b_6\} \rightarrow b_6$
$0 - -\{b_3\} \rightarrow b_3$
The permutation of sequence $D_{2000}$ is $(b_1 \| b_2 \| b_5 \| b_8 \| b_7 \| b_4 \| b_6 \| b_3)$.

### 3.2 The proposed scheme

Now, we are ready to define our proposed scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

**Key Generation.** Let $\mathsf{Gen}_{\mathrm{McE}}$ be the McEliece system generator. On security parameter $1^k$, the generator $\mathsf{Gen}$ runs $\mathsf{Gen}_{\mathrm{McE}}(1^k)$ to obtain

$$sk = sk_{\mathrm{McE}} \quad \text{and} \quad pk = pk_{\mathrm{McE}}.$$

**Encryption.** To encrypt message $m \in \{0, 1\}^k$, $\mathsf{Enc}(pk)$:

- Choose uniformly random binary string $\mathbf{x} = (x_1, \ldots, x_k)$ with $2 < \mathsf{wt}(\mathbf{x}) = h_x < (k - 2)$ such that $v = k/h_x$ is an integer.
- Set $s = \lceil h_x/2 \rceil \cdot (h_x - 1)! - 1$[3] and execute $\mathsf{PCA}$ algorithm (3.1) for generate encoded message $m' = m_s = (b_1^{'} \| b_2^{'} \| \ldots \| b_{h_x}^{'})$ from message $m$.
- Set $m^{''} \leftarrow \mathsf{Lsb}_{\lceil k/2 \rceil}(m^{'})$ and compute $\mathsf{wt}(m^{''}) = h_{m^{''}}$.
- Suppose $x$ be the corresponding decimal value of $\mathbf{x}$. Compute:

[3] For perform a complete permutation, we can choose the value of $s$ close to the value of $h_x! - 1$. Here, we choose an arbitrary value of $s$ to $s = \lceil h_x/2 \rceil \cdot (h_x - 1)! - 1$.

$$C_1 = x \cdot h_{m''}, \quad C_2 = \mathsf{Enc}_{\mathrm{McE}}(m', pk)$$

**Decryption.** Dec for retrieve message $m$ from $C = (C_1, C_2)$, performs the following steps:

- Computes encoded message $m'$ as $m' = \mathsf{Dec}_{\mathrm{McE}}(C_2, sk)$.
- Set $m^{''} \leftarrow \mathsf{Lsb}_{\lceil k/2 \rceil}(m^{'})$ and compute $\mathsf{wt}(m^{''}) = h_{m^{''}}$
- Computes $x = C_1/h_{m''}$, and reject the ciphertext if $x$ is not an integer. otherwise, checks whether

$$k \stackrel{?}{=} \lfloor \log_2(x) \rfloor + 1 \tag{2}$$

  holds, and rejects if not (*consistency* check). If (2) holds, Computes $h_x = \mathsf{wt}(\mathbf{x})$, $s = \lceil h_x/2 \rceil \cdot (h_x - 1)! - 1$ and $v = k/h_x$.
- The *length* of the message blocks, $v$, and the value of permutation factor $s$ are explicit, so, Dec can extract message blocks $b_i$, $1 \leq i \leq h_x$ from encoded message $m'$ via a reverse permutation.

## 4 Security proof

In this section, we proof the CCA2-security of the proposed cryptosystem built using a pre-coding approach with the McEliece cryptosystem.

**Theorem 1.** *: Suppose* $\Pi_{\mathrm{McE}} = (\mathsf{Gen}_{\mathrm{McE}}, \mathsf{Enc}_{\mathrm{McE}}, \mathsf{Dec}_{\mathrm{McE}})$ *be a McEliece encryption scheme. Then, the proposed scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is IND-CCA2 in the standard model based on McEliece assumption.*

**Proof.** Suppose that $C^* = (C_1^*, C_2^*)$ be the challenge ciphertext. Let $S_i$ be the event that the adversary $\mathcal{A}$ wins in Game $i$. Here is the sequence of games.

**Game 0.** We define Game 0 which is an interactive computation between an *adversary* $\mathcal{A}$ and a *simulator*. This game is usual CCA2 game used to define CCA2-security, in which the simulator provides the adversary's environment. Initially, the simulator runs the key generation algorithm and gives the public-key to the adversary. The adversary submits two messages $m_0, m_1$ with $|m_0| = |m_1|$ to the simulator. The simulator chooses $b \in \{0, 1\}$ at random, and encrypts $m_b$, obtaining the challenge ciphertext $C^* = (C_1^*, C_2^*)$. The simulator gives $C^*$ to the adversary. We denote by $\mathbf{x}^*$, $h_{x^*} = \mathsf{wt}(\mathbf{x}^*)$, $v^* = k/h_{x^*}$, $s^* = \lceil h_{x^*}/2 \rceil \cdot (h_{x^*} - 1)! - 1$, $m'^* = m_{s^*}$, $m''^* = \mathsf{Lsb}_{\lceil k/2 \rceil}(m'^*)$ and $h_{m''^*} = \mathsf{wt}(m''^*)$ the corresponding intermediate quantities computed by the encryption algorithm. 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 $b^{'} \in \{0, 1\}$. Let $S_0$ be the event that $b^{'} = b$. Since Game 0 is identical to the CCA2 game we have that

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

by definition and, our goal is to prove that this quantity is negligible.

**Game 1.** we define Game 1 as identical with Game 0, except that $C_1 = C_1^*$ and $h_{m''} = h_{m''^*}$ while $C_2 \neq C_2^*$[4].

In this game, the adversary $\mathcal{A}_{G1}$ queries on input $(C_1 = C_1^*, h_{m''} = h_{m''^*})$ while $C_2 \neq C_2^*$. In this case, the simulator computes $m' = \mathsf{Dec}_{\mathrm{McE}}(C_2) \neq m'^*$, $x = C_1/h_{m''} = x^*$ and $v = v^*$ and $s = s^*$. Although the blocks length $v$ and the permutation factor $s$ are explicit, but since $m' \neq m'^*$, thus the simulator's outputs is not identical to the $m_b$. Therefore, if the McEliece cryptosystem is secure, then the advantage of adversary $\mathcal{A}_{G1}$ in this game is negligible and we have

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{Adv}^{\mathrm{McE}}_{\mathcal{A}_{G1},\Pi}(k) \tag{3}$$

**Game 2.** Define Game 2 as identical with Game 1, except that $h_{m''} \neq h_{m''^*}$.

In this game, the adversary $\mathcal{A}_{G2}$ queries on input $(C_1 = C_1^*, C_2 \neq C_2^*), h_{m''} \neq h_{m''^*}$. In this case, the simulator computes $m' = \mathsf{Dec}_{\mathrm{McE}}(C_2) \neq m'^*$, $x = C_1/h_{m''} \neq x^*$, $h_x \neq h_{x^*}$, $v \neq v^*$ and $s \neq s^*$. Since $m' \neq m'^*$, $v \neq v^*$ and $s \neq s^*$, thus the simulator's outputs is not identical to $m_b$ and so, the advantage of the adversary $\mathcal{A}_{G2}$ in this game is negligible.

We notice that it is possible for $x \neq x^*$, $h_x = h_{x^*}$. In this case we have $v = v^*$, $s = s^*$ and $m' \neq m'^*$. As we see in the previous game, since $m' \neq m'^*$, the simulator's outputs is not identical to $m_b$ and so, the advantage of adversary $\mathcal{A}_{G2}$ is negligible in this case. We have

$$|\Pr[S_2] - \Pr[S_1]| \leq \mathsf{Adv}^{\mathrm{McE}}_{\mathcal{A}_{G2},\Pi}(k) \tag{4}$$

**Game 3.** Define Game 3 as identical with Game 0, except that $(C_2 = C_2^*)$.

In this game, the adversary $\mathcal{A}_{G3}$ queries on input $C = (C_1 \neq C_1^*, C_2 = C_2^*)$. The simulator takes as input $C_1 \neq C_1^*, C_2 = C_2^*$ and computes $m' = \mathsf{Dec}_{\mathrm{McE}}(C_2) = m'^*$ and $x = C_1/h_{m''^*}$. If $x$ is not a $k$-bit integer, then the simulator rejects $C$ in (2). Else, since $C_1 \neq C_1^*$, thus $x \neq x^*$ and so $h_x \neq h_{x^*}$. We have $v \neq v^*$ and $s \neq s^*$. Since the message blocks length $v$ and the permutation factor $s$ are not explicit, thus the simulator's outputs is not identical to $m_b$ and so, the advantage of the adversary $\mathcal{A}_{G3}$ in this game is negligible. We have

$$|\Pr[S_3] - \Pr[S_0]| \leq \mathsf{Adv}_{\mathcal{A}_{G3},\Pi}(k) \tag{5}$$

It is possible for $x \neq x^*$, $h_x = h_{x^*}$. We discuss this special case in the following lemma.

**Lemma 1.** *There exists an efficient adversary $\mathcal{A}_{G'3}$ such that:*

$$|\Pr[S_3]| = \frac{1}{2}$$

[4] It is possible for $C_2 \neq C_2^*$ and therefore $m' \neq m'^*$, $m''$ and $m''^*$ have the same Hamming weight.

We can easily build an adversary $\mathcal{A}_{G'3}$ who aims to recover $m_b$ from Game 3. In the worst-case, we can assume for $x \neq x^*$ we have $h_x = h_{x^*}$. In this case, the simulator runs on input $C = (C_1 \neq C_1^*, C_2 = C_2^*)$, $h_x = h_{x^*}$ and computes $m' = m'^*$, $x = C_1^*/h_{m''}$, $v = v^*$ and $s = s^*$. If $x$ is not a $k$-bit integer, then the simulator rejects $C$ in (2). Otherwise, the simulator return $b^{'} = b$ and the adversary $\mathcal{A}_{G'3}$ wins the game.

There are exactly $\binom{k}{h_{x^*}} - 1$ cases for $x \neq x^*$ such that $h_x = h_{x^*}$ and so for $C_1 \neq C_1^*$. The probability of succeed $\mathcal{A}_{G'3}$ in this case is equal to

$$\Pr[\mathbf{Exp}^{\mathsf{Dec}(C,sk)=m_b}_{\mathcal{A}_{G'3},\Pi}(k) = 1] < \frac{1}{\binom{k}{h_{x^*}} - 1}.$$

With $2 < h_x < (k-2)$, we have $\Pr[\mathbf{Exp}^{\mathsf{Dec}(C,sk)=m_b}_{\mathcal{A}_{G'3},\Pi}(k) = 1] < 1/2$. So, the advantage of the adversary $\mathcal{A}_{G'3}$ is equal to 0, and we have

$$|\Pr[S_3]| = \frac{1}{2} \tag{6}$$

**Remark 1.** From equation (1), we have

$$\left|\Pr[\mathbf{Exp}^{cca2}_{\mathcal{A},\Pi}(k) = 1]\right| \leq \frac{1}{2} + \mathsf{Adv}^{\mathrm{IND-CCA2}}_{\mathcal{A},\Pi}(k).$$

If the advantage of the adversaries $\mathcal{A}$ is equal to 0, then we have

$$\left|\Pr[\mathbf{Exp}^{cca2}_{\mathcal{A},\Pi}(k) = 1]\right| \leq \frac{1}{2}.$$

*Completing the Proof:* We can write

$$\begin{aligned}|\Pr[S_0]| = |&\Pr[S_0] + \Pr[S_0] - \Pr[S_0] + \Pr[S_1] - \Pr[S_1] + \Pr[S_2] - \Pr[S_2] \\ &+ \Pr[S_3] - \Pr[S_3]|.\end{aligned}$$

So we have

$$\begin{aligned}|\Pr[S_0]| \leq &|\Pr[S_3]| + |\Pr[S_3] - \Pr[S_0]| + |\Pr[S_2] - \Pr[S_0]| + |\Pr[S_2] - \Pr[S_1]| \\ &+ |\Pr[S_1] - \Pr[S_0]|.\end{aligned}$$

We have

$$|\Pr[S_2] - \Pr[S_0]| \leq |\Pr[S_2] - \Pr[S_1]| + |\Pr[S_1] - \Pr[S_0]|. \tag{7}$$

From equations (3, 4, 5, 6, 7) we have:

$$|\Pr[S_0] - 1/2| \leq 2\mathsf{Adv}^{\mathrm{McE}}_{\mathcal{A}_{G1},\Pi}(k) + 2\mathsf{Adv}^{\mathrm{McE}}_{\mathcal{A}_{G2},\Pi}(k) + \mathsf{Adv}_{\mathcal{A}_{G3},\Pi}(k).$$

By assumption, the right-hand side of the above equation is negligible, which finishes the proof.

### 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 computing encoded message is negligible compared to the time for computing $(\mathsf{Enc}_{\mathrm{McE}}, \mathsf{Dec}_{\mathrm{McE}})$. The public/secret keys are as in the original scheme, encryption roughly needs one application of $\mathsf{Enc}_{\mathrm{McE}}$ together a multiplication, and decryption roughly needs one application of $\mathsf{Dec}_{\mathrm{McE}}$ together a division. The comparison of the proposed schemes with existing schemes are presented in table 2.

**Table 2.** Comparison with other code-based CCA-2 cryptosystems

| **Scheme** | **Public-key** | **Secret key** | **Ciphertext** Size | **Encryption Complexity** | **Decryption complexity** |
|---|---|---|---|---|---|
| Dowsley et al.[6] | $2k \times pk_{\mathrm{McE}}$ | $2k \times sk_{\mathrm{McE}}$ | $k \times \mathsf{Ciph}_{\mathrm{McE}}+$ 1 sign | $k \times \mathsf{Enc}_{\mathrm{McE}}+$ $1\,\mathcal{OT}-\mathcal{SS}$ | $1\,\mathsf{Ver}_{\mathcal{OT}-\mathcal{SS}}+$ $1 \times \mathsf{Dec}_{\mathrm{McE}}+$ $t \times \mathsf{Enc}_{\mathrm{McE}}$ |
| Freeman et al.[9] | $2k \times pk_{\mathrm{Nie}}$ | $2k \times sk_{\mathrm{Nie}}$ | $k \times \mathsf{Ciph}_{\mathrm{Nie}}+$ 1 sign | $k \times \mathsf{Enc}_{\mathrm{Nie}}+$ $1\,\mathcal{OT}-\mathcal{SS}$ | $1\,\mathsf{Ver}_{\mathcal{OT}-\mathcal{SS}}+$ $1 \times \mathsf{Dec}_{\mathrm{Nie}}+$ $t \times \mathsf{Enc}_{\mathrm{Nie}}$ |
| Mathew et al.[18] | $1\,pk_{\mathrm{Nie}}+$ $1\,(n \times n)$ Matrix | $2 \times sk_{\mathrm{Nie}}$ | $2 \times \mathsf{Ciph}_{\mathrm{Nie}}+$ 1 sign | $2 \times \mathsf{Enc}_{\mathrm{Nie}}+$ 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 | $1\,pk_{\mathrm{McE}}$ | $1\,sk_{\mathrm{McE}}$ | $\approx 1\,\mathsf{Ciph}_{\mathrm{McE}} + k$ $+\lfloor \log_2(h_{m''}) \rfloor$ | $1\,\mathsf{Enc}_{\mathrm{McE}}+$ PCA + 1P | $1\,\mathsf{Dec}_{\mathrm{McE}}+$ $1\mathrm{D} + 1\,\mathsf{PCA}^{-1}$ |

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, sign: Signature, PCA: Permutation Combination Algorithm (3.1), $\mathsf{PCA}^{-1}$: Reverse Permutation Combination Algorithm and $t \leq k$.

## 5 Conclusion

In this manuscript, we propose a new IND-CCA2 variant of the code-based cryptosystems in the standard model. Unlike previous approaches, our approach is a generic transformation and can be applied to *any* code-based trapdoor one-way cryptosystem such as the McEliece or the Niederreiter cryptosystems. This novel approach leads to the elimination of $k$-repetition paradigm and using strongly

unforgeable one-time signature scheme. The publick/secret keys of the proposed scheme are as in the original scheme and the encryption/decryption complexity are comparable to the original scheme, so, compared to other approaches were introduced today, our approach is more efficient. We showed that CCA2-security of the proposed scheme can be reduced in the standard model to the assumption that the underlying primitive is a trapdoor one-way function (i.e. the McEliece assumptions), without *any* change in the system parameters. To the best of our knowledge, this is the first variant of the code-based cryptosystems that is IND-CCA2 in the standard model without using $k$-repetition paradigm and strongly unforgeable one-time signature scheme.

## References


1. S. Agrawal, D. Boneh, and X. Boyen. Efficient Lattice (H) IBE in the Standard Model. In *EUROCRYPT'2010*, LNCS, Vol. 6110, pp.553-572, 2010.1.1
2. D. Bernstein, T. Lange and C. Peters. Attacking and defending the mceliece cryptosystem. In *Post-Quantum Cryptography*, LNCS, Vol.5299. pp.31-46, 2008.1
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, 2009.1
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, 2012.11.3
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. 240251.1.1 1.3
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, 2012.1.1 1.3
7. J. C. Faugère, A. Otmani, L. Perret and J. P. Tillich. Algebraic Cryptanalysis of McEliece Variants with Compact Keys. In *EUROCRYPT'2010*, LNCS, Vol. 6110, pp. 279-298, 2010. 1
8. J. C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret and J. P. Tillich. A Distinguisher for High Rate McEliece Cryptosystems, Cryptology ePrint Archive: Report 2010/331. `http://eprint.iacr.org/2010/3312`
9. M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In ASIACRYPT'09, LNCS, Vol.5912, pp. 88-105, 2009.1
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, 2010.1.1 1.3
11. K. Kobara and H. Imai. Semantically Secure McEliece Public-Key Cryptosystems Conversions for McEliece PKC. In *PKC'2001*, LNCS, Vol.1992, pp. 19-35, 2001.1
12. 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, 2011.1 1.3
13. R. McEliece. A public-key cryptosystem based on algebraic number theory. *Technical report, Jet Propulsion Laboratory*. DSN Progress Report pp. 42-44, 1978.1
14. R. Misoczki and P. Barreto. Compact mceliece keys from goppa codes. In SAC'2009, LNCS, Vol.5867. pp.376-392, 2009.1

15. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Probl. Control and Inform. Theory*, Vol.15, pp.1934, 1986.1
16. 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. 1.1
17. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC'2008*, pp. 187-196, 2008.1.1
18. 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`1.1
19. 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, 2012.1.2 1.3
20. C. Rackoff and D. Simon. Noninteractive Zero-knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *CRYPTO'91*, LNCS, Vol. 576, pp. 433-444, 1992.1
21. A. Rosen and G. Segev. Chosen-Ciphertext Security via Correlated Products. In *TCC'2009*, LNCS, Vol. 5444, pp. 419-436, 2009.1.1
22. N. Sendrier. The tightness of security reductions in code-based cryptography. In *IEEE, Information Theory Workshop (ITW)*, pp.415-419, 2011.1