A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model

Rafael Dowsley[1], Jörn Müller-Quade[2], Anderson C. A. Nascimento[1]

[1] Department of Electrical Engineering, University of Brasilia.
Campus Universitário Darcy Ribeiro,Brasilia, CEP: 70910-900, Brazil,
Email:rafaeldowsley@redes.unb.br, andclay@ene.unb.br
[2] Universität Karlsruhe, Institut für Algorithmen und Kognitive Systeme.
Am Fasanengarten 5, 76128 Karlsruhe, Germany.
E-mail: muellerq@ira.uka.de

We show that a recently proposed construction by Rosen and Segev can be used for obtaining the first public key encryption scheme based on the McEliece assumptions which is secure against adaptive chosen ciphertext attacks in the standard model.

1 Introduction

Indistinguishability of messages under adaptive chosen ciphertext attacks is the strongest known notion of security for public key encryption schemes (PKE). Many computational assumptions have been used in the literature for obtaining cryptosystems meeting such a strong security requirements. Given one-way trapdoor permutations, we know how to obtain CCA2 security from any semantically secure public key cryptosystem [13, 19, 11]. Efficient constructions are also known based on number-theoretic assumptions [5] or on identity based encryption schemes [2]. Obtaining a CCA2 secure cryptosystem (even an inefficient one) based on the McEliece assumptions in the standard model has been an open problem in this area for quite a while.

Recently, Rosen and Segev proposed an elegant and simple new computational assumption for obtaining CCA2 secure PKEs: *correlated products* [18]. They provided constructions of correlated products based on the existence of certain *lossy trapdoor functions* [15] which in turn can be based on the decisional Diffie-Hellman problem and on Paillier's decisional residuosity problem [15].

In this paper, we show that the ideas of Rosen and Segev can also be applied for obtaining the first construction of a PKE built upon the McEliece assumptions. Based on the definition of correlated products [18], we define a new kind of PKE called k-repetition CPA secure cryptosystem and show that the construction proposed in [18] directly translates to this new scenario. We then show that a randomized version of the MCEliece cryptosystem [14] is k-repetition CPA secure and obtain a CCA2 secure scheme in the standard model. The resulting cryptosystem enciphers many bits as opposed to the single-bit PKE obtained in

[18]. We expand the public and private keys and the ciphertext by a factor of k when compared to the original McEliece PKE. Additionally, our result implies a new construction of correlated products based on the McEliece assumptions.

In a concurrent and independent work [8], Goldwasser and Vaikuntanathan proposed a new CCA-secure public-key encryption scheme based on lattices using the construction by Rosen and Segev. Their scheme assumed that the problem of learning with errors (LWE) is hard [17].

2 Preliminaries

2.1 Notation

If x is a string, then $|x|$ denotes its length, while if $|S|$ represents the cardinality of a set S. If $n \in \mathbb{N}$ then 1^n denotes the string of n ones. $s \leftarrow S$ denotes the operation of choosing an element s of a set S uniformly at random. $w \leftarrow \mathcal{A}(x, y, \ldots)$ represents the act of running the algorithm $\mathcal{A}$ with inputs $x, y, \ldots$ and producing output w. We write $w \leftarrow \mathcal{A}^{\mathcal{O}}(x, y, \ldots)$ for representing an algorithm $\mathcal{A}$ having access to an oracle $\mathcal{O}$. We denote by $\Pr[E]$ the probability that the event E occurs. If a and b are two strings of bits or two matrices, we denote by $a|b$ their concatenation. The transpose of a matrix M is M^T. If a and b are two strings of bits, we denote by $\langle a, b \rangle$ their dot product modulo 2 and by $a \oplus b$ their bitwise XOR. $\mathcal{U}_n$ is an oracle that return a random element of $\{0,1\}^n$.

2.2 Public-Key Encryption Schemes

A Public Key Encryption Scheme (PKE) is defined as follows:

Definition 1. *(Public-Key Encryption). A public-key encryption scheme is a triplet of algorithms (*Gen*,* Enc*,* Dec*) such that:*

- Gen *is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter* 1^n *and outputs a public key* pk *and a secret key* sk*. 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 receives as input a public key* pk *and a message* $\mathsf{m} \in \mathcal{M}$*, and outputs a ciphertext* $\mathsf{c} \in \mathcal{C}$*.*
- Dec *is a deterministic polynomial-time decryption algorithm which takes as input a secret key* sk *and a ciphertext* c*, and outputs either a message* $\mathsf{m} \in \mathcal{M}$ *or an error symbol* $\perp$*.*
- *(Soundness) For any pair of public and private keys generated by* Gen *and any message* $\mathsf{m} \in \mathcal{M}$ *it holds that* $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) = \mathsf{m}$ *with overwhelming probability over the randomness used by* Gen *and* Enc*.*

Below we define indistinguishability against chosen-plaintext attacks (IND-CPA) [7] and against adaptive chosen-ciphertext attacks (IND-CCA2) [16]. Our game definition follows the approach of [9].

Definition 2. *(IND-CPA security). To a two-stage adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *against* PKE *we associate the following experiment* $\mathsf{Exp}^{cpa}_{\mathsf{PKE},\mathcal{A}}(n)$*:*

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$
$(\mathsf{m}_0, \mathsf{m}_1, state) \leftarrow \mathcal{A}_1(\mathsf{pk})$ *s.t.* $|\mathsf{m}_0| = |\mathsf{m}_1|$
$b \leftarrow \{0, 1\}$
$\mathsf{c}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$
$b' \leftarrow \mathcal{A}_2(\mathsf{c}^*, state)$
If $b = b'$ *return* 1 *else return* 0

We define the advantage of $\mathcal{A}$ *in the experiment as*

$$\mathsf{Adv}^{cpa}_{\mathsf{PKE},\mathcal{A}}(n) = |Pr[\mathsf{Exp}^{cpa}_{\mathsf{PKE},\mathcal{A}}(n) = 1] - \tfrac{1}{2}|$$

We say that PKE *is indistinguishable against chosen-plaintext attacks (IND-CPA) if for all probabilist polynomial time (PPT) adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *the advantage of* $\mathcal{A}$ *in the experiment is a negligible function of* n*.*

Definition 3. *(IND-CCA2 security). To a two-stage adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *against* PKE *we associate the following experiment* $\mathsf{Exp}^{cca2}_{\mathsf{PKE},\mathcal{A}}(n)$*:*

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$
$(\mathsf{m}_0, \mathsf{m}_1, state) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(\mathsf{sk},\cdot)}(\mathsf{pk})$ *s.t.* $|\mathsf{m}_0| = |\mathsf{m}_1|$
$b \leftarrow \{0, 1\}$
$\mathsf{c}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$
$b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(\mathsf{sk},\cdot)}(\mathsf{c}^*, state)$
If $b = b'$ *return* 1 *else return* 0

The adversary $\mathcal{A}_2$ *is not allowed to query* $\mathsf{Dec}(\mathsf{sk}, \cdot)$ *with* c^**. We define the advantage of* $\mathcal{A}$ *in the experiment as*

$$\mathsf{Adv}^{cca2}_{\mathsf{PKE},\mathcal{A}}(n) = |Pr[\mathsf{Exp}^{cca2}_{\mathsf{PKE},\mathcal{A}}(n) = 1] - \tfrac{1}{2}|$$

We say that PKE *is indistinguishable against adaptive chosen-ciphertext attacks (IND-CCA2) if for all probabilist polynomial time (PPT) adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *that makes a polynomial number of oracle queries the advantage of* $\mathcal{A}$ *in the experiment is a negligible function of* n*.*

2.3 McEliece Cryptosystem

In this Section we define the McEliece cryptosystem [12]. We closely follow [14]. The McEliece PKE consists of a triplet of probabilistic algorithms $(\mathsf{Gen}_{\mathrm{McE}}, \mathsf{Enc}_{\mathrm{McE}}, \mathsf{Dec}_{\mathrm{McE}})$ such that:

– The probabilistic polynomial-time key generation algorithm, $\mathsf{Gen}_{\mathrm{McE}}$, works as follows:

1. Generate a $l \times n$ generator matrix $\mathbf{G}$ of a Goppa code, where we assume that there is an efficient error-correction algorithm $\mathsf{Correct}$ which can always correct up to t errors.
2. Generate a $l \times l$ random non-singular matrix $\mathbf{S}$.
3. Generate a $n \times n$ random permutation matrix $\mathbf{T}$.
4. Set $\mathbf{P} = \mathbf{SGT}$, $\mathcal{M} = \{0,1\}^l$, $\mathcal{C} = \{0,1\}^n$.
5. Output $\mathsf{pk} = (\mathbf{P}, t, \mathcal{M}, \mathcal{C})$ and $\mathsf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{T})$.

– The probabilistic polynomial-time encryption algorithm, $\mathsf{Enc}_{\mathrm{McE}}$, takes the public-key pk and a plaintext $\mathsf{m} \in \{0,1\}^l$ as input and outputs a ciphertext $\mathsf{c} = \mathsf{m}\mathbf{P} \oplus \mathsf{e}$, where $\mathsf{e} \in \{0,1\}^n$ is a random vector of Hamming weight t.
– The deterministic polynomial-time decryption algorithm, $\mathsf{Dec}_{\mathrm{McE}}$, works as follows:
1. Compute $\mathsf{c}\mathbf{T}^{-1} = (\mathsf{m}\mathbf{S})\mathbf{G} \oplus \mathsf{e}\mathbf{T}^{-1}$, where $\mathbf{T}^{-1}$ denotes the inverse matrix of $\mathbf{T}$.
2. Compute $\mathsf{m}\mathbf{S} = \mathsf{Correct}(\mathsf{c}\mathbf{T}^{-1})$.
3. Output $\mathsf{m} = (\mathsf{m}\mathbf{S})\mathbf{S}^{-1}$.

In our work we use a slightly modified version of the McEliece PKC. Instead of creating an error vector by choosing it randomly from the set of vectors with Hamming weight t, we generate e by choosing each of its bits according to the Bernoulli distribution $\mathcal{B}_\theta$ with parameter $\theta = \frac{t}{n} - \epsilon$ for some $\epsilon > 0$. Clearly, due to the law of large numbers, the resulting error vector should be within the error capabilities of the code.

2.4 McEliece Assumptions

In this subsection, we briefly introduce and discuss the McEliece assumptions.

We assume that there is no efficient algorithm which can distinguish the scrambled (according to the description in the previous subsection) generating matrix of the Goppa code P and a random matrix of the same size. The best algorithm attacking this assumption is by Courtois et al. [4] and it is based on the *support splitting algorithm* [20].

Assumption 4 *There is no PPT algorithm which can distinguish the public-key matrix P of the McEliece cryptosystem from a random matrix of the same size with non-negligible probability.*

We note that this assumption was utilized in [4] to construct a digital signature scheme.

We also assume that the Syndrome Decoding Problem is hard. This problem is known to be NP-complete [1], and all currently known algorithms to solve this problem are exponential. In particular, for small number of errors, the best one was presented by Canteaut and Chabaud [3].

Assumption 5 *The Syndrome Decoding Problem problem is hard for every PPT algorithm.*

This problem is equivalent to the problem of learning parity with noise (LPN). Below we give the definition of LPN problem following the description of [14].

Definition 6. *(LPN problem) Let r, a be binary strings of length l. We consider the Bernoulli distribution $\mathcal{B}_\theta$ with parameter $\theta \in (0, \frac{1}{2})$. Let $\mathcal{Q}_{r,\theta}$ be the following distribution:*

$$\{(a, \langle r, a\rangle \oplus v) | a \leftarrow \{0,1\}^l, v \leftarrow \mathcal{B}_\theta\}$$

For an adversary $\mathcal{A}$ trying to discover the random r, we define its advantage as:

$$\mathsf{Adv}_{\mathsf{LPN}_\theta,\mathcal{A}}(l) = \Pr[\mathcal{A}^{\mathcal{Q}_{r,\theta}} = r | r \leftarrow \{0,1\}^l]$$

The LPN_θ problem with parameter θ is hard if the advantage of all PPT adversaries $\mathcal{A}$ that makes a polynomial number of oracle queries is negligible.

2.5 Admissible PKE

Below we define admissible PKEs which are known to imply IND-CPA security [14]. In the following, $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}, \mathsf{r})$ denotes a public key encryption scheme enciphering a message m with a public key pk and randomness r.

Definition 7. *(Admissible PKE [14]) A public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and random space $\mathcal{R}$ is called admissible if there is a pair of deterministic polynomial-time algorithms Enc_1 and Enc_2 satisfying the following properties:*

- *Dividability: Enc_1 takes as input the public key pk and $\mathsf{r} \in \mathcal{R}$, and outputs a $p(n)$ bit-string. Enc_2 takes as input the public key pk and $\mathsf{m} \in \mathcal{M}$, and outputs a $p(n)$ bit-string. Here p is some polynomial in n. Then for any pk generated by Gen, $\mathsf{r} \in \mathcal{R}$ and $\mathsf{m} \in \mathcal{M}$, $\mathsf{Enc}_1(\mathsf{pk}, \mathsf{r}) \oplus \mathsf{Enc}_2(\mathsf{pk}, \mathsf{m}) = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}, \mathsf{r})$.*
- *Pseudorandomness: Consider a probabilistic polynomial time adversary $\mathcal{A}$ against PKE, we associate with it the following experiment $\mathsf{Exp}^{ind}_{\mathsf{PKE},\mathcal{A}}(n)$:*

  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$
  $\mathsf{s}_0 \leftarrow \mathcal{U}_{p(n)}$
  $\mathsf{r} \in \mathcal{R}$
  $\mathsf{s}_1 \leftarrow \mathsf{Enc}_1(\mathsf{pk}, \mathsf{r})$
  $b \leftarrow \{0,1\}$
  $b' \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{s}_b)$
  If $b = b'$ return 1 else return 0

  We define the advantage of $\mathcal{A}$ in the experiment as

$$\mathsf{Adv}^{ind}_{\mathsf{PKE},\mathcal{A}}(n) = |Pr[\mathsf{Exp}^{ind}_{\mathsf{PKE},\mathcal{A}}(n) = 1] - \tfrac{1}{2}|$$

  For all probabilist polynomial time (PPT) adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the experiment must be a negligible function of n.

2.6 Signature Schemes

We explain signature schemes (SS) and define one-time strongly unforgeability.

Definition 8. *(Signature Scheme). A signature scheme is a triplet of algorithms (*Gen*,* Sign*,* Ver*) such that:*

- Gen *is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter* 1^n *and outputs a verification key* vk *and a signing key* dsk*. The verification key specifies the message space* $\mathcal{M}$ *and the signature space* $\mathcal{S}$*.*
- Sign *is a (possibly) probabilistic polynomial-time signing algorithm which receives as input a signing key* dsk *and a message* $\mathsf{m} \in \mathcal{M}$*, and outputs a signature* $\sigma \in \mathcal{S}$*.*
- Ver *is a deterministic polynomial-time verification algorithm which takes as input a verification key* vk*, a message* $\mathsf{m} \in \mathcal{M}$ *and a signature* $\sigma \in \mathcal{S}$*, and outputs a bit indicating whether* σ *is a valid signature for* m *or not (i.e., the algorithm outputs* 1 *if it is a valid signature and outputs* 0 *otherwise).*
- *For any pair of signing and verification keys generated by* Gen *and any message* $\mathsf{m} \in \mathcal{M}$ *it holds that* $\mathsf{Ver}(\mathsf{vk}, \mathsf{m}, \mathsf{Sign}(\mathsf{dsk}, \mathsf{m})) = 1$ *with overwhelming probability over the randomness used by* Gen *and* Sign*.*

Definition 9. *(One-Time Strongly Unforgeability). To a two-stage adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *against* SS *we associate the following experiment* $\mathsf{Exp}^{otsu}_{\mathsf{SS},\mathcal{A}}(n)$*:*

$(\mathsf{vk}, \mathsf{dsk}) \leftarrow \mathsf{Gen}(1^n)$
$(\mathsf{m}, state) \leftarrow \mathcal{A}_1(\mathsf{vk})$
$\sigma \leftarrow \mathsf{Sign}(\mathsf{dsk}, \mathsf{m})$
$(\mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}_2(\mathsf{m}, \sigma, state)$
If $\mathsf{Ver}(\mathsf{vk}, \mathsf{m}^*, \sigma^*) = 1$ *and* $(\mathsf{m}^*, \sigma^*) \neq (\mathsf{m}, \sigma)$ *return* 1*, else return* 0

We say that a signature scheme SS *is one-time strongly unforgeable if for all probabilist polynomial time (PPT) adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *the probability that* $\mathsf{Exp}^{otsu}_{\mathsf{SS},\mathcal{A}}(n)$ *outputs* 1 *is a negligible function of* n*.*

3 *k*-repetition PKE

3.1 Definitions

We define a k-repetition Public-Key Encryption.

Definition 10. *(k-repetition Public-Key Encryption). For a* PKE (Gen, Enc, Dec)*, we define the k-repetition public-key encryption scheme* (PKE_k) *as the triplet of algorithms* $(\mathsf{Gen}_k, \mathsf{Enc}_k, \mathsf{Dec}_k)$ *such that:*

- Gen_k *is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter* 1^n *and calls the* PKE*'s key generation algorithm* k *times obtaining the public keys* $(\mathsf{pk}_1, \ldots, \mathsf{pk}_k)$ *and the secret keys* $(\mathsf{sk}_1, \ldots, \mathsf{sk}_k)$. Gen_k *sets the public key as* $pk = (\mathsf{pk}_1, \ldots, \mathsf{pk}_k)$ *and the secret key as* $sk = (\mathsf{sk}_1, \ldots, \mathsf{sk}_k)$.
- Enc_k *is a (possibly) probabilistic polynomial-time encryption algorithm which receives as input a public key* $pk = (\mathsf{pk}_1, \ldots, \mathsf{pk}_k)$ *and a message* $\mathsf{m} \in \mathcal{M}$*, and outputs a ciphertext* $\mathsf{c} = (\mathsf{c}_1, \ldots, \mathsf{c}_k) = (\mathsf{Enc}(\mathsf{pk}_1, \mathsf{m}), \ldots, \mathsf{Enc}(\mathsf{pk}_k, \mathsf{m}))$.
- Dec_k *is a deterministic polynomial-time decryption algorithm which takes as input a secret key* $sk = (\mathsf{sk}_1, \ldots, \mathsf{sk}_k)$ *and a ciphertext* $\mathsf{c} = (\mathsf{c}_1, \ldots, \mathsf{c}_k)$*. It outputs a message* m *if* $\mathsf{Dec}(\mathsf{sk}_1, \mathsf{c}_1), \ldots, \mathsf{Dec}(\mathsf{sk}_k, \mathsf{c}_k)$ *are all equal to some* $\mathsf{m} \in \mathcal{M}$*. Otherwise, it outputs an error symbol* $\perp$.
- *(Soundness) For any* k *pairs of public and private keys generated by* Gen_k *and any message* $\mathsf{m} \in \mathcal{M}$ *it holds that* $\mathsf{Dec}_k(\mathsf{sk}, \mathsf{Enc}_k(\mathsf{pk}, \mathsf{m})) = \mathsf{m}$ *with overwhelming probability over the randomness used by* Gen_k *and* Enc_k.

We also define security properties that the k-repetition Public-Key Encryption scheme used in the next sections should meet.

Definition 11. *(Security under uniform* k*-repetition of IND-CPA schemes). We say that* PKE_k *(built from an IND-CPA secure scheme* PKE*) is secure under uniform* k*-repetition if* PKE_k *is IND-CPA secure.*

Definition 12. *(Verification under uniform* k*-repetition of IND-CPA schemes). We say that* PKE_k *is verifiable under uniform* k*-repetition if given a ciphertext* $\mathsf{c} \in \mathcal{C}$*, the public key* $pk = (\mathsf{pk}_1, \ldots, \mathsf{pk}_k)$ *and any* sk_i *for* $i \in \{1, \ldots, k\}$*, it is possible to verify if* c *is a valid ciphertext.*

3.2 IND-CCA2 Security from CPA Secure k-repetition PKE

In this subsection we describe the IND-CCA2 secure public key encryption scheme (PKE_{cca2}) and prove its security. We assume the existence of an one-time strongly unforgeable signature scheme and of a PKE_k that is secure and verifiable under uniform k-repetition.

Key Generation: Gen_{cca2} is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter 1^n. Gen_{cca2} does as follows:

1. Calls the PKE's key generation algorithm $2k$ times obtaining the public keys $(\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_k^0, \mathsf{pk}_k^1)$ and the secret keys $(\mathsf{sk}_1^0, \mathsf{sk}_1^1, \ldots, \mathsf{sk}_k^0, \mathsf{sk}_k^1)$.
2. Executes the key generation algorithm of the signature scheme obtaining a signing key dsk^* and a verification key vk^*. Denote by vk_i^* the i-bit of vk^*.
3. Sets the public key as $pk = (\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_k^0, \mathsf{pk}_k^1)$ and the secret key as $sk = (\mathsf{vk}^*, \mathsf{sk}_1^{1-\mathsf{vk}_1^*}, \ldots, \mathsf{sk}_k^{1-\mathsf{vk}_k^*})$.

Encryption: Enc_{cca2} is a (possibly) probabilistic polynomial-time encryption algorithm which receives as input the public key $pk = (\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_k^0, \mathsf{pk}_k^1)$ and a message $\mathsf{m} \in \mathcal{M}$ and proceeds as follows:

1. Executes the key generation algorithm of the signature scheme obtaining a signing key dsk and a verification key vk. Denote by vk_i the i-bit of vk.
2. Computes $\mathsf{c}_i = \mathsf{Enc}(\mathsf{pk}_i^{\mathsf{vk}_i}, \mathsf{m})$ for $i \in \{1, \ldots, k\}$.
3. Computes the signature $\sigma = \mathsf{Sign}(\mathsf{dsk}, (\mathsf{c}_1, \ldots, \mathsf{c}_k))$.
4. Outputs the ciphertext $\mathsf{c} = (\mathsf{c}_1, \ldots, \mathsf{c}_k, \mathsf{vk}, \sigma)$.

Decryption: Dec_{cca2} is a deterministic polynomial-time decryption algorithm which takes as input a secret key $sk = (\mathsf{vk}^*, \mathsf{sk}_1^{1-\mathsf{vk}_1^*}, \ldots, \mathsf{sk}_k^{1-\mathsf{vk}_k^*})$ and a ciphertext $\mathsf{c} = (\mathsf{c}_1, \ldots, \mathsf{c}_k, \mathsf{vk}, \sigma)$ and proceeds as follows:

1. If $\mathsf{vk} = \mathsf{vk}^*$ or $\mathsf{Ver}(\mathsf{vk}, (\mathsf{c}_1, \ldots, \mathsf{c}_k), \sigma) = 0$, it outputs $\perp$ and halts.
2. For some $i \in \{1, \ldots, k\}$ such that $\mathsf{vk}_i \neq \mathsf{vk}_i^*$, it computes $\mathsf{m} = \mathsf{Dec}(\mathsf{sk}^{\mathsf{vk}_i}, \mathsf{c}_i)$.
3. Verifies if $\mathsf{c}_i = \mathsf{Enc}(\mathsf{pk}_i^{\mathsf{vk}_i}, \mathsf{m})$ for all $i \in \{1, \ldots, k\}$. If the condition is satisfied, it outputs m. Otherwise, it outputs $\perp$.

The probability that $\mathsf{Dec}_{cca2}(\mathsf{sk}, \mathsf{Enc}_{cca2}(\mathsf{pk}, \mathsf{m})) \neq \mathsf{m}$ is the same as the probability that $\mathsf{vk} = \mathsf{vk}^*$, but this probability is negligible since the signature scheme is one-time strongly unforgeable.

As in [18], we can apply a universal one-way hash function to the verification keys (as in [6]) and use $k = n^\epsilon$ for a constant $0 < \epsilon < 1$. For ease of presentation, we do not apply this method in our scheme description.

Theorem 1. *Given that* SS *is a one-time strongly unforgeable signature scheme and that* PKE_k *is secure and verifiable under uniform* k*-repetition, the public key encryption scheme* PKE_{cca2} *is IND-CCA2 secure.*

Proof. In this proof we closely follow [18]. Denote by $\mathcal{A}$ the IND-CCA2 adversary. Let Forge be the event that for some decryption query made by $\mathcal{A}$ we have that $\mathsf{Ver}(\mathsf{vk}, (\mathsf{c}_1, \ldots, \mathsf{c}_k), \sigma) = 1$ and $\mathsf{vk} = \mathsf{vk}^*$. The theorem follow from the two lemmas below.

Lemma 1. $\Pr[\mathsf{Forge}]$ *is negligible.*

Proof. Assume that for a PPT adversary $\mathcal{A}$ against PKE_{cca2} the forge probability ($\Pr[\mathsf{Forge}]$) is non-negligible, then we construct an adversary $\mathcal{A}'$ that forge a signature with the same probability. $\mathcal{A}'$ simulates the IND-CCA2 interaction for $\mathcal{A}$ as follows:

Key Generation: $\mathcal{A}'$ invokes the key generation algorithm of the signature scheme and obtains vk^*. It calls the PKE's key generation algorithm $2k$ times obtaining the public keys $(\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_k^0, \mathsf{pk}_k^1)$ and the secret keys $(\mathsf{sk}_1^0, \mathsf{sk}_1^1, \ldots, \mathsf{sk}_k^0, \mathsf{sk}_k^1)$ and uses vk^* for forming the secret key of PKE_{cca2}. It sends the public key to $\mathcal{A}$.

Decryption Queries: Whenever $\mathcal{A}$ makes a decryption query, $\mathcal{A}'$ proceeds as follows:

1. If $\mathsf{vk} = \mathsf{vk}^*$ and $\mathsf{Ver}(\mathsf{vk}, (\mathsf{c}_1, \ldots, \mathsf{c}_k), \sigma) = 1$, $\mathcal{A}'$ outputs $((\mathsf{c}_1, \ldots, \mathsf{c}_k), \sigma)$ as the forgery and halts.
2. Otherwise, $\mathcal{A}'$ decrypts the ciphertext using the procedures of PKE_{cca2}.

Challenging Query: Whenever $\mathcal{A}$ makes the challenging query with two messages $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ such that $|\mathsf{m}_0| = |\mathsf{m}_1|$, $\mathcal{A}'$ proceeds as follows:

1. Chooses randomly $b \in \{0, 1\}$.
2. Encrypts the message m_b using the procedures of PKE_{cca2}. This is possible because $\mathcal{A}'$ can ask the signature oracle to sign one message, so it asks the oracle to sign the value $(\mathsf{c}_1, \ldots, \mathsf{c}_k)$ obtained during the encryption process.

As long as the event Forge did not occur, the simulation is perfect, so the probability that $\mathcal{A}'$ breaks the one-time strongly unforgeable signature scheme is exactly $\Pr[\mathsf{Forge}]$. Since the signature scheme is strongly unforgeable by assumption, $\Pr[\mathsf{Forge}]$ is negligible for all PPT adversaries against PKE_{cca2}.

Lemma 2. *Given that* Forge *did not occur, the advantage of a PPT adversary* $\mathcal{A}$ *against* PKE_{cca2}*,*

$$|\Pr[\overline{\mathsf{Forge}} \wedge \mathsf{Exp}^{cca2}_{\mathsf{PKE}_{cca2},\mathcal{A}}(n) = 1] - \tfrac{1}{2}|,$$

is negligible.

Proof. Assume that for some PPT adversary $\mathcal{A}$ against PKE_{cca2} we have that $|\Pr[\mathsf{Exp}^{cca2}_{\mathsf{PKE}_{cca2},\mathcal{A}}(n) = 1 \wedge \overline{\mathsf{Forge}}] - \frac{1}{2}|$ is non-negligible, then we construct an adversary $\mathcal{A}'$ that breaks the IND-CPA security of PKE_k. $\mathcal{A}'$ simulates the IND-CCA2 interaction for $\mathcal{A}$ as follows:

Key Generation: $\mathcal{A}'$ receives as input the public key $(\mathsf{pk}_1, \ldots, \mathsf{pk}_k)$ of PKE_k. $\mathcal{A}'$ proceeds as follows:

1. Runs the key generation algorithm of the signature scheme and obtain the verification key vk^* and the signing key dsk^*.
2. Sets $\mathsf{pk}_i^{\mathsf{vk}_i^*} = \mathsf{pk}_i$ for $i \in \{1, \ldots, k\}$.
3. Runs PKE's key generation algorithm k times obtaining the public keys $(\mathsf{pk}_1^{1-\mathsf{vk}_1^*}, \ldots, \mathsf{pk}_k^{1-\mathsf{vk}_1^*})$ and the secret keys $(\mathsf{sk}_1^{1-\mathsf{vk}_1^*}, \ldots, \mathsf{sk}_k^{1-\mathsf{vk}_1^*})$.
4. Sets the public key as $pk = (\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_k^0, \mathsf{pk}_k^1)$ and the secret key as $sk = (\mathsf{vk}^*, \mathsf{sk}_1^{1-\mathsf{vk}_1^*}, \ldots, \mathsf{sk}_k^{1-\mathsf{vk}_k^*})$.
5. Sends the public key to $\mathcal{A}$.

Decryption Queries: Whenever $\mathcal{A}$ makes a decryption query, $\mathcal{A}'$ proceeds as follows:

1. If Forge occurs, then $\mathcal{A}'$ halts.
2. Otherwise, $\mathcal{A}'$ decrypts the ciphertext using the procedures of PKE_{cca2}.

Challenging Query: When $\mathcal{A}$ makes the challenging query with two messages $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ such that $|\mathsf{m}_0| = |\mathsf{m}_1|$, $\mathcal{A}'$ proceeds as follows:

1. Sends m_0 and m_1 to $\mathcal{A}'$ challenging oracle and obtain as response $(\mathsf{c}_1^*, \ldots, \mathsf{c}_k^*)$.
2. Signs $(\mathsf{c}_1^*, \ldots, \mathsf{c}_k^*)$ using dsk^*.

3. Outputs the challenge ciphertext $\mathsf{c}^* = (\mathsf{c}_1^*, \ldots, \mathsf{c}_k^*, \mathsf{vk}^*, \sigma^*)$.

Output: When $\mathcal{A}$ outputs b, $\mathcal{A}'$ also outputs b.

As long as the event Forge does not occur, the advantage of $\mathcal{A}'$ in breaking the IND-CPA-security of PKE_k is the same as the advantage of $\mathcal{A}$ in breaking the IND-CCA2-security of PKE_{cca2}. Since PKE_k is IND-CPA-secure by assumption, we have that PKE_{cca2} is IND-CCA2-secure.

4 The Randomized McEliece Scheme

In [14] it was proved that the cryptosystem obtained by changing the encryption algorithm of the McEliece cryptosystem to encrypt $\mathsf{r}|\mathsf{m}$ (where r is random padding) instead of just encrypting the message m, the so called Randomized McEliece Cryptosystem, is IND-CPA secure.

We modify the encryption algorithm of the Randomized McEliece Cryptosystem as follows. Instead of choosing the error vector randomly from the bit strings of length n and Hamming weight t, we choose each bit of the error vector according to the Bernoulli distribution $\mathcal{B}_\theta$ with parameter $\theta = \frac{t}{n} - \epsilon$ for some $\epsilon > 0$.

By the law of large numbers, for large enough n the Hamming weight of error vector e generated by this procedure will be between $t - 2n\epsilon$ and t with overwhelming probability. So this cryptosystem meets the soundness condition. The IND-CPA security follows from assumptions 4 and 5, since ϵ can be arbitrarily small (given that n is large enough).

4.1 Security of the k-repetition Randomized McEliece

We prove that the modified Randomized McEliece is secure and verifiable under k-repetition, i.e., we prove that the cryptosystem formed by encrypting k times $\mathsf{r}|\mathsf{m}$ with different public and private keys ($\mathsf{PKE}_{k,McE}$) is sound, IND-CPA secure and that it allows the verification of a ciphertext validity given the public keys and one secret key.

By the soundness of each instance, the probability that in one instance $i \in \{1, \ldots, k\}$ a correctly generated ciphertext is incorrectly decoded is negligible. Since k is polynomial, it follows by the union bound that the probability that a correctly generated ciphertext of $\mathsf{PKE}_{k,McE}$ is incorrectly decoded is also negligible. So $\mathsf{PKE}_{k,McE}$ meets the soundness requirement.

In order to prove that the cryptosystem $\mathsf{PKE}_{k,McE}$ is admissible (and so IND-CPA secure [14]), we prove that it meets the pseudorandom property (the dividability follows trivially). Denote by $\mathbf{R}_1, \ldots, \mathbf{R}_k$ random matrices of size $l \times n$, by $\mathbf{P}_1, \ldots, \mathbf{P}_k$ the public key matrices of the McEliece cryptosystem and by $\mathsf{e}_1, \ldots, \mathsf{e}_k$ the error vectors. Define $l_1 = |r|$ and $l_2 = |m|$. Let $\mathbf{R}_{i,1}$ and $\mathbf{R}_{i,2}$ be the $l_1 \times n$ and $l_2 \times n$ sub-matrices of $\mathbf{R}_i$ such that $\mathbf{R}_i^T = \mathbf{R}_{i,1}^T | \mathbf{R}_{i,2}^T$. Define $\mathbf{P}_{i,1}$ and $\mathbf{P}_{i,2}$ similarly. We need a lemma from [10]:

Lemma 3. *Say there exists an algorithm $\mathcal{A}$ making q oracle queries, running in time t, and such that*

$$|\Pr[\mathcal{A}^{\mathcal{Q}_{r,\theta}} = 1 | r \leftarrow \{0,1\}^{l_1}] - \Pr[\mathcal{A}^{\mathcal{U}_{l_1+1}} = 1]| \geq \delta$$

Then there exists an adversary $\mathcal{A}'$ making $q' = O(q\delta^{-2}\log l_1)$ oracle queries, running in time $t' = O(tl_1\delta^{-2}\log l_1)$, and such that

$$\mathsf{Adv}_{\mathsf{LPN}_\theta, \mathcal{A}'} \geq \tfrac{\delta}{4}$$

Setting $q = kn$ in the lemma, we have that $(\mathsf{r}\mathbf{R}_{1,1} \oplus \mathsf{e}_1)| \ldots |(\mathsf{r}\mathbf{R}_{k,1} \oplus \mathsf{e}_k)$ is pseudorandom if the LPN_θ is hard.

Now we prove that substituting the random matrices for the public key matrices of the McEliece cryptosystem does not alter the pseudorandomness of the output $(\mathsf{r}\mathbf{P}_{1,1} \oplus \mathsf{e}_1)| \ldots |(\mathsf{r}\mathbf{P}_{k,1} \oplus \mathsf{e}_k)$.

Lemma 4. $(\mathsf{r}\mathbf{P}_{1,1} \oplus \mathsf{e}_1)| \ldots |(\mathsf{r}\mathbf{P}_{k,1} \oplus \mathsf{e}_k)$ *is pseudorandom.*

Proof. Suppose that some PPT adversary $\mathcal{A}$ has non-negligible advantage in distinguishing $(\mathsf{r}\mathbf{R}_{1,1} \oplus \mathsf{e}_1)| \ldots |(\mathsf{r}\mathbf{R}_{k,1} \oplus \mathsf{e}_k)$ from $(\mathsf{r}\mathbf{P}_{1,1} \oplus \mathsf{e}_1)| \ldots |(\mathsf{r}\mathbf{P}_{k,1} \oplus \mathsf{e}_k)$. Denote them by H_0 and H_k respectively. For $i \in \{1, \ldots, k-1\}$, let H_i be

$$(\mathsf{r}\mathbf{P}_{1,1} \oplus \mathsf{e}_1)| \ldots |(\mathsf{r}\mathbf{P}_{i,1} \oplus \mathsf{e}_i)|(\mathsf{r}\mathbf{R}_{i+1,1} \oplus \mathsf{e}_{i+1})| \ldots |(\mathsf{r}\mathbf{R}_{k,1} \oplus \mathsf{e}_k).$$

Since k is polynomial, by the hybrid argument it is possible to build an adversary $\mathcal{A}'$ that uses $\mathcal{A}$ as a black-box and has a non-negligible advantage in distinguishing H_{i-1} from H_i for some $i \in \{1, \ldots, k\}$, but this would imply that $\mathcal{A}'$ has a non-negligible advantage in distinguishing the public-key matrix P of the McEliece cryptosystem from a random matrix of the same size. By assumption 4, there exists no such $\mathcal{A}'$ and so there cannot exist an adversary $\mathcal{A}$ with non-negligible advantage in distinguishing H_0 from H_k.

Theorem 2. $\mathsf{PKE}_{k,McE}$ *is IND-CPA secure.*

Proof. From the lemmas 3 and 4 we have that $(\mathsf{r}\mathbf{P}_{1,1} \oplus \mathsf{e}_1)| \ldots |(\mathsf{r}\mathbf{P}_{k,1} \oplus \mathsf{e}_k)$ is pseudorandom. So the cryptosystem is admissible. The IND-CPA security of the cryptosystem follows from the fact that an admissible cryptosystem is also IND-CPA secure [14].

Theorem 3. $\mathsf{PKE}_{k,McE}$ *is verifiable under k-repetition.*

Proof. To verify if a ciphertext $(\mathsf{c}_1, \ldots, \mathsf{c}_k)$ is valid given the public keys and any secret key of the McEliece cryptosystem $(\mathbf{S}_j, \mathbf{G}_j, \mathbf{T}_j)$, we simply decrypt c_j obtaining $\mathsf{r}|\mathsf{m}$ and for all $i \in \{1, \ldots, k\}$ compute $\mathsf{c'}_i = (\mathsf{r}|\mathsf{m})\mathbf{P}_i$ and verify if the hamming distance between $\mathsf{c'}_i$ and c_i is less than or equal to t .

Theorem 4. *It is possible to construct an IND-CCA2 secure public key encryption scheme based on McEliece assumptions.*

Proof. Follows directly from theorems 1, 2 and 3.

References

1. E.R. Berlekamp, R.J. McEliece, H.C.A van Tilborg, "On the Inherent Intractability of Certain Coding Problems," IEEE Trans. Inf. Theory, vol. 24, pp.384–386, 1978.
2. R. Canetti, S. Halevi, J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. *EUROCRYPT 2004*. pp. 207-222. 2004.
3. A. Canteaut, F. Chabaud "A new algorithm for finding minimum-weight words in a linear code: application to primitive narrow-sense BCH codes of length 511," IEEE Trans. Inf. Theory, vol. 44(1), pp.367–378, 1998.
4. N. Courtois, M. Finiasz, N. Sendrier: How to Achieve a McEliece Digital Signature Scheme. In: Asiacrypt'2001, LNCS 2248, pp. 157–174, 2001.
5. R. Cramer, V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. *CRYPTO 1998*. pp. 13-25. 1998.
6. D. Dolev, C. Dwork, M. Naor. Non-malleable Cryptography. SIAM J. Comput. 30(2): 391-437 (2000).
7. S. Goldwasser, S. Micali: Probabilistic Encryption. J. Comput. Syst. Sci. 28(2): 270-299 (1984).
8. S. Goldwasser and V. Vaikuntanathan. Correlation-secure trapdoor functions from lattices. Manuscript, 2008.
9. D. Hofheinz, E. Kiltz. Secure Hybrid Encryption from Weakened Key Encapsulation. CRYPTO 2007: 553-571.
10. J. Katz, J. S. Shin: Parallel and Concurrent Security of the HB and HB+ Protocols. EUROCRYPT 2006: 73-87. 2006.
11. Y. Lindell. A Simpler Construction of CCA2-Secure Public-Key Encryption under General Assumptions. *EUROCRYPT 2003*. pp. 241-254. 2003.
12. R.J. McEliece: A Public-Key Cryptosystem Based on Algebraic Coding Theory. In Deep Space Network progress Report, 1978.
13. M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Applications. In 21st STOC, pages 3343, 1989.
14. R. Nojima, H. Imai, K. Kobara and K. Morozov, Semantic Security for the McEliece Cryptosystem without Random Oracles, in Proceedings of International Workshop on Coding and Cryptography (WCC) 2007, pp. 257-268, INRIA, 2007. Journal version in Designs, Codes and Cryptography, Vol. 49, No. 1-3, pp. 289-305, December, 2008.
15. C. Peikert, B. Waters. Lossy trapdoor functions and their applications. *STOC 2008*. pp. 187-196. 2008.
16. C. Rackoff, D. R. Simon: Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. CRYPTO 1991: 433-444.
17. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In STOC, pages 84-93, 2005.
18. A. Rosen and G. Segev. Chosen-Ciphertext Security via Correlated Products. Available at http://eprint.iacr.org/2008/116. 2008.
19. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *40th FOCS*, pages 543553, 1999.
20. N. Sendrier, "Finding the Permutation Between Equivalent Linear Codes: The Support Splitting Algorithm," IEEE Trans. Inf. Theory, 46(4), pp.1193–1203, 2000.