A CCA2 Secure Variant of the McEliece Cryptosystem

Nico Döttling, Rafael Dowsley, Jörn Müller-Quade and Anderson C. A. Nascimento

***Abstract*—The McEliece public-key encryption scheme has become an interesting alternative to cryptosystems based on number-theoretical problems. Differently from RSA and ElGamal, McEliece PKC is not known to be broken by a quantum computer. Moreover, even tough McEliece PKC has a relatively big key size, encryption and decryption operations are rather efficient. In spite of all the recent results in coding theory based cryptosystems, to the date, there are no constructions secure against chosen ciphertext attacks in the standard model – the *de facto* security notion for public-key cryptosystems.**

In this work, we show the first construction of a McEliece based public-key cryptosystem secure against chosen ciphertext attacks in the standard model. Our construction is inspired by a recently proposed technique by Rosen and Segev.

***Index Terms*—Public-key encryption, CCA2 security, McEliece assumptions, standard model**

I. Introduction

Indistinguishability of messages under adaptive chosen ciphertext attacks is one of the strongest known notions 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 notion. Given one-way trapdoor permutations, we know how to obtain CCA2 security from any semantically secure public-key cryptosystem [26], [33], [22]. Efficient constructions are also known based on number-theoretic assumptions [9] or on identity based encryption schemes [6]. 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. We note, however, that secure schemes in the random oracle model have been proposed in [18].

Recently, Rosen and Segev proposed an elegant and simple new computational assumption for obtaining CCA2 secure PKEs: *correlated products* [32]. They provided constructions of correlated products based on the existence of certain *lossy trapdoor functions* [28] which in turn can be based on the decisional Diffie-Hellman problem and on Paillier's decisional residuosity problem [28].

In this paper, we show that ideas similar to those of Rosen and Segev can be applied for obtaining an efficient construction of a CCA2 secure PKE built upon the McEliece assumption. Inspired by the definition of correlated products [32], we define a new kind of PKE called k-repetition CPA secure cryptosystem and provide an adaptation of the construction proposed in [32] to this new scenario. Such cryptosystems can be constructed from very weak (one-way CPA secure) PKEs and randomized encoding functions. In contrast, Rosen and Segev give a more general, however less efficient, construction of correlated secure trapdoor functions from lossy trapdoor functions. We show directly that a randomized version of the McEliece cryptosystem [27] is k-repetition CPA secure and obtain a CCA2 secure scheme in the standard model. The resulting cryptosystem encrypts many bits as opposed to the single-bit PKE obtained in [32]. We expand the public and secret-keys and the ciphertext by a factor of k when compared to the original McEliece PKE.

In a concurrent and independent work [15], Goldwasser and Vaikuntanathan proposed a new CCA2 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 [31].

A direct construction of correlated products based on McEliece and Niederreiter PKEs has been obtained by Persichetti and Galbraith [29] in a subsequent work.

II. Preliminaries

A. Notation

If x is a string, then $|x|$ denotes its length, while $|S|$ 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 returns an uniformly random element of $\{0, 1\}^n$.

We use the notion of randomized encoding-function for functions E that take an input m and random coins s and

Rafael Dowsley is with the Department of Computer Science and Engineering, University of California at San Diego (UCSD), 9500 Gilman Drive, La Jolla, California 92093, USA. Email: rdowsley@cs.ucsd.edu. This work was partially done while the author was with the Department of Electrical Engineering, University of Brasilia. Supported in part by NSF grant CCF-0915675 and by a Focht-Powell fellowship.

Nico Döttling and Jörn Müller-Quade are with the Institute Cryptography and Security, Karlsruhe Institute of Technology. Am Fasanengarten 5, 76128 Karlsruhe, Germany. E-mail: {ndoett,muellerq}@ira.uka.de

Anderson C. A. Nascimento is with the Department of Electrical Engineering, University of Brasilia. Campus Universitário Darcy Ribeiro, Brasília, CEP: 70910-900, Brazil. E-mail: andclay@ene.unb.br.

A preliminary version of this work, enciphering just a single message rather than many possibly correlated ones, has appeared at the proceedings of CT-RSA – 2009

output a randomized representation $\mathsf{E}(\mathsf{m};\mathsf{s})$ from which m can be recovered using a decoding-function D. We will use such randomized encoding-functions to make messages entropic or unguessable.

B. 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, a message $\mathsf{m} \in \mathcal{M}$ and random coins r, and outputs a ciphertext $\mathsf{c} \in \mathcal{C}$. We write $\mathsf{Enc}(\mathsf{pk},\mathsf{m};\mathsf{r})$ to indicate explicitly that the random coins r are used and $\mathsf{Enc}(\mathsf{pk},\mathsf{m})$ 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, and outputs either a message $\mathsf{m} \in \mathcal{M}$ or an error symbol $\perp$.
- (Completeness) For any pair of public and secret-keys generated by Gen and any message $\mathsf{m} \in \mathcal{M}$ it holds that $\mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},\mathsf{m};\mathsf{r})) = \mathsf{m}$ with overwhelming probability over the randomness used by Gen and the random coins r used by Enc.

A basic security notion for public-key encryption schemes is One-Wayness under chosen-plaintext attacks (OW-CPA). This notion states that every PPT-adversary $\mathcal{A}$, given a public-key pk and a ciphertext c of a uniformly chosen message $\mathsf{m} \in \mathcal{M}$, has only negligible probability of recovering the message m (The probability runs over the random coins used to generate the public and secret-keys, the choice of m and the coins of $\mathcal{A}$).

Below we define the standard security notions for public-key encryption schemes, namely, indistinguishability against chosen-plaintext attacks (IND-CPA) [14] and against adaptive chosen-ciphertext attacks (IND-CCA2) [30]. Our game definition follows the approach of [16].

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

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

We say that PKE is indistinguishable against chosen-plaintext attacks (IND-CPA) if for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ the advantage of $\mathcal{A}$ in the above 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

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

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

C. McEliece Cryptosystem

In this Section we define the basic McEliece cryptosystem [24], following [35] and [27]. Let $\mathcal{F}_{n,t}$ be a family of binary linear error-correcting codes given by two parameters n and t. Each code $C \in \mathcal{F}_{n,t}$ has code length n and minimum distance greater than $2t$. We further assume that there exists an efficient probabilistic algorithm $\mathsf{Generate}_{n,t}$ that samples a code $C \in \mathcal{F}_{n,t}$ represented by a generator-matrix $\mathbf{G}_C$ of dimensions $l \times n$ together with an efficient decoding procedure Decode_C that can correct up to t errors.

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 $\mathsf{Gen}_{\mathrm{McE}}$, computes $(\mathbf{G}_C, \mathsf{Decode}_C) \leftarrow \mathsf{Generate}_{n,t}()$, sets $\mathsf{pk} = \mathbf{G}_C$ and $\mathsf{sk} = \mathsf{Decode}_C$ and outputs $(\mathsf{pk},\mathsf{sk})$.
- The probabilistic polynomial-time encryption algorithm $\mathsf{Enc}_{\mathrm{McE}}$, takes the public-key $\mathsf{pk} = \mathbf{G}_C$ and a plaintext $\mathsf{m} \in \mathbb{F}_2^l$ as input and outputs a ciphertext $\mathsf{c} = \mathsf{m}\mathbf{G}_C \oplus \mathsf{e}$, where $\mathsf{e} \in \{0,1\}^n$ is a random vector of Hamming-weight t.
- The deterministic polynomial-time decryption algorithm $\mathsf{Dec}_{\mathrm{McE}}$, takes the secret-key $\mathsf{sk} = \mathsf{Decode}_C$ and a ciphertext $\mathsf{c} \in \mathbb{F}_2^n$, computes $\mathsf{m} = \mathsf{Decode}_C(\mathsf{c})$ and outputs m.

This basic variant of the McEliece cryptosystem is OW-CPA secure (for a proof see [35] Proposition 3.1), given that matrices $\mathbf{G}_C$ generated by $\mathsf{Generate}_{n,t}$ are pseudorandom

(Assumption 4 below) and decoding random linear codes is hard when the noise vector has hamming weight t.

There exist several optimization for the basic scheme, mainly improving the size of the public-key. Biswas and Sendrier [5] show that the public generator-matrix $\mathbf{G}$ can be reduced to row echelon form, reducing the size of the public-key from $l \cdot n$ to $l \cdot (n-l)$ bits. However, we cannot adopt this optimization into our scheme of section IV[1], as it implies a simple attack compromising IND-CPA security[2] (whereas [5] prove OW-CPA security).

In this work we use a slightly modified version of the basic McEliece PKE scheme. Instead of sampling an error vector e 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, a simple argument based on the Chernoff bound gives us that the resulting error vector should be within the error capabilities of the code but for a negligible probability in n. The reason for using this error-distribution is that one of our proofs utilizes the fact that the concatenation $\mathsf{e}_1|\mathsf{e}_2$ of two Bernoulli-distributed vectors e_1 and e_2 is again Bernoulli distributed. Clearly, it is not the case that $\mathsf{e}_1|\mathsf{e}_2$ is a uniformly chosen vector of Hamming-weight $2t$ if each e_1 and e_2 are uniformly chosen with Hamming-weight t.

Using the Bernoulli error-distribution, we base the security of our scheme on the pseudorandomness of the McEliece matrices $\mathbf{G}$ and the pseudorandomness of the learning parity with noise (LPN) problem (see below).

D. McEliece Assumptions and Attacks

In this subsection, we discuss the hardness assumptions for the McEliece cryptosystem. Let $\mathcal{F}_{n,t}$ be a family of codes together with a generation-algorithm $\mathsf{Generate}_{n,t}$ as above and let $\mathbf{G}_C$ be the corresponding generator-matrices. An adversary can attack the McEliece cryptosystem in two ways: either he can try to discover the underlying structure which would allow him to decode efficiently or he can try to run a generic decoding algorithm. This high-level intuition that there are two different ways of attacking the cryptosystem can be formalized [35]. Accordingly, the security of the cryptosystem is based on two security assumptions.

The first assumption states that for certain families $\mathcal{F}_{n,t}$, the distribution of generator-matrices $\mathbf{G}_C$ output by $\mathsf{Generate}_{n,t}$ is pseudorandom. Let l be the dimension of the codes in $\mathcal{F}_{n,t}$.

Assumption 4: Let $\mathbf{G}_C$ be distributed by $(\mathbf{G}_C, \mathsf{Decode}_C) \leftarrow \mathsf{Generate}_{n,t}()$ and $\mathbf{R}$ be distributed by $\mathbf{R} \leftarrow \mathcal{U}(\mathbb{F}_2^{k\times n})$. For every PPT algorithm $\mathcal{A}$ it holds that

$$|\Pr[\mathcal{A}(\mathbf{G}_C) = 1] - \Pr[\mathcal{A}(\mathbf{R}) = 1]| < \mathsf{negl}(n).$$

In the classical instantiation of the McEliece cryptosystem, $\mathcal{F}_{n,t}$ is chosen to be the family of irreducible binary Goppa-codes of length $n = 2^m$ and dimension $l = n - tm$. For this instantiation, an efficient distinguisher was built for the case of high-rate codes [11], [12] (i.e., codes where the rate are very close to 1). But, for codes that do not have a high-rate, no generalization of the previous distinguisher is known and the best known attacks [8], [23] are based on the *support splitting algorithm* [34] and have exponential runtime. Therefore, one should be careful when choosing the parameters of the Goppa-codes, but for encryption schemes it is possible to use codes that do not have high-rate.

The second security assumption is the difficulty of the *decoding problem* (a classical problem in coding theory), or equivalently, the difficulty of the *learning parity with noise* (LPN) problem (a classical problem in learning theory). The best known algorithms for decoding a random linear code are based on the *information set decoding* technique [20], [21], [36]. Over the years, there have been improvements in the running time [7], [3], [13], [4], [25], [1], but the best algorithms still run in exponential time.

Below we give the definition of LPN problem following the description of [27].

Definition 5: (LPN search problem). Let s be a random binary string of length l. We consider the Bernoulli distribution $\mathcal{B}_\theta$ with parameter $\theta \in (0, \frac{1}{2})$. Let $\mathcal{Q}_{s,\theta}$ be the following distribution:

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

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

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

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

Katz and Shin [17] introduce a distinguishing variant of the LPN-problem, which is more useful in the context of encryption schemes.

Definition 6: (LPNDP, LPN distinguishing problem). Let s, a be binary strings of length l. Let further $\mathcal{Q}_{s,\theta}$ be as in Definition 5. Let $\mathcal{A}$ be a PPT-adversary. The distinguishing-advantage of $\mathcal{A}$ between $\mathcal{Q}_{s,\theta}$ and the uniform distribution $\mathcal{U}_{l+1}$ is defined as

$$\mathsf{Adv}_{\mathsf{LPNDP}_\theta,\mathcal{A}}(l) = \\ \left|\Pr\left[\mathcal{A}^{\mathcal{Q}_{s,\theta}} = 1 | s \leftarrow \{0,1\}^l\right] - \Pr\left[\mathcal{A}^{\mathcal{U}_{l+1}} = 1\right]\right|$$

The LPNDP_θ with parameter θ is hard if the advantage of all PPT adversaries $\mathcal{A}$ is negligible.

Further, [17] show that the LPN-distinguishing problem is as hard as the LPN search-problem with similar parameters.

Lemma 1: ([17]) Say there exists an algorithm $\mathcal{A}$ making q oracle queries, running in time t, and such that

$$\mathsf{Adv}_{\mathsf{LPNDP}_\theta,\mathcal{A}}(l) \geq \delta$$

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

$$\mathsf{Adv}_{\mathsf{LPN}_\theta,\mathcal{A}'}(l) \geq \frac{\delta}{4}$$

The reader should be aware that in the current state of the art, the average-case hardness of these two assumptions, as

[1] Neither is it possible for the scheme of [27], on which our k-repetition McEliece scheme is based upon.

[2] The scheme of [27] encrypts by computing $\mathsf{c} = (\mathsf{m}|\mathsf{s}) \cdot \mathbf{G} \oplus \mathsf{e})$. If $\mathbf{G}$ is in row-echelon form, $\mathsf{m} \oplus \mathsf{e}'$ is a prefix of c, where e' is a prefix of e. Thus an IND-CPA adversary can distinguish between the encryptions of two plaintexts m_0 and m_1 by checking whether the prefix of c^* is closer to m_0 or m_1.

(m,t)	plaintext size	ciphertext size	security (key)
(10,50)	524	1024	491
(11,32)	1696	2048	344
(12,40)	3616	4096	471

Fig. 1. A table of McEliece key parameters and security estimates taken from [35].

well as all other assumptions used in public-key cryptography, cannot be reduced to the worst-case hardness of a NP-hard problem[3] (and even if that was the case, we do not even know if $\mathcal{P} \neq \mathcal{NP}$). The confidence on the hardness of solving all these problems on average-case (that is what cryptography really needs) comes from the lack of efficient solutions despite the efforts of the scientific community over the years. But more studies are, of course, necessary in order to better assess the difficulties of such problems. We should highlight that when compared to cryptosystems based on number-theoretical assumptions such as the hardness of factoring or of computing the discrete-log, the cryptosystems based on coding and lattice assumptions have the advantage that no efficient quantum algorithm breaking the assumptions is known. One should also be careful when implementing the McEliece cryptosystem as to avoid side-channel attacks [37].

E. Signature Schemes

Now we define signature schemes (SS) and the security notion called one-time strong unforgeability.

Definition 7: (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).
- (Completeness) 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 8: (One-Time Strong 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. One-way functions are sufficient to construct existentially unforgeable one-time signature schemes [19], [26].

III. k-REPETITION PKE

A. Definitions

We now define a k-repetition Public-Key Encryption.

Definition 9: (k-repetition Public-Key Encryption). For a PKE (Gen, Enc, Dec) and a randomized encoding-function E with a decoding-function D, we define the k-repetition public-key encryption scheme (PKE_k) as the triplet of algorithms (Gen_k, Enc_k, 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 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 $\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_k)$ and the secret-key as $\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_k)$.
- Enc_k is a probabilistic polynomial-time encryption algorithm which receives as input a public-key $\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_k)$, a message $\mathsf{m} \in \mathcal{M}$ and coins s and $r_1, \ldots, r_k$, and outputs a ciphertext $\mathsf{c} = (\mathsf{c}_1, \ldots, \mathsf{c}_k) = (\mathsf{Enc}(\mathsf{pk}_1, \mathsf{E}(\mathsf{m}; s); r_1), \ldots, \mathsf{Enc}(\mathsf{pk}_k, \mathsf{E}(\mathsf{m}; s); r_k))$.
- Dec_k is a deterministic polynomial-time decryption algorithm which takes as input a secret-key $\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_k)$ and a ciphertext $\mathsf{c} = (\mathsf{c}_1, \ldots, \mathsf{c}_k)$. It outputs a message m if $\mathsf{D}(\mathsf{Dec}(\mathsf{sk}_1, \mathsf{c}_1)), \ldots, \mathsf{D}(\mathsf{Dec}(\mathsf{sk}_k, \mathsf{c}_k))$ are all equal to some $\mathsf{m} \in \mathcal{M}$. Otherwise, it outputs an error symbol $\perp$.
- (Completeness) For any k pairs of public and secret-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 random coins 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 10: (Security under uniform k-repetition of encryption schemes). We say that PKE_k (built from an encryption scheme PKE) is secure under uniform k-repetition if PKE_k is IND-CPA secure.

Definition 11: (Verification under uniform k-repetition of encryption schemes). We say that PKE_k is verifiable under uniform k-repetition if there exists an efficient deterministic algorithm Verify such that given a ciphertext $\mathsf{c} \in \mathcal{C}$, the public-key $\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_k)$ and any sk_i for $i \in \{1, \ldots, k\}$, it

[3] Quite remarkably, some lattice problems enjoy average-case to worst-case reductions, but these are not for problems known to be NP-hard.

holds that if $\mathsf{Verify}(\mathsf{c}, \mathsf{pk}, \mathsf{sk}_i) = 1$ then $\mathsf{Dec}_k(\mathsf{sk}, \mathsf{c}) = \mathsf{m}$ for some $\mathsf{m} \neq \perp$ (i.e. c decrypts to a valid plaintext).

Notice that for the scheme PKE_k to be verifiable, the underlying scheme PKE cannot be IND-CPA secure, as the verification algorithm of PKE_k implies an efficient IND-CPA adversary against PKE. Thus, we may only require that PKE is OW-CPA secure.

B. IND-CCA2 Security from verifiable IND-CPA Secure k-repetition PKE

In this subsection we construct 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 $\mathsf{SS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ and of a PKE_k that is secure and verifiable under uniform k-repetition.

We use the following notation for derived keys: For a public-key $\mathsf{pk} = (\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_k^0, \mathsf{pk}_k^1)$ and a k-bit string vk we write $\mathsf{pk}^{\mathsf{vk}} = (\mathsf{pk}_1^{\mathsf{vk}_1}, \ldots, \mathsf{pk}_k^{\mathsf{vk}_k})$. We will use the same notation for secret-keys sk.

- Key Generation: Gen_{cca2} is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter 1^n. Gen_{cca2} calls PKE's key generation algorithm $2k$ times to obtain public-keys $\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_k^0, \mathsf{pk}_k^1$ and secret-keys $\mathsf{sk}_1^0, \mathsf{sk}_1^1, \ldots, \mathsf{sk}_k^0, \mathsf{sk}_k^1$. It sets $\mathsf{pk} = (\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_k^0, \mathsf{pk}_k^1)$, $\mathsf{sk} = (\mathsf{sk}_1^0, \mathsf{sk}_1^1, \ldots, \mathsf{sk}_k^0, \mathsf{sk}_k^1)$ and outputs $\mathsf{pk}, \mathsf{sk})$
- Encryption: Enc_{cca2} is a probabilistic polynomial-time encryption algorithm which receives as input the public-key $\mathsf{pk} = (\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_k^0, \mathsf{pk}_k^1)$ 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.
  2) Compute $\mathsf{c}' = \mathsf{Enc}_k(\mathsf{pk}^{\mathsf{vk}}, \mathsf{m}; r)$ where r are random coins.
  3) Computes the signature $\sigma = \mathsf{Sign}(\mathsf{dsk}, \mathsf{c}')$.
  4) Outputs the ciphertext $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$.
- Decryption: Dec_{cca2} is a deterministic polynomial-time decryption algorithm which takes as input a secret-key $\mathsf{sk} = (\mathsf{sk}_1^0, \mathsf{sk}_1^1, \ldots, \mathsf{sk}_k^0, \mathsf{sk}_k^1)$ and a ciphertext $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$ and proceeds as follows:
  1) If $\mathsf{Ver}(\mathsf{vk}, \mathsf{c}', \sigma) = 0$, it outputs $\perp$ and halts.
  2) It computes and outputs $\mathsf{m} = \mathsf{Dec}_k(\mathsf{sk}^{\mathsf{vk}}, \mathsf{c}')$.

Note that if c' is an invalid ciphertext (i.e. not all c_i' decrypt to the same plaintext), then Dec_{cca2} outputs $\perp$ as Dec_k outputs $\perp$.

As in [32], we can apply a universal one-way hash function to the verification keys (as in [10]) and use $k = n^{\epsilon}$ for a constant $0 < \epsilon < 1$. Note that the hash function in question need not be modeled as a random oracle. For ease of presentation, we do not apply this method in our scheme description.

Theorem 1: Given that SS is an one-time strongly unforgeable signature scheme and that PKE_k is IND-CPA 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 [32]. Denote by $\mathcal{A}$ the IND-CCA2 adversary. Consider the following sequence of games.

- **Game 1** This is the IND-CCA2 game.
- **Game 2** Same as game 1, except that the signature-keys $(\mathsf{vk}^*, \mathsf{dsk}^*)$ that are used for the challenge-ciphertext c^* are generated before the interaction with $\mathcal{A}$ starts. Further, game 2 always outputs $\perp$ if $\mathcal{A}$ sends a decryption query $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$ with $\mathsf{vk} = \mathsf{vk}^*$.

We will now establish the remaining steps in two lemmata.

Lemma 2: It holds that $\mathsf{view}_{\mathsf{Game1}}(\mathcal{A}) \approx_c \mathsf{view}_{\mathsf{Game2}}(\mathcal{A})$, given that $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ is an one-time strongly unforgeable signature scheme.

Proof: Given that $\mathcal{A}$ does not send a valid decryption query $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$ with $\mathsf{vk} = \mathsf{vk}^*$ and $\mathsf{c} \neq \mathsf{c}^*$, $\mathcal{A}$'s views in game 1 and game 2 are identical. Thus, in order to distinguish game 1 and game 2 $\mathcal{A}$ must send a valid decryption query $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$ with $\mathsf{vk} = \mathsf{vk}^*$ and $\mathsf{c} \neq \mathsf{c}^*$. We will use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ against the one-time strong unforgeability of the signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$. $\mathcal{B}$ basically simulates the interaction of game 2 with $\mathcal{A}$, however, instead of generating vk^* itself, it uses the vk^* obtained from the one-time strong unforgeability experiment. Furthermore, B generates the signature σ for the challenge-ciphertext c^* by using its signing oracle provided by the one-time strong unforgeability game. Whenever $\mathcal{A}$ sends a valid decryption query $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$ with $\mathsf{vk} = \mathsf{vk}^*$ and $\mathsf{c} \neq \mathsf{c}^*$, $\mathcal{B}$ terminates and outputs (c', σ). Obviously, $\mathcal{A}$'s output is identically distributed in Game 2 and $\mathcal{B}$'s simulation. Therefore, if $\mathcal{A}$ distinguishes between game 1 and game 2 with non-negligible advantage ϵ, then $\mathcal{B}$'s probability of forging a signature is also ϵ, thus breaking the one-time strong unforgeability of $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$. ■

Lemma 3: It holds that $\mathsf{Adv}_{\mathsf{Game2}}(\mathcal{A})$ is negligible in the security parameter, given that PKE_k is verifiable and IND-CPA secure under uniform k-repetition.

Proof: Assume that $\mathsf{Adv}_{\mathsf{Game2}}(\mathcal{A}) \geq \epsilon$ for some non-negligible ϵ. We will now construct an IND-CPA adversary $\mathcal{B}$ against PKE_k that breaks the IND-CPA security of PKE_k with advantage ϵ. Instead of generating pk like game 2, $\mathcal{B}$ proceeds as follows. Let $\mathsf{pk}^* = (\mathsf{pk}_1^*, \ldots, \mathsf{pk}_k^*)$ be the public-key provided by the IND-CPA experiment to $\mathcal{B}$. $\mathcal{B}$ first generates a pair of keys for the signature scheme $(\mathsf{vk}^*, \mathsf{dsk}^*) \leftarrow \mathsf{Gen}(1^n)$. Then, the public-key pk is formed by setting $\mathsf{pk}^{\mathsf{vk}^*} = \mathsf{pk}^*$. All remaining components pk_i^j of pk are generated by $(\mathsf{pk}_i^j, \mathsf{sk}_i^j) \leftarrow \mathsf{Gen}(1^n)$, for which $\mathcal{B}$ stores the corresponding sk_i^j. Clearly, the pk generated by $\mathcal{B}$ is identically distributed to the pk generated by game 2, as the Gen-algorithm of PKE_k generates the components of pk independently. Now, whenever $\mathcal{A}$ sends a decryption query $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$, where $\mathsf{vk} \neq \mathsf{vk}^*$ (decryption queries with $\mathsf{vk} = \mathsf{vk}^*$ are not answered by game 2), $\mathcal{B}$ picks an index i with $\mathsf{vk}_i \neq \mathsf{vk}_i^*$ and checks if $\mathsf{Verify}(\mathsf{c}', \mathsf{pk}, \mathsf{sk}_i^{\mathsf{vk}_i}) = 1$, if not it outputs $\perp$. Otherwise it computes $\mathsf{m} = \mathsf{D}(\mathsf{Dec}(\mathsf{sk}_i, \mathsf{c}_i'))$. Verifiability guarantees that it holds that $\mathsf{Dec}_k(\mathsf{sk}^{\mathsf{vk}}, \mathsf{c}') = \mathsf{m}$, i.e. the output m is identically distributed as in game 2. When $\mathcal{A}$ sends the challenge-messages $\mathsf{m}_0, \mathsf{m}_1$, $\mathcal{B}$ forwards

$\mathsf{m}_0, \mathsf{m}_1$ to the IND-CPA experiments and receives a challenge-ciphertext $\mathsf{c}^{*\prime}$. $\mathcal{B}$ then computes $\sigma = \mathsf{Sign}(\mathsf{dsk}^*, \mathsf{c}^{*\prime})$ and sends $\mathsf{c}^* = (\mathsf{c}^{*\prime}, \mathsf{vk}^*, \sigma)$ to $\mathcal{A}$. This c^* is identically distributed as in game 2. Once $\mathcal{A}$ produces output, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. Putting it all together, $\mathcal{A}$'s views are identically distributed in game 2 and in the simulation of $\mathcal{B}$. Therefore it holds that $\mathsf{Adv}_{\mathsf{IND-CPA}}(\mathcal{B}) = \mathsf{Adv}_{\mathsf{Game2}}(\mathcal{A}) \geq \epsilon$. Thus $\mathcal{B}$ breaks the IND-CPA security of PKE_k with non-negligible advantage ϵ, contradicting the assumption. ■

Plugging Lemma 2 and Lemma 3 together immediately establishes that any PPT IND-CCA2 adversary $\mathcal{A}$ has at most negligible advantage in winning the IND-CCA2 experiment for the scheme $\mathsf{PKE}_{\mathsf{cca2}}$. ■

IV. A Verifiable k-repetition McEliece Scheme

In this section, we will instantiate a verifiable k-repetition encryption scheme $\mathsf{PKE}_{\mathsf{McE,k}} = (\mathsf{Gen}_{\mathsf{McE,k}}, \mathsf{Enc}_{\mathsf{McE,k}}, \mathsf{Dec}_{\mathsf{McE,k}})$ based on the McEliece cryptosystem.

In [27] it was proved that the cryptosystem obtained by changing the encryption algorithm of the McEliece cryptosystem to encrypt $\mathsf{s}|\mathsf{m}$ (where s is random padding) instead of just encrypting the message m (the so called Randomized McEliece cryptosystem) is IND-CPA secure, if $|s|$ is chosen sufficiently large for the LPNDP to be hard (e.g. linear in the security-parameter n). We will therefore use the randomized encoding-function $\mathsf{E}(\mathsf{m};\mathsf{s}) = \mathsf{s}|\mathsf{m}$ (with $|s| \in \Omega(n)$) in our verifiable k-repetition McEliece scheme. As basis scheme PKE for our verifiable k-repetition McEliece scheme we use the OW-CPA secure textbook McEliece with a Bernoulli error-distribution.

The verification algorithm $\mathsf{Verifiy}_{\mathsf{McE}}(\mathsf{c}, \mathsf{pk}, \mathsf{sk}_i)$ works as follows. Given a secret-key sk_i from the secret-key vector sk, it first decrypts the i-th component of c by $\mathsf{x} = \mathsf{Dec}_{\mathsf{McE}}(\mathsf{sk}_i, \mathsf{c}_i)$. Then, for all $j = 1, \ldots, k$, it checks whether the vectors $\mathsf{c}_j \oplus \mathsf{x}\mathbf{G}_j$ have a Hamming-weight smaller than t, where $\mathbf{G}_j$ is the generator-matrix given in pk_j. If so, $\mathsf{Verify}_{\mathsf{McE}}$ outputs 1, otherwise 0. Clearly, if $\mathsf{Verify}_{\mathsf{McE}}$ accepts, then all ciphertexts c_j are close enough to the respective codewords xG_j, i.e. invoking $\mathsf{Dec}_{\mathsf{McE}}(\mathsf{sk}_j, \mathsf{c}_j)$ would also output x. Therefore, we have that $\mathsf{Verifiy}_{\mathsf{McE}}(\mathsf{c}, \mathsf{pk}, \mathsf{sk}_i) = 1$, if and only if $\mathsf{Dec}_{\mathsf{McE,k}}(\mathsf{sk}, \mathsf{c}) = \mathsf{m}$ for some $\mathsf{m} \in \mathcal{M}$.

A. Security of the k-repetition Randomized McEliece

We now prove that the modified Randomized McEliece is IND-CPA secure under k-repetition.

By the completeness 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 completeness requirement.

Denote by $\mathbf{R}_1, \ldots, \mathbf{R}_k$ random matrices of size $l \times n$, by $\mathbf{G}_1, \ldots, \mathbf{G}_k$ the public-key matrices of the McEliece cryptosystem and by $\mathsf{e}_1, \ldots, \mathsf{e}_k$ the error vectors. Define $l_1 = |\mathsf{s}|$ and $l_2 = |\mathsf{m}|$. 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 $\mathbf{G}_{i,1}$ and $\mathbf{G}_{i,2}$ similarly.

Lemma 4: The scheme $\mathsf{PKE}_{\mathsf{McE},k}$ is IND-CPA secure, given that both the McEliece assumption and the LPNDP assumption hold.

Proof: Let $\mathcal{A}$ be an IND-CPA adversary against $\mathsf{PKE}_{\mathsf{McE},k}$. Consider the following three games.

- **Game 1** This is the IND-CPA game.
- **Game 2** Same as game 1, except that the components pk_i of the public-key pk are computed by $\mathsf{pk}_i = (\mathbf{R}_i, t, \mathcal{M}, \mathcal{C})$ instead of $\mathsf{pk}_i = (\mathbf{G}_i, t, \mathcal{M}, \mathcal{C})$, where $\mathbf{R}_i$ is a randomly chosen matrix of the same size as $\mathbf{G}_i$
- **Game 3** Same as game 2, except that the components c_i of the challenge-ciphertext c^* are not computed by $\mathsf{c}_i = (\mathsf{s}|\mathsf{m})\mathbf{R}_i \oplus \mathsf{e}_i$ but rather chosen uniformly at random.

Indistinguishability of game 1 and game 2 follows by a simple hybrid-argument using the McEliece assumption, we omit this for the sake of brevity. The indistinguishability of game 2 and game 3 can be established as follows. First observe that it holds that $\mathsf{c}_i = (\mathsf{s}|\mathsf{m})\mathbf{R}_i \oplus \mathsf{e}_i = (\mathsf{s}\mathbf{R}_{i,1} \oplus \mathsf{e}_i) \oplus \mathsf{m}\mathbf{R}_{i,2}$ for $i = 1, \ldots, k$. Setting $\mathbf{R}_1 = \mathbf{R}_{1,1}| \ldots, |\mathbf{R}_{k,1}$, $\mathbf{R}_2 = \mathbf{R}_{1,2}| \ldots, |\mathbf{R}_{k,2}$ and $\mathsf{e} = \mathsf{e}_1| \ldots |\mathsf{e}_k$, we can write $\mathsf{c}^* = (\mathsf{s}\mathbf{R}_1 \oplus \mathsf{e}) \oplus \mathsf{m}\mathbf{R}_2$. Now, the LPNDP assumption allows us to substitute $\mathsf{s}\mathbf{R}_1 \oplus \mathsf{e}$ with a uniformly random distributed vector u, as s and $\mathbf{R}_1$ are uniformly distributed and e is Bernoulli distributed. Therefore $\mathsf{c}^* = \mathsf{u} \oplus \mathsf{m}\mathbf{R}_2$ is also uniformly distributed. Thus we have reached game 3. $\mathcal{A}$'s advantage in game 3 is obviously 0, as the challenge-ciphertext c^* is statistically independent of the challenge bit b. This concludes the proof. ■

V. Generalized Scheme

As in [32], it is possible to generalize the scheme to encrypt correlated messages instead of encrypting k times the same message m. In this Section, we show that a similar approach is possible for our scheme, yielding an IND-CCA2 secure McEliece variant that has asymptotically the same ciphertext expansion as the efficient IND-CPA scheme of [18]. We now present a generalized version of our encryption scheme using a correlated plaintext space.

A. Definitions

Definition 12: (τ-Correlated Messages) We call a tuple of messages $(\mathsf{m}_1, \ldots, \mathsf{m}_k)$ τ-correlated for some constant $0 < \gamma < 1$ and $\tau = (1 - \gamma)k$, if given any τ messages of tuple it is possible to efficiently recover all the messages. We denote the space of such messages tuples by $\mathcal{M}_{\mathsf{Cor}}$.

Basically, τ-correlated messages can be erasure-corrected. Now we define a correlated public-key encryption scheme.

Definition 13: (Correlated Public-Key Encryption). For a PKE (Gen, Enc, Dec) and a randomized encoding-function E that maps from the plaintext-space $\mathcal{M}$ to the correlated plaintext-space $\mathcal{M}_{\mathsf{Cor}}$ (with corresponding decoding-function D), we define the correlated public-key encryption scheme ($\mathsf{PKE}_{\mathsf{Cor}}$) as the triplet of algorithms ($\mathsf{Gen}_{\mathsf{Cor}}$, $\mathsf{Enc}_{\mathsf{Cor}}$, $\mathsf{Dec}_{\mathsf{Cor}}$) such that:

- $\mathsf{Gen}_{\mathsf{Cor}}$ is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter 1^n and calls PKE's key generation algorithm k times obtaining the public-keys $(\mathsf{pk}_1, \dots, \mathsf{pk}_k)$ and the secret-keys $(\mathsf{sk}_1, \dots, \mathsf{sk}_k)$. $\mathsf{Gen}_{\mathsf{Cor}}$ sets the public-key as $\mathsf{pk} = (\mathsf{pk}_1, \dots, \mathsf{pk}_k)$ and the secret-key as $\mathsf{sk} = (\mathsf{sk}_1, \dots, \mathsf{sk}_k)$.
- $\mathsf{Enc}_{\mathsf{Cor}}$ is a probabilistic polynomial-time encryption algorithm which receives as input a public-key $\mathsf{pk} = (\mathsf{pk}_1, \dots, \mathsf{pk}_k)$ and a message $\mathsf{m} \in \mathcal{M}$. The algorithm computes $\tilde{\mathsf{m}} = (\tilde{\mathsf{m}}_1, \dots, \tilde{\mathsf{m}}_k) = \mathsf{E}(\mathsf{m}; s)$ (with fresh random coins s) and outputs the ciphertext $\mathsf{c} = (\mathsf{c}_1, \dots, \mathsf{c}_k) = (\mathsf{Enc}(\mathsf{pk}_1, \tilde{\mathsf{m}}_1), \dots, \mathsf{Enc}(\mathsf{pk}_k, \tilde{\mathsf{m}}_k))$.
- $\mathsf{Dec}_{\mathsf{Cor}}$ is a deterministic polynomial-time decryption algorithm which takes as input a secret-key $\mathsf{sk} = (\mathsf{sk}_1, \dots, \mathsf{sk}_k)$ and a ciphertext $\mathsf{c} = (\mathsf{c}_1, \dots, \mathsf{c}_k)$. It first computes a tuple $\tilde{\mathsf{m}} = (\tilde{\mathsf{m}}_1, \dots, \tilde{\mathsf{m}}_k) \in \mathcal{M}_{\mathsf{Cor}}$, outputs $\mathsf{m} = \mathsf{D}(\tilde{\mathsf{m}})$ if $\tilde{\mathsf{m}} \in \mathcal{M}_{\mathsf{Cor}}$, if not it outputs an error symbol $\perp$.
- (Completeness) For any k pairs of public and secret-keys generated by $\mathsf{Gen}_{\mathsf{Cor}}$ and any message $\mathsf{m} = (\mathsf{m}_1, \dots, \mathsf{m}_k) \in \mathcal{M}_{\mathsf{Cor}}$ it holds that $\mathsf{Dec}_{\mathsf{Cor}}(\mathsf{sk}, \mathsf{Enc}_{\mathsf{Cor}}(\mathsf{pk}, \mathsf{m})) = \mathsf{m}$ with overwhelming probability over the randomness used by $\mathsf{Gen}_{\mathsf{Cor}}$ and $\mathsf{Enc}_{\mathsf{Cor}}$.

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

Definition 14: (Security of Correlated Public-Key Encryption). We say that $\mathsf{PKE}_{\mathsf{Cor}}$ (built from an encryption scheme PKE) is secure if $\mathsf{PKE}_{\mathsf{Cor}}$ is IND-CPA secure.

Definition 15: (τ-Verification). We say that $\mathsf{PKE}_{\mathsf{Cor}}$ is τ-verifiable if the exists a efficient deterministic algorithm Verify, such that given a ciphertext $\mathsf{c} \in \mathcal{C}$, the public-key $\mathsf{pk} = (\mathsf{pk}_1, \dots, \mathsf{pk}_k)$ and any τ distinct secret-keys $\mathsf{sk}_T = (\mathsf{sk}_{t_1}, \dots, \mathsf{sk}_{t_\tau})$ (with $T = \{t_1, \dots, t_\tau\}$), it holds that if $\mathsf{Verify}(\mathsf{c}, \mathsf{pk}, T, \mathsf{sk}_T) = 1$ then $\mathsf{Dec}_{\mathsf{Cor}}(\mathsf{sk}, \mathsf{c}) = \mathsf{m}$ for some $\mathsf{m} \neq \perp$ (i.e. c decrypts to a valid plaintext).

B. IND-CCA2 Security from IND-CPA Secure Correlated PKE

We now describe the IND-CCA2 secure public-key encryption scheme (PKE'_{cca2}) built using the correlated PKE and prove its security. We assume the existence of a correlated PKE, $\mathsf{PKE}_{\mathsf{Cor}}$, that is secure and τ-verifiable. We also use an error correcting code $\mathsf{ECC} : \Sigma^l \rightarrow \Sigma^k$ with minimum distance τ and polynomial-time encoding. Finally, we assume the existence of an one-time strongly unforgeable signature scheme $\mathsf{SS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ in which the verification keys are elements of Σ^l (we assumed that the verification keys are elements of Σ^l only for simplicity, we can use any signature scheme if there is a injective mapping from the set of verification keys to Σ^l).

We will use the following notation: For a codeword $\mathsf{d} = (\mathsf{d}_1, \dots, \mathsf{d}_k) \in \mathsf{ECC}$, set $\mathsf{pk}^{\mathsf{d}} = (\mathsf{pk}_1^{\mathsf{d}_1}, \dots, \mathsf{pk}_k^{\mathsf{d}_k})$. Analogously for sk.

- Key Generation: Gen'_{cca2} is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter 1^n. Gen'_{cca2} proceeds as follows. It calls PKE's key generation algorithm $|\Sigma|k$ times obtaining the public-keys $(\mathsf{pk}_1^1, \dots, \mathsf{pk}_1^{|\Sigma|}, \dots, \mathsf{pk}_k^1, \dots, \mathsf{pk}_k^{|\Sigma|})$ and the secret-keys $(\mathsf{sk}_1^1, \dots, \mathsf{sk}_1^{|\Sigma|}, \dots, \mathsf{sk}_k^1, \dots, \mathsf{sk}_k^{|\Sigma|})$. Outputs $\mathsf{pk} = (\mathsf{pk}_1^1, \dots, \mathsf{pk}_1^{|\Sigma|}, \dots, \mathsf{pk}_k^1, \dots, \mathsf{pk}_k^{|\Sigma|})$ and $\mathsf{sk} = (\mathsf{sk}_1^1, \dots, \mathsf{sk}_1^{|\Sigma|}, \dots, \mathsf{sk}_k^1, \dots, \mathsf{sk}_k^{|\Sigma|})$.
- Encryption: Enc'_{cca2} is a probabilistic polynomial-time encryption algorithm which receives as input the public-key $\mathsf{pk} = (\mathsf{pk}_1^1, \dots, \mathsf{pk}_1^{|\Sigma|}, \dots, \mathsf{pk}_k^1, \dots, \mathsf{pk}_k^{|\Sigma|})$ and a message $\mathsf{m} = (\mathsf{m}_1, \dots, \mathsf{m}_k) \in \mathcal{M}$ and proceeds as follows:
  1) Executes the key generation algorithm of the signature scheme SS obtaining a signing key dsk and a verification key vk. Computes $\mathsf{d} = \mathsf{ECC}(\mathsf{vk})$. Let d_i denote the i-element of d.
  2) Computes $\mathsf{c}' = \mathsf{Enc}_{\mathsf{Cor}}(\mathsf{pk}^{\mathsf{d}}, \mathsf{m})$.
  3) Computes the signature $\sigma = \mathsf{Sign}(\mathsf{dsk}, \mathsf{c}')$.
  4) Outputs the ciphertext $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$.
- Decryption: Dec'_{cca2} is a deterministic polynomial-time decryption algorithm which takes as input a secret-key $\mathsf{sk} = (\mathsf{sk}_1^1, \dots, \mathsf{sk}_1^{|\Sigma|}, \dots, \mathsf{sk}_k^1, \dots, \mathsf{sk}_k^{|\Sigma|})$ and a ciphertext $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$ and proceeds as follows:
  1) If $\mathsf{Ver}(\mathsf{vk}, \mathsf{c}', \sigma) = 0$, it outputs $\perp$ and halts. Otherwise, it performs the following steps.
  2) Compute $\mathsf{d} = \mathsf{ECC}(\mathsf{vk})$.
  3) Compute $\mathsf{m} = \mathsf{Dec}_{\mathsf{Cor}}(\mathsf{sk}^{\mathsf{d}}, \mathsf{c})$ and output m.

Theorem 2: Given that SS is an one-time strongly unforgeable signature scheme and that $\mathsf{PKE}_{\mathsf{Cor}}$ is secure and τ-verifiable, the public-key encryption scheme PKE'_{cca2} is IND-CCA2 secure.

Proof: The proof is almost identical to the proof of theorem 1. Denote by $\mathcal{A}$ the IND-CCA2 adversary. Consider the following two of games.

- **Game 1** This is the IND-CCA2 game.
- **Game 2** Same as game 1, except that the signature-keys $(\mathsf{vk}^*, \mathsf{dsk}^*)$ that are used for the challenge-ciphertext c^* are generated before the interaction with $\mathcal{A}$ starts. Further, game 2 terminates and outputs $\perp$ if $\mathcal{A}$ sends a decryption query with $\mathsf{c} = (\mathsf{c}', \mathsf{vk}, \sigma)$ with $\mathsf{vk} = \mathsf{vk}^*$.

Again, we will split the proof of Theorem 2 in two lemmata.

Lemma 5: From $\mathcal{A}$'s view, game 1 and game 2 are computationally indistinguishable, given that SS is an existentially unforgeable one-time signature-scheme.

We omit the proof, since it is identical to the proof of lemma 2.

Lemma 6: It holds that $\mathsf{Adv}_{\mathsf{Game2}}(\mathcal{A})$ is negligible in the security parameter, given that $\mathsf{PKE}_{\mathsf{Cor}}$ is verifiable IND-CPA secure correlated public-key encryption scheme.

Proof: We proceed as in the proof of Lemma 3. Assume that $\mathsf{Adv}_{\mathsf{Game2}}(\mathcal{A}) \geq \epsilon$ for some non-negligible ϵ. We will now construct an IND-CPA adversary $\mathcal{B}$ against $\mathsf{PKE}_{\mathsf{Cor}}$ that breaks the IND-CPA security of $\mathsf{PKE}_{\mathsf{Cor}}$ with advantage ϵ. Again, instead of generating pk like game 2, $\mathcal{B}$ will construct pk using the public-key pk' provided by the IND-CPA experiment. Let $\mathsf{d} = \mathsf{ECC}(\mathsf{vk}^*)$. $\mathcal{B}$ sets $\mathsf{pk}^{\mathsf{d}} = \mathsf{pk}^*$. All remaining components pk_i^j of pk are generated by $(\mathsf{pk}_i^j, \mathsf{sk}_i^j) \leftarrow \mathsf{Gen}(1^n)$, for which $\mathcal{B}$ stores the corresponding sk_i^j. Obviously, the pk generated by $\mathcal{B}$ is identically distributed to the pk generated by game 2, as in both cases all components are pk_i^j are generated independently by the key-generation algorithm Gen of PKE.

Whenever $\mathcal{A}$ sends a decryption query with $\mathsf{vk} \neq \mathsf{vk}^*$, $\mathcal{B}$ does the following. Let $\mathsf{d} = \mathsf{ECC}(\mathsf{vk})$ and $\mathsf{d}^* = \mathsf{ECC}(\mathsf{vk}^*)$. Since the two codewords d and d^* are distinct and the code ECC has minimum-distance τ, there exist a τ-set of indices $T \subseteq \{1, \ldots, k\}$ such that it holds for all $t \in T$ that $\mathsf{d}_t \neq \mathsf{d}_t^*$. Thus, the public-keys $\mathsf{pk}_t^{\mathsf{d}_t}$, for $t \in T$ were generated by $\mathcal{B}$ and it thus knows the corresponding secret-keys $\mathsf{sk}_t^{\mathsf{d}_t}$. $\mathcal{B}$ checks if $\mathsf{Verify}(\mathsf{c}', \mathsf{pk}^{\mathsf{d}}, T, \mathsf{sk}_T^{\mathsf{d}}) = 1$ holds, i.e. if c' is a valid ciphertext for $\mathsf{PKE}_{\mathsf{Cor}}$ under the public-key pk^{d}. If so, $\mathcal{B}$ decrypts $\tilde{\mathsf{m}}_T = (\tilde{\mathsf{m}}_t | t \in T) = (\mathsf{Dec}(\mathsf{sk}_t^{\mathsf{d}_t}, \mathsf{c}'_t) | t \in T)$. Since the plaintext-space $\mathcal{M}_{\mathsf{Cor}}$ is τ-correlated, $\mathcal{B}$ can efficiently recover the whole message $\tilde{\mathsf{m}}$ from the τ-submessage $\tilde{\mathsf{m}}_T$. Finally, $\mathcal{B}$ decodes $\mathsf{m} = \mathsf{D}(\tilde{\mathsf{m}})$ to recover the message m and outputs m to $\mathcal{A}$. Observe that the verifiability-property of $\mathsf{PKE}_{\mathsf{Cor}}$ holds regardless of the subset T used to verify. Thus, from $\mathcal{A}$'s view the decryption-oracle behaves identically in game 2 and in $\mathcal{B}$'s simulation.

Finally, when $\mathcal{A}$ sends its challenge messages m_0 and m_1, $\mathcal{B}$ forwards m_0 and m_1 to the IND-CPA experiment for $\mathsf{PKE}_{\mathsf{Cor}}$ and receives a challenge-ciphertext $\mathsf{c}^{*\prime}$. $\mathcal{B}$ then computes $\sigma = \mathsf{Sign}(\mathsf{sk}^*, \mathsf{c}^{*\prime})$ and outputs the challenge-ciphertext $\mathsf{c}' = (\mathsf{c}^{*\prime}, \mathsf{vk}^*, \sigma)$ to $\mathcal{A}$. When $\mathcal{A}$ generates an output, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

Putting it all together, $\mathcal{A}$'S views are identically distributed in game 2 and $\mathcal{B}$'s simulation. Therefore, it holds that $\mathsf{Adv}_{\mathsf{IND-CPA}}(\mathcal{B}) = \mathsf{Adv}_{\mathsf{game2}}(\mathcal{A}) \geq \epsilon$. Thus, $\mathcal{B}$ breaks the IND-CPA security of $\mathsf{PKE}_{\mathsf{Cor}}$ with non-negligible advantage ϵ, contradicting the assumption. ■

Plugging Lemma 5 and Lemma 6 establish that any PPT IND-CCA2 adversary $\mathcal{A}$ has at most negligible advantage in winning the IND-CCA2 experiment for the scheme $\mathsf{PKE}'_{\mathsf{cca2}}$. ■

C. Verifiable Correlated PKE based on the McEliece Scheme

We can use a modified version of the scheme presented in Section IV to instantiate a τ-correlated verifiable IND-CPA secure McEliece scheme $\mathsf{PKE}_{McE,Cor}$. A corresponding IND-CCA2 secure scheme is immediately implied by the construction in Section V-B. As plaintext-space $\mathcal{M}_{\mathsf{Cor}}$ for $\mathsf{PKE}_{McE,Cor}$, we choose the set of all tuples $(\mathsf{s}|\mathsf{y}_1, \ldots, \mathsf{s}|\mathsf{y}_k)$, where s is a n-bit string and $(\mathsf{y}_1, \ldots, \mathsf{y}_k)$ is a codeword from code C that can efficiently correct $k - \tau$ erasures. Clearly, $\mathcal{M}_{\mathsf{Cor}}$ is τ-correlated. Let E_{C} be the encoding-function of C and D_{C} the decoding-function of C. The randomized encoding-function $\mathsf{E}_{\mathsf{McE,Cor}}$ used by $\mathsf{PKE}_{\mathsf{McE,Cor}}$ proceeds as follows. Given a message m and random coins s, it first computes $(\mathsf{y}_1, \ldots, \mathsf{y}_k) = \mathsf{E}_{\mathsf{C}}(\mathsf{m})$ and outputs $(\mathsf{s}|\mathsf{y}_1, \ldots, \mathsf{s}|\mathsf{y}_k)$. The decoding-function $\mathsf{D}_{\mathsf{McE,Cor}}$ takes a tuple $(\mathsf{s}|\mathsf{y}_1, \ldots, \mathsf{s}|\mathsf{y}_k)$ and outputs $\mathsf{D}_{\mathsf{C}}(\mathsf{y}_1, \ldots, \mathsf{y}_k)$. Like in the scheme of Section IV, the underlying OW-CPA secure encryption-scheme PKE is textbook-McEliece.

The τ-correlatedness of $\mathsf{PKE}_{\mathsf{McE,Cor}}$ follows directly by the construction of $\mathcal{M}_{\mathsf{Cor}}$, $\mathsf{E}_{\mathsf{Mce,Cor}}$ and $\mathsf{D}_{\mathsf{Mce,Cor}}$. It remains to show verifiability and IND-CPA security of the scheme. The $\mathsf{Verify}_{\mathsf{McE}}$-algorithm takes a ciphertext $\mathsf{c} = (\mathsf{c}_1, \ldots, \mathsf{c}_k)$, a public-key pk, an a partial secret-key sk_T (for a τ-sized index-set T) and proceeds as follows. First, it decrypts the components of c at the indices of T, i.e. it computes $\mathsf{x}_t = \mathsf{Dec}_{\mathsf{McE}}(\mathsf{sk}_t, \mathsf{c}_t)$ for $t \in T$. Then, it checks whether all x_t are of the form $\mathsf{x}_t = \mathsf{s}|\mathsf{y}_t$ for the same string s. If not, it stops and outputs 0. Next, it constructs a vector $\tilde{\mathsf{y}} \in \Sigma^k$ with $\tilde{\mathsf{y}}_i = \mathsf{y}_i$ for $i \in T$ and $\tilde{\mathsf{y}}_i = \perp$ (erasure) for $i \notin T$. Verify then runs the erasure-correction algorithm of C on $\tilde{\mathsf{y}}$. If the erasure-correction fails, it stops and outputs 0. Otherwise let $\mathsf{y} = (\mathsf{y}_1, \ldots, \mathsf{y}_k)$ be the corrected vector returned by the erasure-correction. Then, Verify sets $\mathsf{x} = (\mathsf{s}|\mathsf{y}_1, \ldots, \mathsf{s}|\mathsf{y}_k)$. Let $\mathbf{G}_1, \ldots, \mathbf{G}_k$ be the generator-matrices given in $\mathsf{pk}_1, \ldots, \mathsf{pk}_k$. Finally, Verify checks whether all the vectors $\mathsf{c}_j \oplus \mathsf{x}\mathbf{G}_j$, for $j = 1, \ldots, k$, have Hamming-weight smaller than t. If so, it outputs 1, otherwise 0. Clearly, if $\mathsf{Verify}_{\mathsf{McE}}$ outputs 1, then the ciphertext-components c_j of c are valid McEliece encryptions.

The IND-CPA-security is proven analogously to Lemma 4. First, the McEliece generator-matrices $\mathbf{G}_i$ are replaced by random matrices $\mathbf{R}_i$, then, using the LPNDP-assumption, vectors of the form $\mathsf{s}\mathsf{R}_i \oplus \mathsf{e}_i$ are replaced by uniformly random vectors u_i. Likewise, after this transformation the adversarial advantage is 0.

VI. Acknowledgments

We would like to thank Edoardo Persichetti for comments on the definition of k-repetition PKE in an earlier version of this work. We would like to thank the anonymous referees who provided us with valuable feedback that greatly improved the quality of this paper.

References

[1] A. Becker, A. Joux, A. May, A. Meurer. Decoding Random Binary Linear Codes in $2^{n/20}$: How $1 + 1 = 0$ Improves Information Set Decoding. *EUROCRYPT 2012*. pp. 520–536.

[2] 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.

[3] D. J. Bernstein,T. Lange, C. Peters. Attacking and Defending the McEliece Cryptosystem. *PQCrypto 2008*. pp. 31–46.

[4] D. J. Bernstein, T. Lange, C. Peters. Smaller Decoding Exponents: Ball-Collision Decoding. *CRYPTO 2011*. pp. 743-760. 2011.

[5] B. Biswas, N. Sendrier. McEliece Cryptosystem Implementation: Theory and Practice. *PQCrypto*. pp. 47-62. 2008.

[6] R. Canetti, S. Halevi, J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. *EUROCRYPT 2004*. pp. 207–222.

[7] 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.

[8] N. Courtois, M. Finiasz, N. Sendrier. How to Achieve a McEliece Digital Signature Scheme. *ASIACRYPT 2001*. pp. 157–174.

[9] R. Cramer, V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. *CRYPTO 1998*. pp. 13–25.

[10] D. Dolev, C. Dwork, M. Naor. Non-malleable Cryptography. *SIAM J. Comput*. Vol 30(2), pp. 391–437. 2000.

[11] J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, J.-P. Tillich. A Distinguisher for High Rate McEliece Cryptosystems. *Information Theory Workshop (ITW), 2011 IEEE*. pp. 282-286, 2011

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

[13] M. Finiasz and N. Sendrier. Security Bounds for the Design of Code-based Cryptosystems. *Asiacrypt 2009*, LNCS 5912, pp. 88–105.

[14] S. Goldwasser, S. Micali. Probabilistic Encryption. *J. Comput. Syst. Sci.* Vol 28(2), pp. 270–299. 1984.

[15] S. Goldwasser, V. Vaikuntanathan. Correlation-secure Trapdoor Functions from Lattices. Manuscript, 2008.

[16] D. Hofheinz, E. Kiltz. Secure Hybrid Encryption from Weakened Key Encapsulation. *CRYPTO 2007*. pp. 553–571.

[17] J. Katz, J. S. Shin: Parallel and Concurrent Security of the HB and HB+ Protocols. *EUROCRYPT 2006*. pp. 73–87.

[18] K. Kobara and H. Imai. Semantically Secure McEliece Public-Key Cryptosystems Conversions for McEliece PKC, LNCS 1992, Springer, 2001.

[19] L. Lamport. Constructing Digital Signatures from One-Way Functions, *SRI intl. CSL-98*. Oct. 1979.

[20] P. J. Lee and E. F. Brickell. An Observation on the Security of McElieces Public-key Cryptosystem. *EUROCRYPT 1988*, pages 275280, 1988.

[21] J. S. Leon. A Probabilistic Algorithm for Computing Minimum Weights of Large Error-correcting Codes. *IEEE Transactions on Information Theory*, 34(5):1354 1359, 1988.

[22] Y. Lindell. A Simpler Construction of CCA2-Secure Public-Key Encryption under General Assumptions. *EUROCRYPT 2003*. pp. 241–254.

[23] P. Loidreau, N. Sendrier. Weak keys in McEliece Public-key Cryptosystem. *IEEE Transactions on Information Theory*. pp. 1207–1212. 2001.

[24] R.J. McEliece: A Public-Key Cryptosystem Based on Algebraic Coding Theory. *Deep Space Network Progress Report*. 1978.

[25] A. May, A. Meurer, E. Thomae. Decoding Random Linear Codes in $\tilde{\mathcal{O}}(2^{0.054n})$. *ASIACRYPT 2011*. pp. 107–124.

[26] M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Applications. *21st STOC*. pp. 33–43. 1989.

[27] R. Nojima, H. Imai, K. Kobara, K. Morozov, Semantic Security for the McEliece Cryptosystem without Random Oracles. *International Workshop on Coding and Cryptography (WCC) 2007*. pp. 257–268. Journal version in *Designs, Codes and Cryptography*. Vol. 49, No. 1-3, pp. 289–305. 2008.

[28] C. Peikert, B. Waters. Lossy Trapdoor Functions and Their Applications. *STOC 2008*. pp. 187–196.

[29] E. Persichetti, Personal Communication.

[30] C. Rackoff, D. R. Simon: Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. *CRYPTO 1991*. pp. 433–444.

[31] O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *STOC 2005*. pp. 84–93.

[32] A. Rosen, G. Segev. Chosen-Ciphertext Security via Correlated Products. *TCC 2009*. pp. 419–436.

[33] A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen- Ciphertext Security. In *40th FOCS*. pp. 543–553.

[34] N. Sendrier. Finding the Permutation Between Equivalent Linear Codes: The Support Splitting Algorithm. *IEEE Trans. Inf. Theory*. Vol. 46(4), pp.1193–1203. 2000.

[35] N. Sendrier. On the Use of Structured Codes in Code Based Cryptography. *Coding Theory and Cryptography III, The Royal Flemish Academy of Belgium for Science and the Arts*. 2010.

[36] J. Stern. A Method for Finding Codewords of Small Weight. *3rd International Colloquium on Coding Theory and Applications*, pp. 106–113, 1989.

[37] F. Strenzke, E. Tews, H. G. Molter, R. Overbeck, A. Shoufan. Side Channels in the McEliece PKC. *PQCrypto 2008*, pp. 216-229.