On the CCA2 Security of McEliece in the Standard Model

Edoardo Persichetti

Florida Atlantic University

Abstract. In this paper we study public-key encryption schemes based on error-correcting codes that are IND-CCA2 secure in the standard model. In particular, we analyze a protocol due to Dowsley, Müller-Quade and Nascimento, based on a work of Rosen and Segev. The original formulation of the protocol contained some ambiguities and incongruences, which we point out and correct; moreover, the protocol deviates substantially from the work it is based on. We then present a construction which resembles more closely the original Rosen-Segev framework, and show how this can be instantiated with the McEliece scheme.

1 Introduction

The McEliece cryptosystem [11] is the first scheme based on coding theory problems and it makes use of error-correcting codes (binary Goppa codes in the original proposal). Persichetti [15] has shown that it is possible to produce a very efficient CCA2-secure scheme in the random oracle model; it is however of interest to study systems that are secure in the standard model.

Rosen and Segev in [16] gave a general approach for CCA2 security in the standard model incorporating tools like lossy trapdoor functions and one-time signature schemes. This general protocol can be applied directly to many different hard problems such as Quadratic Residuosity, Composite Residuosity, the d-linear Assumption and the Syndrome Decoding Problem, as shown in [6]. Dowsley et al. [3] have attempted to adopt the Rosen-Segev approach to the McEliece framework. To do this, a new structure called k-repetition PKE is introduced, as well as a number of differences in the key generation, encryption and decryption processes. It is claimed that the scheme has IND-CCA2 security in the standard model, but some ambiguities in the constructions were present which undermined this claim. These have been addressed in subsequent works: in a follow-up paper [2], the authors, with the addition of Döttling, present a corrected version of the scheme of [3]. The paper was published in 2012, around the same time an earlier version of this work [14] was released. It is therefore safe to assume the results were obtained independently.

Mathew et al. [10] introduced an alternative construction for code-based IND-CCA2 secure PKE in the standard model, which is more efficient than the proposals studied in this work. However, their construction is based on the Niederreiter

scheme [12]. Finally, in an independent work [19], Yoshida, Morozov and Tanaka proved that it is possible to obtain Key Privacy for both the Rosen-Segev scheme and the Dowsley et al. scheme. This is an alternative security notion that aims at guaranteeing the non-malleability of public keys, rather than ciphertexts. For this reason, it is also known as Anonymity or Indistinguishability of Keys (IK). Note that this notion was proved to hold for code-based schemes in the random oracle model, again in [15].

In this paper we analyze in detail the construction of [3], since we believe it introduced an interesting alternative to the Rosen-Segev approach. First of all, we make some observations, point out the ambiguities of the description of the scheme, and discuss the fixes of [2,14]. For the sake of completeness, we provide a correct formulation together with a proof of security. Finally, we show how to get a CCA2-secure encryption scheme based on the McEliece assumptions using the original Rosen-Segev approach.

2 Preliminaries

We will summarize here all the objects we are going to work with in the paper.

Formally, we define a Public-Key Encryption scheme (PKE) to be formed by the 6-tuple $(\mathsf{K}, \mathsf{P}, \mathsf{C}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, defined as follows:

- K: The pair $(\mathsf{K}_{\mathsf{publ}}, \mathsf{K}_{\mathsf{priv}})$, respectively the public key and private key spaces.
- P: The set of messages to be encrypted, or *plaintext space*.
- C: The set of the messages transmitted over the channel, or *ciphertext space*.
- KeyGen: A probabilistic key generation algorithm that takes as input a security parameter 1^δ and outputs a public key $\mathsf{pk} \in \mathsf{K}_{\mathsf{publ}}$ and a private key $\mathsf{sk} \in \mathsf{K}_{\mathsf{priv}}$.
- Enc: A (possibly probabilistic) encryption algorithm that receives as input a public key $\mathsf{pk} \in \mathsf{K}_{\mathsf{publ}}$ and a plaintext $\phi \in \mathsf{P}$ and returns a ciphertext $\psi \in \mathsf{C}$.
- Dec: A deterministic decryption algorithm that receives as input a private key $\mathsf{sk} \in \mathsf{K}_{\mathsf{priv}}$ and a ciphertext $\psi \in \mathsf{C}$ and outputs a plaintext $\phi \in \mathsf{P}$ or the failure symbol $\perp$.

Similarly, we define a Signature scheme (SS) as a 6-tuple $(\mathsf{K}, \mathsf{M}, \Sigma, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver})$, defined as follows:

– K: The pair $(\mathsf{K_{sign}}, \mathsf{K_{ver}})$, respectively the signing key and verification key spaces.

– M: The set of documents to be signed, or *message space*.

– Σ: The set of the signatures to be transmitted with the messages, or *signature space*.

– KeyGen: A probabilistic key generation algorithm that takes as input a security parameter 1^δ and outputs a signing key $\mathsf{sgk} \in \mathsf{K_{sign}}$ and a verification key $\mathsf{vk} \in \mathsf{K_{ver}}$.

– Sign: A (possibly probabilistic) signing algorithm that receives as input a signing key $\mathsf{sgk} \in \mathsf{K_{sign}}$ and a message $\mu \in \mathsf{M}$ and returns a signature $\sigma \in \Sigma$.

– Ver: A deterministic decryption algorithm that receives as input a verification key $\mathsf{vk} \in \mathsf{K_{ver}}$, a message $\mu \in \mathsf{M}$ and a signature $\sigma \in \Sigma$ and outputs 1, if the signature is recognized as valid, or 0 otherwise.

2.1 Security notions

Here we refresh the security notions which will be addressed in this work.

Definition 1 (IND). *An adversary $\mathcal{A}$ for the indistinguishability (IND) property is a two-stage polynomial-time algorithm. In the first stage, $\mathcal{A}$ takes as input a public key $\mathsf{pk} \in \mathsf{K_{publ}}$, then outputs two arbitrary plaintexts ϕ_0, ϕ_1. In the second stage, it receives a ciphertext $\psi^* = \mathsf{Enc_{pk}}(\phi_b)$, for $b \in \{0, 1\}$, and returns a bit b^*. The adversary succeeds if $b^* = b$. More precisely, we define the* advantage *of $\mathcal{A}$ against PKE as*

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \mathsf{Pr}[b^* = b] - \frac{1}{2}. \tag{1}$$

Indistinguishability can be achieved in various attack models. In the strongest model (that of interest to us), called CCA2, the adversary is allowed to make use of a decryption oracle during the game, with the only exception that it is not allowed to ask for the decryption of the challenge ciphertext.

Definition 2 (IND-CCA2). *The attack game for IND-CCA2 (or* active attack*) proceeds as follows:*

- *Query a key generation oracle to obtain a public key pk.*
- *Make a sequence of calls to a decryption oracle, submitting any string ψ of the proper length (not necessarily an element of C). The oracle will respond with $\mathsf{Dec_{sk}}(\psi)$.*

- *Choose* $\phi_0, \phi_1 \in \mathsf{P}$ *and submit them to an encryption oracle. The oracle will choose a random* $b \in \{0,1\}$ *and reply with the "challenge" ciphertext* $\psi^* = \mathsf{Enc}_{\mathsf{pk}}(\phi_b)$.
- *Keep performing decryption queries. If the submitted ciphertext is* $\psi = \psi^*$, *return* $\perp$.
- *Output* $b^* \in \{0,1\}$.

We say that a PKE has Indistinguishability against Adaptive Chosen Ciphertext Attacks (IND-CCA2) *if the advantage* $\mathsf{Adv}_{\mathsf{CCA2}}$ *of any IND adversary* $\mathcal{A}$ *in the CCA2 attack model is negligible.*

There are many notions of security for signature schemes; the one we present here is what we need for the Rosen-Segev scheme.

Definition 3 (One-Time Strong Unforgeability). *We define an adversary* $\mathcal{A}$ *as a polynomial-time algorithm that acts as follows:*

- *Query a key generation oracle to obtain a verification key* vk.
- *Choose a message* $\mu \in \mathsf{M}$ *and submit it to a signing oracle. The oracle will reply with* $\sigma = \mathsf{Sign}_{\mathsf{sgk}}(\mu)$.
- *Output a pair* (μ^*, σ^*).

The adversary succeeds if $\mathsf{Ver}_{\mathsf{vk}}(\mu^*, \sigma^*) = 1$ *and* $(\mu^*, \sigma^*) \neq (\mu, \sigma)$. *We say that a signature scheme is* One-Time Strongly Unforgeable *if the probability of success of any adversary* $\mathcal{A}$ *is negligible in the security parameter, i.e.*

$$\mathsf{Pr}[\mathsf{vk} \leftarrow \mathsf{K}_{\mathsf{ver}} : \mathsf{Ver}_{\mathsf{vk}}(\mathcal{A}(\mathsf{vk}, \mathsf{Sign}_{\mathsf{sgk}}(\mu))) = 1] \in \mathit{negl}(\lambda). \tag{2}$$

Note that in this scenario the adversary is only allowed to ask for the signature of a **single** message (hence the One-Time), so this is a relatively weak security assumption.

Definition 4 (Hard-Core Predicate). *Let* f *be a one-way function and* h *be a predicate, i.e. a function whose output is a single bit. Define an adversary* $\mathcal{A}$ *to be a probabilistic polynomial-time algorithm that, on input* $f(x)$, *tries to compute* $h(x)$, *i.e.* $\mathcal{A}(f(x)) = b \in \{0,1\}$. *The predicate* h *is a* Hard-Core Predicate *of the function* f *if the probability* $\mathsf{Pr}[b = h(x)] - \frac{1}{2}$ *is negligible for all random choices of* x.

2.2 The McEliece cryptosystem

The McEliece cryptosystem, based on coding theory, was introduced in 1978 by Robert J. McEliece [11] and, for an appropriate choice of parameters, it is still unbroken. In the original proposal, binary Goppa codes are used as a basis for

the construction. We give here a more general and modern description extending the scheme to generic finite fields $\mathbb{F}_q$ and introducing a few little optimizations. The input parameters are the code length n, the code dimension k and the error-correction capacity w.

- Setup: Choose a code family and fix public parameters n, k, w.
- $\mathsf{K_{publ}}$: The set of $k \times n$ matrices over $\mathbb{F}_q$.
- $\mathsf{K_{priv}}$: The set[1] of "code descriptions" for the chosen code family.
- P: The vector space $\mathbb{F}_q^k$.
- C: The vector space $\mathbb{F}_q^n$.
- KeyGen: Sample a random generator matrix G for a code of the chosen family. Compute the "scrambled" generator matrix $\hat{G}$, then publish the public key $\hat{G} \in \mathsf{K_{publ}}$ and store the private key $\Gamma \in \mathsf{K_{priv}}$.
- Enc: On input a public key $\hat{G} \in \mathsf{K_{publ}}$ and a plaintext $m \in \mathsf{P}$, sample a random error vector e of weight w in $\mathbb{F}_q^n$ and return the ciphertext $\psi = m\hat{G} + e \in \mathsf{C}$.
- Dec: On input the private key $\Gamma \in \mathsf{K_{priv}}$ and a ciphertext $\psi \in \mathsf{C}$, apply the decoding algorithm D_Γ to it. If the decoding succeeds, return the resulting plaintext $\phi = m$. Otherwise, output $\perp$.

Remark 1. In the original McEliece proposal the scrambling process was accomplished using an invertible matrix S and a permutation matrix P, and $\hat{G}$ was obtained as SGP. This is rather outdated and unpractical; moreover, it can introduce vulnerabilities to the scheme as per the work of Strenzke et al. (for example [17,18]). A still secure (Biswas and Sendrier, [1]), but much simpler description would be to take the public key $\hat{G}$ to be just the systematic form of G.

The security of the McEliece scheme relies on two computational assumptions.

Assumption 1 (Indistinguishability) *The matrix $\hat{G}$ output by KeyGen is computationally indistinguishable from a uniformly chosen matrix of the same size.*

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

It is immediately clear that the following corollary is true.

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

[1] For instance for Goppa codes, this is given by the support $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^m}$ and the Goppa polynomial g.

Remark 2. In a recent paper [4], Faugère et al. presented a distinguisher for instances of the McEliece cryptosystem that make use of high-rate Goppa codes. While the distinguisher works only in a special case and doesn't affect security for the general scheme, it is still recommended to avoid such insecure choices.

As we mentioned in the introduction, it is possible to easily obtain CCA2 security for the McEliece cryptosystem in the Random Oracle Model using either standard conversions (as in [7,8]) or the dedicated paradigm of [15]. We therefore consider only the issue of achieving such a security level in the Standard Model.

2.3 Computable functions and correlated products

We define here the notion of security under correlated products for a collection of functions. Formally, we describe a collection of *efficiently computable functions* as a pair of algorithms $\mathcal{F} = (\mathsf{G},\mathsf{F})$ where G is a generation algorithm that samples the description f of a function and $\mathsf{F}(f,x)$ is an evaluation algorithm that evaluates the function f on a given input x. We then define a k-wise product as follows:

Definition 5. *Let $\mathcal{F} = (\mathsf{G},\mathsf{F})$ be a collection of efficiently computable functions and k be an integer. The* k-wise product *$\mathcal{F}_\mathsf{k}$ is a pair of algorithms $(\mathsf{G}_\mathsf{k}, \mathsf{F}_\mathsf{k})$ such that:*

- G_k is a generation algorithm that independently samples k functions from $\mathcal{F}$ by invoking k times the algorithm G and returns a tuple $(f_1, \ldots, f_\mathsf{k})$.
- F_k is an evaluation algorithm that receives as input a sequence of functions $(f_1, \ldots, f_\mathsf{k})$ and a sequence of points $(x_1, \ldots, x_\mathsf{k})$ and invokes F to evaluate each function on the corresponding point, i.e.

$$\mathsf{F}_\mathsf{k}(f_1, \ldots, f_\mathsf{k}, x_1, \ldots, x_\mathsf{k}) = (\mathsf{F}(f_1, x_1), \ldots, \mathsf{F}(f_\mathsf{k}, x_\mathsf{k})).$$

A trapdoor one-way function is then an efficiently computable function that, given the image of a uniform chosen input, is easy to invert with the use of a certain trapdoor td but hard to invert otherwise; i.e. there exists an algorithm F^{-1} such that $\mathsf{F}^{-1}(\mathsf{td}, \mathsf{F}(f,x)) = x$.
We may think to extend the notion to the case where the input is given according to a certain distribution, that is, there exists a correlation between the points $x_1, \ldots, x_\mathsf{k}$.

Definition 6. *Let $\mathcal{F} = (\mathsf{G},\mathsf{F})$ be a collection of efficiently computable functions with domain D and $\mathcal{C}_\mathsf{k}$ be a distribution of points in $D_1 \times \cdots \times D_\mathsf{k}$. We say that $\mathcal{F}$ is* secure under a $\mathcal{C}_\mathsf{k}$-correlated product *if $\mathcal{F}_\mathsf{k}$ is one-way with respect to the input distribution $\mathcal{C}_\mathsf{k}$.*

In the special case where the input distribution $\mathcal{C}_\mathsf{k}$ is exactly the uniform k-repetition distribution (that is, k copies of the same input $x \in D$) we simply speak about *one-wayness under k-correlated inputs*. Rosen and Segev in [16] showed that a collection of lossy trapdoor functions for an appropriate choice of parameters can be used to construct a collection of functions that is one-way under k-correlated inputs. Their work is summarized in the next section.

3 The Rosen-Segev scheme

The computational assumption underlying the scheme is that there exists a collection of functions $\mathcal{F} = (\mathsf{G},\mathsf{F})$ which is secure under k-correlated inputs. The scheme makes use of a strongly-unforgeable signature scheme and of a hard-core predicate h for the collection $\mathcal{F}_{\mathsf{k}}$.

$\mathsf{KeyGen}^{\mathrm{RS}}$: Invoke G for $2\mathsf{k}$ times independently and obtain the descriptions of functions $(f_1^0, f_1^1, \ldots, f_{\mathsf{k}}^0, f_{\mathsf{k}}^1)$ and the corresponding trapdoors $(\mathsf{td}_1^0, \mathsf{td}_1^1, \ldots, \mathsf{td}_{\mathsf{k}}^0, \mathsf{td}_{\mathsf{k}}^1)$. The former is distributed as the public key pk, while the latter is the private key sk.

$\mathsf{Enc}^{\mathrm{RS}}$: To encrypt a plaintext $m \in \{0,1\}$ with the public key pk, sample a key from a strongly-unforgeable one-time signature scheme, say $(\mathsf{vk}, \mathsf{sgk})$ and a random $x \in \{0,1\}^N$. Write vk_i for the i-th bit of vk and let h be a hard-core predicate, then:

- $c_i = \mathsf{F}(f_i^{\mathsf{vk}_i}, x)$ for $i = 1, \ldots, \mathsf{k}$.
- $y = m \oplus h(f_1^{\mathsf{vk}_1}, \ldots, f_{\mathsf{k}}^{\mathsf{vk}_{\mathsf{k}}}, x)$.
- $\sigma = \mathsf{Sign}_{\mathsf{sgk}}^{\mathrm{SS}}(c_1, \ldots, c_{\mathsf{k}}, y)$.

It is assumed that $\mathsf{vk} \in \{0,1\}^{\mathsf{k}}$: if not, it is enough to apply a universal one-way hash function to obtain the desired length.
Finally, output the ciphertext $\psi = (\mathsf{vk}, c_1, \ldots, c_{\mathsf{k}}, y, \sigma)$.

$\mathsf{Dec}^{\mathrm{RS}}$: Upon receipt of a ciphertext ψ:

- Verify the signature; if $\mathsf{Ver}_{\mathsf{vk}}^{\mathrm{SS}}((c_1, \ldots, c_{\mathsf{k}}, y), \sigma) = 0$ output $\perp$.
- Otherwise compute $x_i = \mathsf{F}^{-1}(\mathsf{td}_i^{\mathsf{vk}_i}, c_i)$ for $i = 1, \ldots, \mathsf{k}$.
- If $x_1 = \cdots = x_{\mathsf{k}}$ then set $m = y \oplus h(f_1^{\mathsf{vk}_1}, \ldots, f_{\mathsf{k}}^{\mathsf{vk}_{\mathsf{k}}}, x_1)$ and return the plaintext m, otherwise output $\perp$.

The security of the scheme is summarized in the next theorem, which was proved in [16].

Theorem 1. *Assuming that $\mathcal{F}$ is secure under k-correlated inputs, and that the signature scheme is one-time strongly unforgeable, the above encryption scheme is IND-CCA2-secure.*

The proof consists of a standard argument, divided in two parts. The first part shows that if an adversary exists capable to break the CCA2 security of the scheme, it can be converted to an adversary able to forge the signature scheme. In the second part, assuming that the forgery doesn't occur, an adversary is built that contradicts the security of the hard-core predicate. Due to space constraints, we don't present the proof here, but we refer the reader to [16] for more details.

4 Previous proposals

It would be natural to describe the McEliece encryption process as a function $f_G(x, y) = xG + y$. However, this function is clearly not secure under correlated inputs. Let us assume $\mathbb{F}_q$ has characteristic 2 like in the original McEliece scheme. Then, given two evaluations $f_{G_1}(x, y) = xG_1 + y$ and $f_{G_2}(x, y) = xG_2 + y$, an attacker could simply sum the outputs together and, since the error vector cancels out, obtain $x(G_1 + G_2)$, from which it is easy to recover x. The problem is that, since we are defining a function, there is no randomness anymore, whereas McEliece requires a random error vector in order to be secure under k-correlated inputs. A mapping that incorporates a random element would in fact give a different result for multiple encryptions of the same plaintext and so would not have a unique image.

We now present two schemes that have been proposed to deal with the matter.

4.1 Syndrome decoding

This construction was presented in [6] and is based on the Niederreiter cryptosystem [12]. Since this relies on the properties of the parity-check matrix rather than the generator matrix, it is often considered the "dual" cryptosystem and the computational assumptions for the security change accordingly.
The Niederreiter trapdoor function can be described as the family $\mathcal{N} = (\mathsf{G},\mathsf{F})$ in the following way:

Generation: on input n, k the algorithm G generates a random parity-check matrix H for an $[n, k]$-linear code with an efficient decoding algorithm over $\mathbb{F}_q$, then computes its systematic form $\hat{H}$. The algorithm returns the public key $\hat{H}$ and the private key Γ.

Evaluation: on input $\hat{H}, e$, where e is a string of fixed weight w in $\mathbb{F}_q^n$, the algorithm F computes $\psi = \hat{H}e$ and returns the ciphertext ψ.

It is possible to invert F using the trapdoor: on input Γ and ψ, simply decode to obtain e using the decoding algorithm connected to Γ. The function is proved to be one-way under k-correlated inputs in [6, Th. 6.2] if k is chosen such that the Niederreiter assumptions hold for n and $(n - k)\mathsf{k}$, and it is intended to be used in the general Rosen-Segev framework.

4.2 k-repetition PKE

Dowsley, Müller-Quade and Nascimento [3] propose a scheme that resembles the Rosen-Segev protocol trying to apply it to the McEliece cryptosystem. Despite the authors' claim that this is the "direct translation" of [16], this is not exactly the case.

Among other differences, the main discrepancy is that the scheme doesn't rely on a collection of functions but instead defines a structure called *k-repetition Public-Key Encryption* (PKE_{k}). This is essentially an application of k samples of the PKE to the same input, in which the decryption algorithm also includes a verification step on the k outputs. The encryption step produces a signature directly on the McEliece ciphertexts instead of introducing a random vector x as in the original scheme. This means that it is necessary to use an IND-CPA secure variant of McEliece's cryptosystem to achieve CCA2 security. For this task, the authors propose to use the "Randomized McEliece" variant by Nojima et al. [13]. This variant uses, as the name says, additional randomness, in the form of a random string. The string is sampled from a randomness set R with elements of length k_2, and then concatenated to the plaintext so that the resulting string has length k and can be encoded as normal. We briefly recall the scheme below.

- Setup: Fix public system parameters $q, n, k, w \in \mathbb{N}$ such that $k = k_1 + k_2$.
- $\mathsf{K_{publ}}$: The set of $k \times n$ matrices over $\mathbb{F}_q$.
- $\mathsf{K_{priv}}$: The set of "code descriptions" for the chosen code family.
- P: The vector space $\mathbb{F}_q^{k_1}$.
- R: The vector space $\mathbb{F}_q^{k_2}$.
- C: The vector space $\mathbb{F}_q^n$.
- KeyGen: Sample a random generator matrix G for a code of the chosen family. Compute the "scrambled" generator matrix $\hat{G}$, then publish the public key $\hat{G} \in \mathsf{K_{publ}}$ and store the private key $\Gamma \in \mathsf{K_{priv}}$.
- Enc: On input a public key $\hat{G} \in \mathsf{K_{publ}}$, a plaintext $m \in \mathsf{P}$ and a random string $r \in \mathsf{R}$, sample a random error vector e of weight w in $\mathbb{F}_q^n$ and return the ciphertext $\psi = (r|m)\hat{G} + e \in \mathsf{C}$.
- Dec: On input the private key $\Gamma \in \mathsf{K_{priv}}$ and a ciphertext $\psi \in \mathsf{C}$, apply the decoding algorithm D_Γ to it. If the decoding succeeds, parse the result as $(r|m)$ and return the plaintext $\phi = m$. Otherwise, output $\perp$.

Remark 3. It is clear that, as already mentioned by the authors in [13], the IND-CPA security of the randomized McEliece scheme is not absolute, but depends on the choice of the sizes of the message m and randomness r in the encryption procedure $(r|m)\hat{G}+e$. In the context of a CPA attack game, in fact, this ciphertext is subject to general decoding attacks with partial information about the plaintext. As illustrated

in [13, Table 1], if the randomness r is not large enough, the IND-CPA security of the scheme can be easily broken.

We now present the scheme described in [3]. Note that, in the paper, this is presented as a general scheme, applicable to any IND-CPA secure PKE which is secure and verifiable under k-correlated inputs.

$\mathsf{KeyGen}^{\mathrm{DMQN}}$: Invoke $\mathsf{KeyGen}^{\mathrm{PKE}}$ for $2\mathsf{k}$ times independently and obtain the collection of public keys $(\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_\mathsf{k}^0, \mathsf{pk}_\mathsf{k}^1)$ and the corresponding private keys $(\mathsf{sk}_1^0, \mathsf{sk}_1^1, \ldots, \mathsf{sk}_\mathsf{k}^0, \mathsf{sk}_\mathsf{k}^1)$, then run the key generation algorithm for the signature scheme to obtain a key $(\mathsf{vk}^*, \mathsf{sgk}^*)$. Publish the public key $\mathsf{pk} = (\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_\mathsf{k}^0, \mathsf{pk}_\mathsf{k}^1)$ and choose the private key accordingly to vk^*, i.e. $\mathsf{sk} = (\mathsf{vk}^*, \mathsf{sk}_1^{1-\mathsf{vk}_1^*}, \ldots, \mathsf{sk}_\mathsf{k}^{1-\mathsf{vk}_\mathsf{k}^*})$.

$\mathsf{Enc}^{\mathrm{DMQN}}$: To encrypt a plaintext m with the public key pk, sample another, different key $(\mathsf{vk}, \mathsf{sgk})$ from the signature scheme, then:

- $c_i = \mathsf{Enc}^{\mathrm{PKE}}_{\mathsf{pk}_i^{\mathsf{vk}_i}}(m)$ for $i = 1, \ldots, \mathsf{k}$.
- $\sigma = \mathsf{Sign}^{\mathrm{SS}}_{\mathsf{sgk}}(c_1, \ldots, c_\mathsf{k})$.
- Output the ciphertext $\psi = (\mathsf{vk}, c_1, \ldots, c_\mathsf{k}, \sigma)$.

$\mathsf{Dec}^{\mathrm{DMQN}}$: Upon receipt of a ciphertext ψ:

- If $\mathsf{vk} = \mathsf{vk}^*$ or $\mathsf{Ver}^{\mathrm{SS}}_{\mathsf{vk}}((c_1, \ldots, c_\mathsf{k}), \sigma) = 0$ output $\perp$.
- Otherwise compute $m = \mathsf{Dec}^{\mathrm{PKE}}_{\mathsf{sk}_i^{\mathsf{vk}_i}}(c_i)$ for some i such that $\mathsf{vk}_i \neq \mathsf{vk}_i^*$.
- Verify that $c_i = \mathsf{Enc}^{\mathrm{PKE}}_{\mathsf{pk}_i^{\mathsf{vk}_i}}(m)$ for all $i = 1, \ldots, \mathsf{k}$. If the verification is successful return the plaintext m, otherwise output $\perp$.

Since we know that $\mathsf{vk} \neq \mathsf{vk}^*$, there is at least one position in which they differ, hence the decryption process is well defined.

Remark 4. Note that, even though the encryption process is not deterministic, for McEliece encryption it is still possible to perform the check in the last step of $\mathsf{Dec}^{\mathrm{DMQN}}$. It is in fact enough to check the Hamming weight of $c_i - m\hat{G}_i$ where $\hat{G}_i$ is the generator matrix corresponding to the public key $\mathsf{pk}_i^{\mathsf{vk}_i}$. This is not clearly stated by the authors along with the description of the general scheme, but it is mentioned later on in [3, Theorem 3] for the particular case of the randomized McEliece.

The above specification of the scheme appears to be ambiguous. In fact, even assuming that the underlying encryption scheme is IND-CPA secure, the encryption step is described simply as $\mathsf{Enc}^{\mathrm{PKE}}_{\mathsf{pk}_i^{\mathsf{vk}_i}}(m)$ for $i = 1, \ldots, \mathsf{k}$, without indicating explicitly the role of the randomness. In [3, Section 4] some remarks are made

about the security and there is the suggestion that the scheme in use be the randomized McEliece scheme from [13]; however, precise details on how this should be instantiated are missing. One could in general think at the k encryptions as $c_i = \mathsf{Enc}^{\mathrm{PKE}}_{\mathsf{pk}_i^{\mathsf{vk}_i}}(m, r_i) = (r_i|m)\hat{G}_i + e_i$. In this case, since we check the Hamming weight of $c_i - (r_i|m)\hat{G}_i$, the check would obviously fail unless $r_1 = \cdots = r_{\mathsf{k}} = r$.

Remark 5. The KeyGen algorithm is slightly different from the Rosen-Segev case. In particular, $2\mathsf{k}$ keys are generated, then a random verification key vk^* is chosen and half of the private keys (the ones corresponding to vk^*) are discarded. This also implies that decryption only works when $\mathsf{vk} \neq \mathsf{vk}^*$. This technique is used in the context of the proof of Theorem 1, specifically in the second part while constructing an efficient distinguisher for the hard-core predicate. While, as we will see in the following, this is necessary for the proof (both for the original paper and for the proposed scheme), it is certainly a redundant requirement in the KeyGen process.

In light of the previous observations, we describe below a corrected description of the three algorithms composing the scheme:

$\mathsf{KeyGen}^{\mathrm{DMQN}}$: Invoke $\mathsf{KeyGen}^{\mathrm{PKE}}$ for $2\mathsf{k}$ times independently and obtain the collection of public keys $(\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_{\mathsf{k}}^0, \mathsf{pk}_{\mathsf{k}}^1)$ and the corresponding private keys $(\mathsf{sk}_1^0, \mathsf{sk}_1^1, \ldots, \mathsf{sk}_{\mathsf{k}}^0, \mathsf{sk}_{\mathsf{k}}^1)$. The former is distributed as the public key pk, while the latter is the private key sk.

$\mathsf{Enc}^{\mathrm{DMQN}}$: To encrypt a plaintext m with the public key pk, sample a key $(\mathsf{vk}, \mathsf{sgk})$ from the signature scheme *and a randomness* r, then:

- $c_i = \mathsf{Enc}^{\mathrm{PKE}}_{\mathsf{pk}_i^{\mathsf{vk}_i}}(m, r)$[2] for $i = 1, \ldots, \mathsf{k}$.
- $\sigma = \mathsf{Sign}^{\mathrm{SS}}_{\mathsf{sgk}}(c_1, \ldots, c_{\mathsf{k}})$.
- Output the ciphertext $\psi = (\mathsf{vk}, c_1, \ldots, c_{\mathsf{k}}, \sigma)$.

$\mathsf{Dec}^{\mathrm{DMQN}}$: Upon receipt of a ciphertext ψ:

- If $\mathsf{Ver}^{\mathrm{SS}}_{\mathsf{vk}}((c_1, \ldots, c_{\mathsf{k}}), \sigma) = 0$ output $\perp$.
- Otherwise compute $(m, r) = \mathsf{Dec}^{\mathrm{PKE}}_{\mathsf{sk}_i^{\mathsf{vk}_i}}(c_i)$ for some i.
- Verify that $c_i = \mathsf{Enc}^{\mathrm{PKE}}_{\mathsf{pk}_i^{\mathsf{vk}_i}}(m, r)$ for all $i = 1, \ldots, \mathsf{k}$. If the verification is successful return the plaintext m, otherwise output $\perp$.

The original construction is proved to be CCA2-secure in [3, Theorem 1]. We have constructed our own arguments for security, but due to space limitations, these have been moved to Appendix A.

[2] Note that the randomness we are expliciting here is the one necessary to realize the IND-CPA security of PKE, hence Enc is still a randomized algorithm. In particular, for the McEliece instantiation we would have $c_i = (r|m)\hat{G}_i + e_i$.

Remark 6. The follow-up paper of [2] also includes a modified version that allows to encrypt correlated inputs. Note that, however, this is still not a "direct translation" of the Rosen-Segev scheme. Moreover, improvements such as encrypting correlated inputs are not necessarily relevant when public-key encryption is used to exchange a single symmetric key (e.g. as a Key Encapsulation Mechanism, or KEM), which is (or should be) its main purpose. Therefore, in the next section, we propose a version that is simpler, and much closer to [16].

5 A direct translation of McEliece

We now explain how to realize the Rosen-Segev scheme using McEliece. The construction arises naturally if we want to be as close as possible to the original McEliece formulation. We hence follow the usual approach of the McEliece cryptosystem, that is to choose a different random error vector every time we call the evaluation algorithm; this implies that we are not using functions anymore. The construction is proved to be secure under k-correlated inputs in Theorem 2. It proceeds as follows:

Describe McEliece as a pair $\mathsf{McE} = (\mathsf{G}, \mathsf{F})$ composed by two algorithms: G is a generation algorithm that samples a description, and F is an evaluation algorithm that provides the evaluation on a given input.

Generation: on input n, k the algorithm G generates a random generator matrix G for an $[n, k]$-linear code with an efficient decoding algorithm over $\mathbb{F}_q$, computes the "scrambled" generator matrix $\hat{G}$, then publishes the public key $\hat{G}$ and stores the private key Γ.

Evaluation: on input $\hat{G}, m$ the algorithm F generates a random error vector e of fixed weight w in $\mathbb{F}_q^n$, computes $\psi = m\hat{G} + e$ and outputs the ciphertext ψ.

It is possible to invert F using the trapdoor: on input Γ and ψ, simply decode to obtain e using the decoding algorithm connected to Γ, then retrieve m using linear algebra.

We claim that this encryption process is secure under k-correlated inputs. First, we need a technical lemma.

Lemma 1. *If Assumption 2 holds for parameters $\hat{n}, k$ and $\hat{w}$, then the ensembles $\{(G, mG + e) : G \in \mathbb{F}_q^{k\times\hat{n}}, m \in \mathbb{F}_q^k, e \in \mathcal{W}_{\hat{n},\hat{w}}\}$ and $\{(G, y) : G \in \mathbb{F}_q^{k\times\hat{n}}, y \xleftarrow{R} \mathbb{F}_q^{\hat{n}}\}$ are computationally indistinguishable.*

Proof. Consider the problem of distinguishing the ensembles $\{(H, He^T) : H \in \mathbb{F}_q^{(\hat{n}-k)\times\hat{n}}, e \in \mathcal{W}_{\hat{n},\hat{w}}\}$ and $\{(H, y) : H \in \mathbb{F}_q^{(\hat{n}-k)\times\hat{n}}, y \xleftarrow{R} \mathbb{F}_q^{\hat{n}-k}\}$ as in [5] and suppose $\mathcal{A}$ is a probabilistic polynomial-time algorithm that is able to distinguish the

ensembles described above. In particular, say $\mathcal{A}$ outputs 1 if the challenge ensemble is of the form $(G, mG+e)$ and 0 otherwise. We show how to construct an adversary $\mathcal{A}'$ that solves the above problem.

Let (H, z) be the received input, where z is either He^T for a certain error vector $e \in \mathcal{W}_{\hat{n},\hat{w}}$ or a random vector of $\mathbb{F}_q^{\hat{n}-k}$. By linear algebra, is easy to find a vector $x \in \mathbb{F}_q^{\hat{n}}$ with $\mathsf{wt}(x) \geq \hat{w}$ such that $z = Hx^T$. Submit $(\tilde{G}, x)$ to $\mathcal{A}$, where $\tilde{G}$ is the generator matrix associated to H. Now, if $z = He^T$ we can write $x = \tilde{m}\tilde{G} + e$; in this case, in fact, we have $Hx^T = z = He^T \implies H(x-e)^T = 0$ and clearly this implies that $(x-e)^T$ is a codeword. Then $\mathcal{A}$ will output 1 and so will $\mathcal{A}'$. Otherwise, $\mathcal{A}$ will output 0 and so will $\mathcal{A}'$. In both cases, $\mathcal{A}'$ is able to distinguish correctly and this terminates the proof. □

Note that this was proved in [5] for the syndrome decoding (Niederreiter) case. We know [9] that the two formulations are equivalent; in particular, any adversary able to distinguish the above ensembles can be used to build an adversary for the Niederreiter case.

The security of the construction is proved in the following theorem, which closely follows the proof of [6, Th. 6.2].

Theorem 2. *Fix an integer k. If the parameters n, k, w are chosen such that decoding a random linear code with parameters $n\mathsf{k}, k$ and $w\mathsf{k}$ is hard, then the above encryption process is secure under k-correlated inputs.*

Proof. Let $\mathcal{A}$ be an adversary for the one-wayness under k-correlated inputs. We define the advantage of $\mathcal{A}$ to be

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \mathsf{Pr}[\mathcal{A}(\hat{G}_1, \ldots, \hat{G}_{\mathsf{k}}, \mathsf{F}(\hat{G}_1, m), \ldots, \mathsf{F}(\hat{G}_{\mathsf{k}}, m)) = m]$$

where $\hat{G}_1, \ldots, \hat{G}_{\mathsf{k}}$ are k independent public keys generated by G.
We assume the indistinguishability assumption holds: we can then exchange all the matrices $\hat{G}_i$ with uniform matrices U_i with a negligible advantage for the attacker. Now, let's define the $k \times n\mathsf{k}$ matrix U by concatenating the rows of the matrices U_i, i.e. $U = (U_1 | \ldots | U_{\mathsf{k}})$. We assume that the distributions $(U_1, \ldots, U_{\mathsf{k}}, \mathsf{F}(U_1, m), \ldots, \mathsf{F}(U_{\mathsf{k}}, m))$ and $(U, \mathsf{F}(U, m))$ are interchangeable without a significant advantage for the attacker. Note that in the latter the error vector used will have length $n\mathsf{k}$ and weight $w\mathsf{k}$. A formal argument for this indistinguishability assumption will be provided below.

We now invoke Lemma 1 with $\hat{n} = n\mathsf{k}$ and $\hat{w} = w\mathsf{k}$. Hence

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \mathsf{Pr}[\mathcal{A}(U, \mathsf{F}(U, m)) = m] - \mathsf{Pr}[\mathcal{A}(U, y) = m] \in \mathrm{negl}(n)$$

and since this last one is of course negligible, we conclude the proof. □

An indistinguishability assumption on error vectors Similarly to what happens for the IND-CPA security of the McEliece variant (as pointed out in Remark 3), also in this case the security we are trying to achieve is not absolute, but depends on a suitable choice of parameters. The assumption in this case is that we can replace the vector $(mU_1 + e_1 | \dots | mU_{\mathsf{k}} + e_{\mathsf{k}})$ with the vector $mU + e$, where $U = (U_1 | \dots | U_{\mathsf{k}})$ and e is a random error vector of weight $w\mathsf{k}$; in other words, we would like to argue that $e' = (e_1 | \dots | e_{\mathsf{k}})$ is indistinguishable from e. Note that $\mathsf{wt}(e') = \mathsf{wt}(e)$ but while the distribution of the error positions on e is truly pseudorandom, e' is formed by k blocks of weight w each. It is plausible that the number of vectors of this kind (that we denote $\#_{e'}$) is not too small compared to the total of error vectors with same length and weight. Unfortunately, the only estimate we can provide is not of help:

$$\frac{\#_{e'}}{|\mathcal{W}_{n\mathsf{k},w\mathsf{k}}|} = \frac{\binom{n}{w}^{\mathsf{k}}}{\binom{n\mathsf{k}}{w\mathsf{k}}} \geq \frac{\left(\frac{n}{w}\right)^{w\mathsf{k}}}{\left(\frac{ne}{w}\right)^{w\mathsf{k}}} = \frac{1}{e^{w\mathsf{k}}}. \tag{3}$$

However, the bound is not tight, and experimental evidence indicates that this ratio is much bigger.

It is possible to implement the Rosen-Segev scheme using the choice of F and G that we described above. We present the details below.

$\mathsf{KeyGen}^{\mathrm{P}}$: Invoke G for $2\mathsf{k}$ times independently and obtain the collections of public keys $\mathsf{pk} = (\mathsf{pk}_1^0, \mathsf{pk}_1^1, \dots, \mathsf{pk}_{\mathsf{k}}^0, \mathsf{pk}_{\mathsf{k}}^1)$ and private keys $\mathsf{sk} = (\mathsf{sk}_1^0, \mathsf{sk}_1^1, \dots, \mathsf{sk}_{\mathsf{k}}^0, \mathsf{sk}_{\mathsf{k}}^1)$, where $\mathsf{pk}_j^i = (\hat{G}_j)^i$ and $\mathsf{sk}_j^i = (S, P, \Gamma)_j^i$ as above.

$\mathsf{Enc}^{\mathrm{P}}$: To encrypt a plaintext m with the public key pk, sample a key $(\mathsf{vk}, \mathsf{sgk})$ and a random $x \in \{0,1\}^k$, then:

- $c_i = \mathsf{F}(\mathsf{pk}_i^{\mathsf{vk}_i}, x)$ for $i = 1, \dots, \mathsf{k}$.
- $y = m \oplus h(\mathsf{pk}_1^{\mathsf{vk}_1}, \dots, \mathsf{pk}_{\mathsf{k}}^{\mathsf{vk}_{\mathsf{k}}}, x)$.
- $\sigma = \mathsf{Sign}_{\mathsf{sgk}}^{\mathrm{SS}}(c_1, \dots, c_{\mathsf{k}}, y)$.

where vk_i represents the i-th bit of vk. The ciphertext is $\psi = (\mathsf{vk}, c_1, \dots, c_{\mathsf{k}}, y, \sigma)$.

$\mathsf{Dec}^{\mathrm{P}}$: Upon receipt of a ciphertext ψ:

- Verify the signature; if $\mathsf{Ver}_{\mathsf{vk}}^{\mathrm{SS}}((c_1, \dots, c_{\mathsf{k}}, y), \sigma) = 0$ output $\perp$.
- Otherwise compute $x_i = \mathsf{F}^{-1}(\mathsf{sk}_i^{\mathsf{vk}_i}, c_i)$ for $i = 1, \dots, \mathsf{k}$.[3]
- If $x_1 = \dots = x_{\mathsf{k}}$ then set $m = y \oplus h(\mathsf{pk}_1^{\mathsf{vk}_1}, \dots, \mathsf{pk}_{\mathsf{k}}^{\mathsf{vk}_{\mathsf{k}}}, x_1)$ and return the plaintext m, otherwise output $\perp$.

For simplicity, as in the original construction, we can assume m to be a single bit, in which case h describes a hard-core predicate for McEliece. However, the protocol extends easily to multiple bits plaintexts: as suggested in [16], to encrypt a polynomial number T of bits, it is enough to replace the hard-core predicate h with a hard-core *function* $h' : \{0,1\}^* \to \{0,1\}^T$.

The security is summarized in the following corollary.

Corollary 2. *The above encryption scheme is IND-CCA2 secure in the standard model.*

Proof. By Theorem 2, the collection of McEliece encryption schemes McE is k-correlation secure. Then this is analogous to Theorem 1, noting that the same argument applies when $\mathcal{F} = \mathsf{McE}$, i.e. f describes a randomized algorithm rather than a function. The proof uses the same steps as in Theorem 3, with the exception that in our case Lemma 3 is proved by constructing an adversary $\mathcal{A}'$ that works as a predictor for the hard-core predicate h. □

6 Conclusions

The scheme of Dowsley et al. [3] is a first proposal to translate the Rosen-Segev protocol to the McEliece setting. However, the construction is ambiguous, as we have shown in Section 4, and features some strange and unnecessary modifications such as "forgetting" half the private keys, or forbidding ciphertexts to feature the verification key vk^*. The original Rosen-Segev scheme has no such requirements.
The scheme was subsequently fixed in the follow-up joint work with Döttling [2], but still deviates substantially from the original Rosen-Segev framework. We therefore present a construction that, instead, follows more closely the original framework. We provide a choice of algorithms F and G, based on the McEliece cryptosystem, that can be used directly into the Rosen-Segev scheme. We then show that our construction is IND-CCA2 secure following the original security arguments of Rosen and Segev.

References

1. B. Biswas and N. Sendrier. McEliece cryptosystem implementation: theory and practice. In *PQCrypto*, pages 47–62, 2008.
2. N. Döttling, R. Dowsley, J. Müller-Quade, and A. C. Nascimento. A CCA2 secure variant of the McEliece cryptosystem. *IEEE Transactions on Information Theory*, 58(10):6672–6680, 2012.
3. R. Dowsley, J. Müller-Quade, and A. C. Nascimento. A CCA2 secure public key encryption scheme based on the McEliece assumptions in the standard model. In *Cryptographers? Track at the RSA Conference*, pages 240–251. Springer, 2009.

[3] By analogy with the Rosen-Segev scheme. Clearly in practice it would be much more efficient, rather than decoding k ciphertexts, to just decode one and then re-encode and test as in [3, Theorem 3].

4. J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. In *Information Theory Workshop (ITW), 2011 IEEE*, pages 282 –286, oct. 2011.
5. J.-B. Fischer and J. Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 245–255. Springer, 1996.
6. D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. In *International Workshop on Public Key Cryptography*, pages 279–295. Springer, 2010.
7. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In M. J. Wiener, editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 537–554. Springer, 1999.
8. K. Kobara and H. Imai. Semantically secure McEliece public-key cryptosystems-conversions for McEliece pkc. In *International Workshop on Public Key Cryptography*, pages 19–35. Springer, 2001.
9. Y. X. Li, R. H. Deng, and X. M. Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, 1994.
10. K. P. Mathew, S. Vasant, S. Venkatesan, and C. P. Rangan. An efficient ind-cca2 secure variant of the niederreiter encryption scheme in the standard model. In *ACISP*, pages 166–179, 2012.
11. R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116, Jan. 1978.
12. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Control and Inf. Theory*, 15(2):159–166, 1986.
13. R. Nojima, H. Imai, K. Kobara, and K. Morozov. Semantic security for the McEliece cryptosystem without random oracles. *Des. Codes Cryptography*, 49(1-3):289–305, 2008.
14. E. Persichetti. On a CCA2-secure variant of McEliece in the standard model. *IACR Cryptology ePrint Archive*, 2012:268, 2012.
15. E. Persichetti. Secure and anonymous hybrid encryption from coding theory. In P. Gaborit, editor, *Post-Quantum Cryptography: 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings*, pages 174–187, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
16. A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *Theory of Cryptography Conference*, pages 419–436. Springer, 2009.
17. F. Strenzke. A timing attack against the secret permutation in the McEliece pkc. In *PQCrypto*, pages 95–107, 2010.
18. F. Strenzke, E. Tews, H. G. Molter, R. Overbeck, and A. Shoufan. Side channels in the McEliece pkc. In *PQCrypto*, pages 216–229, 2008.
19. Y. Yoshida, K. Morozov, and K. Tanaka. CCA2 key-privacy for code-based encryption in the standard model. In *International Workshop on Post-Quantum Cryptography*, pages 35–50. Springer, 2017.

A Security Arguments for the Corrected Scheme

Theorem 3. *Assuming that PKE_{k} is IND-CPA secure and verifiable under k-correlated inputs, and that the signature scheme is one-time strongly unforgeable, the above encryption scheme is IND-CCA2-secure.*

Let $\mathcal{A}$ be an IND-CCA2 adversary. During the attack game, $\mathcal{A}$ submits m_0, m_1 and gets back the challenge ciphertext $\psi^* = (\mathsf{vk}^*, c_1^*, \ldots, c_{\mathsf{k}}^*, \sigma^*)$. Indicate with Forge the event that, for one of $\mathcal{A}$'s decryption queries $\psi = (\mathsf{vk}, c_1, \ldots, c_{\mathsf{k}}, \sigma)$, it holds

$\mathsf{vk} = \mathsf{vk}^*$ and $\mathsf{Ver}^{\mathrm{SS}}_{\mathsf{vk}}((c_1, \ldots, c_\mathsf{k}), \sigma) = 1$. The theorem is proved by means of the two following lemmas.

Lemma 2. *Pr[Forge] is negligible.*

Proof. Assume that there exists an adversary $\mathcal{A}$ for which $\mathsf{Pr}[\mathsf{Forge}]$ is not negligible. We build an adversary $\mathcal{A}'$ that breaks the security of the one-time strongly unforgeable scheme. $\mathcal{A}'$ works as follows:

Key Generation: Invoke $\mathsf{KeyGen}^{\mathrm{DMQN}}$ as above and return pk to $\mathcal{A}$.

Decryption queries: Upon a decryption query $\psi = (\mathsf{vk}, c_1, \ldots, c_\mathsf{k}, \sigma)$:

1. If $\mathsf{vk} = \mathsf{vk}^*$ and $\mathsf{Ver}^{\mathrm{SS}}_{\mathsf{vk}}((c_1, \ldots, c_\mathsf{k}), \sigma) = 1$ output $\perp$ and halt.
2. Otherwise, decrypt normally using $\mathsf{Dec}^{\mathrm{DMQN}}$.

Challenge queries: Upon a challenge query m_0, m_1:

1. Choose random $b \in \{0, 1\}$.
2. Use $\mathsf{Enc}^{\mathrm{DMQN}}$ to compute $c_i^* = \mathsf{Enc}_{\mathsf{pk}_i^{\mathsf{vk}_i^*}}(m_b, r)$ for $i = 1, \ldots, \mathsf{k}$.
3. Obtain the signature σ^* on $(c_1^*, \ldots, c_\mathsf{k}^*)$ with respect to vk^*[4].
4. Return the challenge ciphertext $\psi^* = (\mathsf{vk}^*, c_1^*, \ldots, c_\mathsf{k}^*, \sigma^*)$.

Note that, if Forge doesn't occur, the simulation of the CCA2 interaction is perfect. Therefore, the probability that $\mathcal{A}'$ breaks the security of the one-time signature scheme is exactly $\mathsf{Pr}[\mathsf{Forge}]$. The one-time strong unforgeability implies that this probability is negligible. □

Lemma 3. $\left| Pr[b = b^* \wedge \neg Forge] - \frac{1}{2} \right|$ *is negligible.*

Proof. Assume that there exists an adversary $\mathcal{A}$ for which $\left| \mathsf{Pr}[b = b^* \wedge \neg \mathsf{Forge}] - \frac{1}{2} \right|$ is not negligible. We build an adversary $\mathcal{A}'$ that breaks the IND-CPA security of PKE_k. $\mathcal{A}'$ works as follows:
Key Generation: On input the public key $(\mathsf{pk}_1, \ldots, \mathsf{pk}_\mathsf{k})$ for PKE_k:

1. Execute $\mathsf{KeyGen}^{\mathrm{SS}}$ and obtain a key $(\mathsf{vk}^*, \mathsf{sgk}^*)$.
2. Set $\mathsf{pk}_i^{\mathsf{vk}^*} = \mathsf{pk}_i$ for $i = 1, \ldots, \mathsf{k}$.
3. Run $\mathsf{KeyGen}^{\mathrm{PKE}}$ for k times and denote the resulting public keys by $(\mathsf{pk}_1^{1-\mathsf{vk}_1^*}, \ldots, \mathsf{pk}_\mathsf{k}^{1-\mathsf{vk}_\mathsf{k}^*})$ and private keys by $(\mathsf{sk}_1^{1-\mathsf{vk}_1^*}, \ldots, \mathsf{sk}_\mathsf{k}^{1-\mathsf{vk}_\mathsf{k}^*})$.

[4] Remember that in the one-time strong unforgeability game the adversary is allowed to ask to a signing oracle for the signature on one message.

4. Return the public key $\mathsf{pk} = (\mathsf{pk}_1^0, \mathsf{pk}_1^1, \ldots, \mathsf{pk}_\mathsf{k}^0, \mathsf{pk}_\mathsf{k}^1)$ to $\mathcal{A}$.

Decryption queries: Upon a decryption query from $\mathcal{A}$:

1. If Forge occurs output $\perp$ and halt.
2. Otherwise, there will be some i such that $\mathsf{vk}_i \neq \mathsf{vk}_i^*$. Decrypt normally using $\mathsf{Dec}^{\mathrm{DMQN}}$ with the key $\mathsf{sk}_i^{\mathsf{vk}_i}$ previously generated.

Challenge queries: Upon a challenge query m_0, m_1:

1. Send m_0, m_1 to the challenge oracle for the IND-CPA game of $\mathcal{A}'$ and obtain the corresponding challenge ciphertext $(c_1^*, \ldots, c_\mathsf{k}^*)$.
2. Sign $(c_1^*, \ldots, c_\mathsf{k}^*)$ using sgk^* to get the signature σ^*.
3. Return the challenge ciphertext $\psi^* = (\mathsf{vk}^*, c_1^*, \ldots, c_\mathsf{k}^*, \sigma^*)$.

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

As long as Forge doesn't occur, it is clear that the IND-CPA advantage of $\mathcal{A}'$ against PKE_k is the same as the IND-CCA2 advantage of $\mathcal{A}$ against the above scheme. Since we are assuming the IND-CPA security of PKE_k, we have the IND-CCA2 security as desired. □