Study of McEliece cryptosystem

Report in MTAT.07.022 Research Seminar in Cryptography, Spring 2015

Author: Sander Siim
Supervisor: Vitaly Skachek

May 28, 2015

1 Introduction

This report presents a study of the public-key cryptosystem (PKC) proposed by R. J. McEliece in 1978 [25]. The McEliece cryptosystem is based on error-correcting linear codes and is one of the first and so far the most succesful cryptosystem based on notions of coding theory.

The original construction in [25] uses binary Goppa codes to encrypt and decrypt messages. Many other variants of the cryptosystem using different linear codes have been proposed over the years, but most of them have been subsequently proven to be insecure by presenting efficient attacks. However, the original construction from 1978 has resisted over 30 years of cryptanalysis and is still today considered to be secure with the right choice of paramaters. This puts McEliece PKC on par with the RSA public-key cryptosystem dating back to 1977, which is the most common public-key scheme used today [31]. One can argue of course that due to the popularity of RSA, its security has also been more rigorously analyzed and is therefore more well-established. Still, the relation of the McEliece PKC to well-studied fundamental problems in coding theory gives confidence in its security [4].

Compared to RSA, the McEliece PKC in fact provides much faster encryption and decryption of messages [6]. However, the key sizes are much larger than for RSA, which is why McEliece PKC has rarely been used in practice. The growing interest in the McEliece PKC in the cryptographic community over the last years is due to the fact that it is one of the best candidates for a post-quantum secure PKC [5]. The dual variant of McEliece PKC called the Niederreiter scheme also allows to construct a secure digital signature scheme [28][9]. Even oblivious transfer can be constructed from the McEliece PKC security assumptions, which is not generally implied from the existence of PKC [12]. Overall, the McEliece PKC definitely merits attention and further analysis as it can today already be considered as a viable alternative to RSA and would also be a secure scheme in the post-quantum world.

In this report, we present the McEliece cryptosystem including some of its variants and discuss their security and best known attacks. We first present some basic concepts in coding theory in Section 2. We then formalize the McEliece cryptosystem and its dual variant – the Niederreiter scheme – in Section 3. In Section 4, we discuss the security of the McEliece cryptosystem and its variants and present the currently best-known attacks against the original construction. A good systematic overview of cryptanalysis done on the McEliece cryptosystem and its variants can be found in [13].

2 Preliminaries

2.1 Linear codes

We first present some basic concepts in coding theory that are needed to understand the McEliece cryptosystem. We begin with the fundamental concept of a linear code.

Definition 1 (Linear code)**.** *Let $\mathbb{F}$ be a finite field. An $[n,k]$-linear code $\mathcal{C}$ is a k-dimensional linear subspace of $\mathbb{F}^n$, that is, for every two codewords $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}$ and scalars $a_1, a_2 \in \mathbb{F}$, we have $a_1\mathbf{c}_1 + a_2\mathbf{c}_2 \in \mathcal{C}$.*

Throughout this paper, we are interested only in binary linear codes over the finite field $\mathbb{F}_2$. Codewords of binary linear codes can be naturally represented as bit-strings. We then define the *distance* between two words $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{F}_2^n$ as their Hamming distance and denote it as $\mathsf{d}(\mathbf{y}_1, \mathbf{y}_2)$. For a word $\mathbf{y} \in \mathbb{F}_2^n$, its *weight* $\mathsf{w}(\mathbf{y})$ is defined as the distance from the zero-vector $\mathsf{d}(\mathbf{0}, \mathbf{c})$.

The *minimum distance* of $\mathcal{C}$ is defined as the minimum distance of any two distinct codewords of $\mathcal{C}$

$$d = \min_{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C} \,:\, \mathbf{c}_1 \neq \mathbf{c}_2} \mathsf{d}(\mathbf{c}_1, \mathbf{c}_2) \ .$$

If $\mathcal{C}$ is an $[n,k]$-linear code with minimum distance d, then we say that $\mathcal{C}$ is an $[n,k,d]$-linear code. For linear codes, we can show that the minimum distance is always equal to the minimum weight of non-zero codewords in the code [32], that is,

$$d = \min_{\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}} \mathsf{w}(\mathbf{c}) \ .$$

Linear codes are used for encoding *information words* from $\mathbb{F}^k$ into *codewords* in $\mathcal{C} \subset \mathbb{F}^n$. We can see that an $[n,k]$-linear code $\mathcal{C}$ over $\mathbb{F}_q$ consists of q^k codewords, since the dimension of $\mathcal{C}$ is k and therefore, the linear combinations of any basis of $\mathcal{C}$ generate q^k distinct codewords. This means that the linear code $\mathcal{C}$ can be used to encode at most q^k distinct information words. For any $[n,k]$-linear code $\mathcal{C}$ over $\mathbb{F}$, we can define a one-to-one mapping $\mathbb{F}^k \to \mathcal{C}$, which is compactly described by the *generator matrix* of $\mathcal{C}$.

Definition 2 (Generator matrix of a linear code)**.** *For an $[n,k]$-linear code $\mathcal{C}$ over $\mathbb{F}$, its generator matrix G is a $k \times n$ matrix over $\mathbb{F}$ whose rows form a basis of $\mathcal{C}$.*

Note that the generator matrix for a linear code is generally not unique and every basis of $\mathcal{C}$ gives a different, but equivalent generator matrix for $\mathcal{C}$. Since the rows of the generator matrix form a basis of $\mathcal{C}$, then the span of row vectors of any generator matrix of $\mathcal{C}$ contain exactly all the codewords of $\mathcal{C}$. Using a generator matrix G for $\mathcal{C}$, we can define a mapping $\mathbb{F}^k \to \mathcal{C}$ for information words $\mathbf{u} \in \mathbb{F}^k$ as

$$\mathbf{u} \mapsto \mathbf{u}G \ .$$

Since the row vectors of G are linearly independent, this mapping is one-to-one. This mapping is used to encode information words into codewords for a given linear code.

Another important representation of a linear code is its *parity-check matrix*.

Definition 3 (Parity-check matrix of a linear code)**.** *Let $\mathcal{C}$ be an $[n,k]$-linear code over $\mathbb{F}$. A parity-check matrix of $\mathcal{C}$ is an $(n-k) \times n$ matrix H over $\mathbb{F}$ such that for every $\mathbf{c} \in \mathbb{F}^n$,*

$$\mathbf{c} \in \mathcal{C} \iff \mathbf{c}H^T = \mathbf{0} \ .$$

We can see that the definition of the parity-check matrix coincides with the definition of a kernel. In other words, H is a parity-check matrix for $\mathcal{C}$ iff $\mathcal{C} = ker(H)$ is the right kernel of H in $\mathbb{F}^n$. Let G be a $k \times n$ generator matrix of $\mathcal{C}$. It can be shown that since the rows of G span $\ker(H)$, then also the rows of H span $\ker(G)$ [32]. This means that the parity-check matrix for a linear code $\mathcal{C}$ can be efficiently calculated from a generator matrix of $\mathcal{C}$ by finding a basis of its kernel. The elements of this basis form the rows of a parity-check matrix for $\mathcal{C}$. Similarly, a generator matrix of a linear code can also be efficiently calculated from its parity-check matrix.

From the definition of the parity-check matrix it is easy to see that if G and H are the generator and parity-check matrices of the same linear code, then $HG^T = GH^T = 0$, where 0 is an $(n-k) \times k$ allzero matrix. Using this duality, one can also define a *dual code* for a linear code. If $\mathcal{C}$ is an $[n,k,d]$-linear code over $\mathbb{F}$ with generator matrix G, then the dual code $\mathcal{C}^T$ is defined as

$$\mathcal{C}^T = \left\{ \mathbf{x} \in \mathbb{F}^n \ : \ \mathbf{x}G^T = \mathbf{0} \right\} .$$

Namely, the generator matrix of $\mathcal{C}$ is the parity-check matrix of $\mathcal{C}^T$ and vice-versa.

A very important property of linear codes is their error correction capability. This property ensures that it is possible to decode codewords correctly even if a measured error is introduced into the codeword (possibly during transmission).

Definition 4 (Error-correcting linear code)**.** *Let $\mathcal{C}$ be an $[n,k,d]$-linear code over $\mathbb{F}$ with generator matrix G. We say that $\mathcal{C}$ can correct up to t errors, if there exists a decoding algorithm $\mathcal{D} : \mathbb{F}^n \to \mathcal{C}$ such that for every $\mathbf{u} \in \mathbb{F}^k$ and every vector $\mathbf{e} \in \mathbb{F}^n$ with weight $\mathsf{w}(\mathbf{e}) \leq t$, the word*

$$\mathbf{y} = \mathbf{u}G + \mathbf{e}$$

is always correctly decoded as $\mathcal{D}(\mathbf{y}) = \mathbf{u}$.

From the definition above we can see that using a binary linear code that can correct up to t errors, it is possible to decode codewords where up to t bits are flipped.

A standard way to implement decoding of error-correcting linear codes is using the *nearest-codeword* decoding method, which can be defined as follows. Let $\mathcal{C}$ be an $[n, k, d]$-linear code. Given a received word $\mathbf{y} \in \mathbb{F}^n$, find a codeword $\mathbf{c} \in \mathcal{C}$ that minimizes the value $\mathsf{d}(\mathbf{y}, \mathbf{c})$. Using the idea of nearest-codeword decoding, it can be shown that the following theorem holds [32].

Theorem 1. *Let $\mathcal{C}$ be an $[n, k, d]$-linear code over $\mathbb{F}$. There exists a decoding algorithm $\mathcal{D} : \mathbb{F}^n \to \mathcal{C}$ that correctly decodes codewords with up to $\lfloor (d-1)/2 \rfloor$ errors.*

Thus, the error-correcting capability of linear codes is directly related to its minimum distance and for every $[n, k, 2t+1]$-linear code, there exists a (nearest-codeword) decoding algorithm that corrects up to t errors.

We finally define the notion of *permutation equivalence* for linear codes. For a vector $\mathbf{x} \in \mathbb{F}^n$, let $(x_1, \ldots, x_n)$ denote its coordinates from $\mathbb{F}$. Let

$$\mathcal{S}_n = \{\pi : \{0,1\}^n \to \{0,1\}^n \mid \pi \text{ is a bijection}\}$$

denote the permutation group over n elements.

Definition 5 (Permutation-equivalent linear codes)**.** *Let $\mathcal{C}$ and $\mathcal{C}'$ be two $[n, k]$-linear codes over $\mathbb{F}$. We say that $\mathcal{C}$ and $\mathcal{C}'$ are permutation-equivalent if there exists a permutation $\pi \in \mathcal{S}_n$, such that*

$$\mathcal{C}' = \pi(\mathcal{C}) = \left\{(c_{\pi^{-1}(1)}, \ldots, c_{\pi(^{-1}n)}) \mid (c_1, \ldots, c_n) \in \mathcal{C}\right\} \ .$$

This definition is easily understood with regard to generator matrices. If G and G' are the generator matrices of two permutation-equivalent linear codes, then G' can be obtained from G by permuting its columns.

2.2 Binary Goppa codes

In this section, we briefly describe binary Goppa codes, which were defined by V. D. Goppa in 1970 [16] and are used in the original construction of the McEliece cryptosystem.

Definition 6 (Binary Goppa code)**.** *Let n, m and t be positive integers and let*

$$g(X) = \sum_{i=0}^{t} g_i X^i \in \mathbb{F}_{2^m}[X]$$

be a monic polynomial of degree t. Let $\mathbf{L} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_{2^m}^n$ be a tuple of n distinct elements from $\mathbb{F}_{2^m}$, such that

$$g(\alpha_i) \neq 0, \ \ \forall i : 1 \leq i \leq n \ .$$

The Goppa code $\mathcal{G} = \mathcal{G}(\alpha_1, \ldots, \alpha_n, g(X))$ *consists of all elements* $\mathbf{c} = (c_1, \ldots, c_n) \in \{0,1\}^n$ *that satisfy*

$$\sum_{i=1}^{n} \frac{c_i}{X - \alpha_i} \equiv 0 \mod g(X) \ .$$

Note that for an irreducible polynomial $g(X)$, all elements $\alpha \in \mathbb{F}_{2^m}$ satisfy $g(\alpha) \neq 0$. We then call the corresponding code an irreducible binary Goppa code. Thus, for an irreducible Goppa code, the elements of the tuple $\mathbf{L}$ can be chosen uniformly from all elements of $\mathbb{F}_{2^m}$. Throughout this paper, we assume all Goppa codes are irreducible. The dimension of $\mathcal{G}$ can then be shown to be at least $k \geq n - tm$ and for cryptographic applications, we can assume it is exactly $n - tm$ [13].

Also note that the maximal value for n is 2^m. Although maximizing n also maximizes the dimension of $\mathcal{G}$, for cryptographic applications it might be useful to choose the value of n smaller than 2^m.

It can be shown that the weight of every codeword in a Goppa code $\mathcal{G}$ is at least $2t + 1$ and thus the minimum distance of $\mathcal{G}$ is at least $2t + 1$ [13]. Therefore, we know that there exists a decoder for $\mathcal{G}$ which corrects up to $t = \lfloor (2t + 1) - 1/2 \rfloor$ errors. An efficient decoding algorithm was introduced by Patterson in 1975 [30], which requires $O(n \cdot t \cdot m^2)$ binary operations and corrects all t errors.

Thus, an irreducible polynomial $g(X) \in \mathbb{F}_{2^m}[X]$ of degree t and a codeword size of n defines an $[n, n - tm, 2t + 1]$-binary Goppa code capable of correcting up to t errors.

3 McEliece cryptosystem

3.1 Original construction

The McEliece cryptosystem uses error-correcting linear codes for encrypting messages [25]. The secret key retains the description of the structured linear code, chosen at key generation, and the public key is taken as a sufficiently "randomized" version of the same code, which is hard to distinguish from a completely random linear code. Decryption requires that there exists an efficient decoding algorithm for the chosen linear-code. Intuitively, knowing the structure of the underlying linear code (secret key) provides a trapdoor for fast decryption, but it is hard to decrypt without this knowledge.

Many variations of the cryptosystem have been proposed since McEliece published his original construction [25] using different linear codes and parameters (most notable is the Niederreiter scheme [28] presented in the next section). The original construction in [25] uses irreducible binary Goppa codes, which are well suited for cryptographic applications due to their high error-correcting capabilities and a dense generator matrix, which is hard to distinguish from a random binary matrix (no efficient algorithm for this is known).

Note that an $n \times n$ permutation matrix P is a binary matrix whose every column and every row each contains a single 1 and all other elements are zeroes. Multiplying any $k \times n$ matrix A with a permutation results in a matrix $A' = AP$ which contains

the same columns as A, but in permuted order. We now formally define the McEliece cryptosystem with respect to general linear codes.

Key generation.

- Pick a random $[n, k, 2t+1]$-linear code $\mathcal{C}$ over $\mathbb{F}_2$ that has an efficient decoding algorithm $\mathcal{D}$ that can correct up to t errors.
- Compute a $k \times n$ generator matrix G for $\mathcal{C}$.
- Generate a random $k \times k$ binary non-singular (invertible) matrix S.
- Generate a random $n \times n$ permutation matrix P.
- Compute the $k \times n$ matrix $G' = SGP$. The public key is (G', t) and the private key is $(S, G, P, \mathcal{D})$.

Encryption. To encrypt a plaintext $\mathbf{m} \in \{0,1\}^k$, choose a random vector $\mathbf{e} \in \{0,1\}^n$ of weight t and compute the ciphertext as

$$\mathbf{c} = \mathbf{m}G' + \mathbf{e} \ .$$

Decryption. To decrypt a ciphertext $\mathbf{c} \in \{0,1\}^n$, first calculate

$$\mathbf{c}P^{-1} = (\mathbf{m}S)G + \mathbf{e}P^{-1} \ .$$

Since $(\mathbf{m}S)G$ is a valid codeword for the chosen linear code and $\mathbf{e}P^{-1}$ has weight t, the decoding algorithm $\mathcal{D}$ can be applied to $\mathbf{c}P^{-1}$ to obtain $\mathbf{c}' = \mathbf{m}S$. Then calculate $\mathbf{m}$ with

$$\mathbf{m} = \mathbf{c}'S^{-1} \ .$$

Note that the public key G' corresponds to an $[n, k, 2t+1]$-linear code that is permutation-equivalent to the chosen secret key (P permutes the columns of G and S switches to a different basis of the same code). The original construction in [25] uses irreducible binary Goppa codes, for which an efficient decoding algorithm was presented by Patterson [30]. In order to apply Patterson's algorithm, the polynomial generating the Goppa code must be known. Therefore, in the case of binary Goppa codes, we can consider the public key as $(S, G, P, g(X))$, where $g(X)$ is the Goppa polynomial for the chosen code. Then $\mathcal{D}$ is implicitly Patterson's algorithm.

Bernstein et al. also propose an improved decoding algorithm, which makes use of the fact that Patterson's decoding algorithm is rather fast and uses trial-and-error to guess the locations of a few error bits before decoding the whole codeword [7]. This allows the sender to introduce more error bits into the codeword, which allows

to decrease the size of the code parameters while retaining security, overall making decryption faster.

Although any linear code could theoretically be used instead of Goppa codes, the numerous attempts of using different linear codes in the McEliece cryptosystem have been thwarted by various structural decoding attacks and using Goppa codes seems to be the most secure variant to this day (see Section 4.6).

The parameters for the Goppa code still have to be chosen carefully in order to be secure against the best known attacks for the McEliece cryptosystem. The correct choice of parameters is discussed in Section 4.5. As an illustration, originally McEliece proposed using an $[1024, 512]$-Goppa code with a degree 50 polynomial, however these parameter choices have been since empirically broken on modern hardware [7].

3.2 Niederreiter cryptosystem

A noteworthy variant of the McEliece cryptosystem was proposed by H. Niederreiter in 1986 [28]. The Niederreiter cryptosystem is very similar to the McEliece cryptosystem, but it uses a parity-check matrix instead of a generator matrix.

To introduce the Niederreiter cryptosystem we need to define the notion of a syndrome of a word in $\mathbb{F}^n$. Let $\mathcal{C}$ be an $[n, k, d]$-linear code over $\mathbb{F}$ and let H be its parity-check matrix. The *syndrome* of a word $\mathbf{y} \in \mathbb{F}^n$ is defined as

$$\mathbf{s} = \mathbf{y}H^T \ .$$

According to the definition of the parity-check matrix, the codewords of $\mathcal{C}$ are exactly those whose syndrome equals $\mathbf{0}$. Let $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{F}^n$ be two vectors. Then

$$\mathbf{y}_1 - \mathbf{y}_2 \in \mathcal{C} \iff (\mathbf{y}_1 - \mathbf{y}_2)H^T = \mathbf{0} \iff \mathbf{y}_1 H^T = \mathbf{y}_2 H^T \ .$$

The fact that $\mathbf{y}_1 - \mathbf{y}_2$ is a codeword of $\mathcal{C}$ iff the syndromes of $\mathbf{y}_1$ and $\mathbf{y}_2$ are equal is the basis for an efficient method to implement nearest-codeword decoding that is called *syndrome decoding*.

Given a word $\mathbf{y} \in \mathbb{F}^n$, a syndrome decoding algorithm $\mathcal{D}$ finds a minimum-weight word $\mathbf{e} \in \mathbb{F}^n$ such that

$$\mathbf{y}H^T = \mathbf{e}H^T \ .$$

If $\mathbf{y}$ is in the form $\mathbf{y} = \mathbf{c} + \mathbf{e}'$, where $\mathbf{c} \in \mathcal{C}$ and $\mathsf{w}(\mathbf{e}') \leq t$, then by the above reasoning, $\mathbf{e} = \mathbf{e}'$, that is, the syndrome decoding algorithm finds exactly the error vector introduced into the codeword.

Now let us define the Niederreiter cryptosystem, which is based on the idea of syndrome decoding.

Key generation.

- Pick a random $[n, k, 2t + 1]$-linear code $\mathcal{C}$ over $\mathbb{F}_2$ that has an efficient syndrome decoding algorithm $\mathcal{D}$ that can correct up to t errors.

- Compute a $(n-k) \times n$ parity-check matrix H for $\mathcal{C}$.
- Generate a random $(n-k) \times (n-k)$ binary non-singular matrix S.
- Generate a random $n \times n$ permutation matrix P.
- Compute the $(n-k) \times n$ matrix $H' = SHP$. The public key is (H', t) and the private key is $(S, H, P, \mathcal{D})$.

Encryption. To encrypt a plaintext $\mathbf{m} \in \{0,1\}^k$ with weight t, compute the ciphertext as the syndrome of $\mathbf{m}$

$$\mathbf{c} = \mathbf{m}H'^T \ .$$

Decryption. To decrypt a ciphertext $\mathbf{c}$, first calculate

$$S^{-1}\mathbf{c}^T = HP\mathbf{m}^T \ .$$

Then using linear algebra, find a vector $\mathbf{z} \in \mathbb{F}^n$ such that $H\mathbf{z}^T = HP\mathbf{m}^T$. Then $\mathbf{z} - (P\mathbf{m}^T)^T = \mathbf{z} - \mathbf{m}P^T$ is a valid codeword in $\mathcal{C}$ based on the observation of words with equal syndromes. As $\mathbf{m}P^T$ has weight t, we can therefore apply $\mathcal{D}$ on $\mathbf{z}$ to find the error vector $\mathbf{m}P^T$ and thereby $\mathbf{m}$.

Notice that in Niederreiter's scheme, the plaintext message is represented as the error of the codeword instead of the original information word. Therefore, the plaintext message needs to be additionally encoded to a weight-t vector for encryption.

In practice, the decryption algorithm of the Niederreiter scheme (syndrome decoding) can actually be implemented more efficiently than decryption for the McEliece cryptosystem. The Niederreiter scheme is the basis for the fastest public-key encryption implementation to date called McBits by Bernstein et al [6]. In contrast to the McEliece cryptosystem, the Niederreiter scheme can also be used to construct a digital signature scheme [9].

Although Niederreiter originally used generalized Reed-Solomon codes in his construction, this was later proven insecure as discussed in Section 4.6. Binary Goppa codes are also a good choice for the Niederreiter scheme as the security of the McEliece and Niederreiter cryptosystems are in fact equivalent as it is shown in [22]. That is, if an attacker can break the McEliece cryptosystem, he can also break the Niederreiter scheme and vice versa. This is easily seen, since the McEliece and Neiderreiter cryptosystems essentially use linear codes that are dual to each other and a generator matrix can efficiently computed from a parity-check matrix and vice versa.

4 Security of McEliece cryptosystem

In this section we focus on the security notions for the original McEliece cryptosystem using binary irreducible Goppa codes and discuss the best known attacks against the

cryptosystem. Since the security of McEliece and Neiderreiter schemes is equivalent [22], then all the discussed attacks implicitly concern the security of the Niederreiter scheme as well.

For the following, we will fix a Goppa code $\mathcal{G}(\alpha_1, \ldots, \alpha_n, g(X)) \subset \mathbb{F}_2^n$ with $g(X) \in \mathbb{F}_{2^m}[X]$ and $\alpha_i \in \mathbb{F}_{2^m}$, capable of correcting up to t errors. The dimension of the code is then $k = n - tm$. Let G be a binary $k \times n$ generator matrix of $\mathcal{G}$ and $G' = SGP$ a McEliece public key with $k \times k$ non-singular binary matrix S and $n \times n$ permutation matrix P.

As already pointed by McEliece in [25], there are two main ways how an adversary can attack the cryptosystem:

1. The attacker can try to recover the secret key G from the public key G' and then decrypt the message.

2. The attacker can try to decode the message directly without learning the structure of the Goppa code.

The first attack (extracting the private key from the public key) is significantly harder and the best known attack is exponential in $(n - k)$ (see Section 4.1). The second attack is more promising for the adversary since he can use an information-set decoding approach (see Sections 4.2 and 4.3).

The security of the McEliece cryptosystem is suggested by the intractability of the following fundamental problems in coding theory.

Problem 1 (General decoding problem of linear codes)**.** *Let $\mathcal{C}$ be an $[n, k]$-linear code over $\mathbb{F}$ and $\mathbf{y} \in \mathbb{F}^n$. Find a codeword $\mathbf{c} \in \mathcal{C}$ such that the distance $\mathsf{d}(\mathbf{y}, \mathbf{c})$ is minimal.*

Note that if $\mathbf{y} = \mathbf{c} + \mathbf{e}$ and $\mathsf{w}(e) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$, where d is the minimum distance of $\mathcal{C}$, then there is a unique solution to the general decoding problem due to Theorem 1.

Problem 2 (Problem of finding a codeword with given weight)**.** *Let $\mathcal{C}$ be an $[n, k]$-linear code over $\mathbb{F}$ and $w \in \mathbb{N}$. Find a codeword $\mathbf{c} \in \mathcal{C}$ such that $\mathsf{w}(\mathbf{c}) = w$.*

Both of the above problems are proven to be $\mathcal{NP}$-hard [4]. However, this does not directly imply that breaking the McEliece cryptosystem is $\mathcal{NP}$-hard, since irreducible binary Goppa codes only cover a fraction of all possible linear codes. Therefore, the security of the McEliece cryptosystem relies on the assumption that the public key is indistinguishable from a random matrix. So far, this assumption seems to hold for the original McEliece construction using binary Goppa codes. A formal security reduction for breaking the Niederreiter scheme by decoding a random linear code or distinguishing the public key from a random matrix is given in [34].

In the next sections, we discuss both attacks which try to recover the secret key from the public key (Section 4.1) and attacks which aim to extract the plaintext message from a single ciphertext (Section 4.2 and 4.3). So far, there is no known sub-exponential algorithm for neither type of attack. Note that the classical McEliece PKC construction

presented in the previous section only provides very weak formal security. That is, given one arbitrary ciphertext, the attacker is not able to decrypt the whole plaintext. However, efficient modificiations exist that also provide formal IND-CPA and IND-CCA2 security based on the same hardness assumptions, which we summarize briefly in Section 4.4.

In Section 4.5 we discuss optimal secure parameters for the McEliece cryptosystem based on the best known attacks. We also give an overview of attempts to use other linear codes besides Goppa codes in Section 4.6.

4.1 Attacks against private key

There are very few attacks known against the McEliece cryptosystem that can recover the secret key from the public key. The best known attack was proposed by Loidreau and Sendier in 2001 [23]. The presented attack is only feasible when a specific weak key is used, namely, when the Goppa polynomial has binary coefficients ($\mathbb{F}_2$ rather than $\mathbb{F}_{2^m}$).

The idea of the attack is to use the support-splitting algorithm proposed by Sendier [33], which allows to calculate the permutation between two permutation-equivalent linear codes. The general attack then exhaustively searches for a permutation-equivalent code with the McEliece public key among all possible Goppa codes to find the secret key. For the specific case where the coefficients of the Goppa polynomial are from $\mathbb{F}_2$, the attack can be made much faster, but still takes a considerable amount of computation to break a single key [23].

In the general case, this attack quickly becomes infeasible for any reasonable choice of parameters for the Goppa code, since roughly $2^{m(t-3)}/mt$ Goppa codes need to be enumerated and the support-splitting algorithm has to be run in every iteration [23]. For the original McEliece parameters $n = 1024$ and $k = 512$ this mounts up to roughly 2^{461}.

4.2 Generalized information-set decoding attack

A relevant attack to recover the message from a McEliece ciphertext was already presented by McEliece in his original paper [25] and further improved by Lee and Brickell in 1988 [21]. In fact, the proposed attack is a general algorithm for decoding any error-correcting linear code and therefore solves the $\mathcal{NP}$-hard general decoding problem (Problem 1). As can be expected, the attack runs in exponential time. However, it is useful to analyze as the same general idea is used in subsequent more efficient attacks.

The attack is based on the information-set decoding method. We first present the general idea following the original description of the attack by McEliece [25]. Let $\mathcal{C}$ be an $[n, k, 2t + 1]$-linear code over $\mathbb{F}$ and G its generator matrix. We assume that the adversary does not know an efficient decoding algorithm for $\mathcal{C}$ (i.e $\mathcal{C}$ is a binary Goppa code and the generating polynomial is unknown). Let $\mathbf{c}$ be a McEliece ciphertext

$\mathbf{c} = \mathbf{m}G + \mathbf{e}$, where $\mathbf{m} \in \mathbb{F}^k$, $\mathbf{e} \in \mathbb{F}^n$ and $\mathsf{w}(\mathbf{e}) = t$. We denote by $G_{\mathcal{I}}$ the matrix which contains only columns at indexes from $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ of G. To decrypt $\mathbf{c}$ the attacker does the following:

1. The attacker randomly chooses k indexes $\mathcal{I} \subset \{1, 2, \ldots, n\}$, $|\mathcal{I}| = k$, in hopes that there is no error in $\mathbf{c}$ at those indexes.

2. Then the following relationship holds: $\mathbf{c}_{\mathcal{I}} = \mathbf{m}G_{\mathcal{I}} + \mathbf{e}_{\mathcal{I}}$. Note that if $\mathsf{w}(\mathbf{e}_{\mathcal{I}}) = 0$, then the attacker can find $\mathbf{m}$ by calculating $\mathbf{m} = \mathbf{c}_{\mathcal{I}} G_{\mathcal{I}}^{-1}$.

3. The attacker checks whether $\mathsf{w}(\mathbf{c}_{\mathcal{I}} G_{\mathcal{I}}^{-1} G + \mathbf{c}) = t$. If that is the case, then the attacker calculates $\mathbf{m} = \mathbf{c}_{\mathcal{I}} G_{\mathcal{I}}^{-1}$. Otherwise, go back to Step 1.

The expected work factor for this attack is roughly

$$W = k^3 \frac{\binom{n}{k}}{\binom{n-t}{k}}$$

since the probability of having no errors in the chosen k indexes is $\binom{n-t}{k} / \binom{n}{k}$ and matrix inversion has complexity roughly k^3.

However, this attack can be somewhat improved by trying to guess the correct error vector $\mathbf{e}_{\mathcal{I}}$ with weight $\mathsf{w}(\mathbf{e}_{\mathcal{I}}) \leq j$ for some small $j < t$ [21]. The improved algorithm then goes as follows:

1. The attacker fixes $j < t$ and randomly chooses k indexes $\mathcal{I} \subset \{1, 2, \ldots, n\}$, $|\mathcal{I}| = k$.

2. The relationship $\mathbf{c}_{\mathcal{I}} = \mathbf{m}G_{\mathcal{I}} + \mathbf{e}_{\mathcal{I}}$ holds. The attacker calculates $Q = G_{\mathcal{I}}^{-1} G$.

3. For all vectors $\mathbf{e}_{\mathcal{I}}$ with weight $\mathsf{w}(\mathbf{e}_{\mathcal{I}}) \in \{0, 1, \ldots, j\}$, the attacker does the following. Since $\mathbf{c}_{\mathcal{I}} Q = \mathbf{m}G + \mathbf{e}_{\mathcal{I}} Q$, the attacker calculates $\mathbf{e}' = \mathbf{c} + \mathbf{c}_{\mathcal{I}} Q + \mathbf{e}_{\mathcal{I}} Q$.

4. If $\mathsf{w}(\mathbf{e}') = t$, the attacker returns $\mathbf{m} = (\mathbf{c}_{\mathcal{I}} + \mathbf{e}_{\mathcal{I}}) G_{\mathcal{I}}^{-1}$. Otherwise continue from Step 3. If all $\mathbf{e}_{\mathcal{I}}$ have been unsuccesfully tried, go back to Step 1.

The expected number of tries to choose $\mathcal{I}$ such that there are at most j errors in $\mathbf{c}_{\mathcal{I}}$ is

$$T_j = \frac{\binom{n}{k}}{\sum_{i=0}^{j} \binom{t}{i} \binom{n-t}{k-i}} \ .$$

The number of error vectors $\mathbf{e}_{\mathcal{I}}$ with $\mathsf{w}(\mathbf{e}_{\mathcal{I}}) \leq j$ is

$$N_j = \sum_{i=0}^{j} \binom{k}{i} \ .$$

Therefore, the expected work factor for the improved algorithm is

$$W_j = T_j(k^3 + kN_j) \ .$$

Notice that choosing $j = 0$ exactly corresponds to the simpler version of the attack. When choosing $n = 1024$, the parameters $k = 654$ and $t = 37$ maximize the work factor $W_0 \approx 2^{84.1}$. To illustrate the improvement, the same paramater choices give $W_2 \approx 2^{73.4}$, for which $j = 2$ is optimal [21]. The authors also introduce further algorithmical optimizations to the attack to reduce this to $W_1' \approx 2^{66}$ with $j = 1$.

We can see that this attack already does not provide 80-bit security for the McEliece cryptosystem when the codeword size is $n = 1024$. The next section discusses further improvements on this idea and the currently best known attack to recover the plaintext message from a ciphertext.

4.3 Finding low-weight-codeword attacks

Based on the information-set decoding idea, many authors have presented more efficient attacks against the ciphertext than the ones discussed in the previous section. Finding the plaintext from a ciphertext $\mathbf{y} \in \mathbb{F}_{2^n}$ encrypted with a linear code $\mathcal{C}$ can in fact be reduced to finding a low-weight codeword in a slightly larger linear code as noted by Canteaut and Chabaud [8]. This reduction thus relates the attack to the $\mathcal{NP}$-hard problem of finding codewords of specific weight (Problem 2). Again, this attack is not specific to Goppa codes, but can be applied for any error-correcting linear code.

Let $\mathcal{C}$ be an $[n, k, 2t + 1]$-linear code over $\mathbb{F}$. Let $\mathbf{y} \in \mathbb{F}_{2^n}$ be a word for which the closest codeword is $\mathbf{c} \in \mathcal{C}$ with $\mathsf{d}(\mathbf{y}, \mathbf{c}) = t$. Then we can show that $\mathbf{y} + \mathbf{c}$ is the unique weight-t codeword in an extended code $\mathcal{C} + \{\mathbf{y}\}$. By $\mathcal{C} + \{\mathbf{y}\}$ we mean that the vector $\mathbf{y}$ is appended to the generator matrix of $\mathcal{C}$ as a new row.

Since the minimum distance of $\mathcal{C}$ is $2t + 1$, then $\mathbf{y}$ cannot be a codeword in $\mathcal{C}$. Therefore, the generator matrix of $\mathcal{C}$ with $\mathbf{y}$ added as a row generates a new linear code $\mathcal{C}' = \{\mathbf{y} + \mathbf{x} \mid \mathbf{x} \in \mathcal{C}\}$ with dimension $k + 1$ and also $\mathbf{y} + \mathbf{c} \in \mathcal{C}'$.

Due to Theorem 1, $\mathbf{c}$ must be the unique codeword in $\mathcal{C}$ with distance t from $\mathbf{y}$, since the minimum distance of $\mathcal{C}$ is $2t + 1$. Thus, the codeword $\mathbf{y} + \mathbf{c}$ is in fact the only codeword in $\mathcal{C}'$ with weight t. If the attacker can find this codeword, he can easily solve for the plaintext [7].

The currently best known attack of this type was presented by Bernstein in 2008 [7], which is an improved modification of an earlier similar attack by Stern [37]. All the attacks of this type first try to find a low-weight codeword in a smaller linear code by choosing a subset of the columns of the generator matrix. They then check whether the found codeword has the desired weight in the original code. The attacks differ only in how they choose the smaller linear code and the strategy of finding the low-weight keyword there [13].

However, all of the advanced information-set decoding attacks of this type can still be shown to have a cost of $c^{(1+o(1))n/\lg n}$, where $c = 1/(1 - R)^{1-R}$ and R is the rate of the linear code k/n [5]. Therefore, the best known attacks against McEliece today are

still exponential in the size of the linear code, which allows choosing the parameters sufficiently large to defend against these attacks.

4.4 CPA and CCA2 security

The original McEliece cryptosystem can only be shown to guarantee one-wayness against encrypting a singel plaintext message. That is, an attacker without any knowledge of the target plaintext and no access to decryption oracles is not able to completely decrypt a single ciphertext. However, the original construction is susceptible to partially-known-plaintext, related-plaintext, reaction attacks and does not provide security against malleability as discussed in [19]. For example, when encrypting the same plaintext twice, it is very unlikely that the errors are introduced in the same places. For $\mathbf{c} = \mathbf{m}G + \mathbf{e}$ and $\mathbf{c}' = \mathbf{m}G + \mathbf{e}'$, the attacker learns $\mathbf{c} + \mathbf{c}' = \mathbf{e} + \mathbf{e}'$, which gives information about the error positions in the ciphertexts, making the previously discussed information-set decoding attacks much easier.

In practice, a PKC can be considered secure to use if it provides IND-CPA (indistinguishability under chosen-plaintext attacks) or IND-CCA2 (indistinguishability under adaptive chosen-ciphertext attacks) security, depending on the context. For most scenarios, IND-CCA2 security is required. For definitions of ciphertext indistinguishability properties of public-key cryptosystems, we refer the reader to [39]. For a more formal treatment, see [18].

However, Kobaia and Imai present efficient conversions to IND-CCA2 security which nullify all the aforementioned attacks [19]. These conversions however require the use of random oracles (hash functions) [2] to randomize the input and break relations of the plaintext and ciphertext. The authors formally prove semantic IND-CCA2 security based on the assumption that an adversary cannot fully decrypt an arbitrary McEliece ciphertext without access to decryption oracles or any knowledge about the plaintext. We have seen in the previous section that indeed, no polynomial-time algorithm for this is currently known.

Nojima et al. also present a construction using randomized padding that provides semantic IND-CPA security under standard assumptions only [29]. Döttling et al further build a IND-CCA2-secure public key cryptosystem in the standard model based on this randomized McEliece construction [11]. However, the scheme in the standard model has much larger ciphertext-expansion rate, since it involves encrypting a single message k times with different public keys, thus a random oracle based instantiation is better suited for practical applications. The best IND-CCA2 conversion in the random oracle reported in [19] actually has even lower ciphertext-expansion rate than the original McEliece construction for the same security level.

4.5 Secure optimal parameters for McEliece

Here we present the current state-of-the-art parameters for the Niederreiter scheme using an $[n, k, 2t + 1]$-irreducible binary Goppa code with $g(X) \in \mathbb{F}_{2^m}[X]$ presented

in [6]. The security level is the same for McEliece PKC, but the public key size is calculated for a systematic parity-check matrix. We present the best parameters that provide the smallest key size for given level of security in Table 1.

Security	2^m	n	k	t	Key size (bytes)
81	2048	2048	1751	27	65 006
105	4096	2480	1940	45	130 950
129	4096	4096	3604	41	221 646
187	8192	4624	3389	95	523 177
263	8192	6960	5413	119	1 046 739

Table 1: Parameters for Niederreiter PKC with binary Goppa codes that minimize public key size

4.6 Using other linear codes

In this section, we present an historic overview of the different variants of the McEliece cryptosystem that have been proposed over the years using different linear codes to try to reduce the key sizes and improve ciphertext expansion rate compared to the original construction. As of today, most of the variants have been showed to be insecure. However, recent constructions using specific low-density parity-check (LDPC) codes have not yet been successfully attacked [1] [27]. Although, these variants greatly reduce the key sizes, which is the most significant drawback of the original McEliece PKC, they have only recently been proposed and the original construction can still be considered to have the most well-established security.

Niederreiter originally used generalized Reed-Solomon codes (GRS) in his construction from 1986 [28]. However, using GRS was proven to be insecure by Sidelnikov and Shostakov in 1992 [36], in particular, they presented an attack to recover the secret parameters of the GRS code in polynomial time.

Sidelnikov in 1994 proposed using Reed-Muller codes instead for the Niederreiter variant allowing for very fast decryption, smaller key sizes and a ciphertext expansion rate close to 1. However, an efficient attack was presented against Reed-Muller codes by Minder and Shokrollahi [26] in 2007, which recovers the private key from the public key.

In 2005, Berger and Loidreau presented a construction of McEliece PKC based on sub-codes of Reed-Solomon codes [3], but an efficient attack recovering the secret key was found by Wieschebrink in 2010 for all practical parameters of the cryptosystem [38].

Gabidulin codes have also been proposed to be used in the McEliece cryptosystem due to smaller code sizes in 1991 [14]. This scheme was also proven insecure by Gibson in 1996, who presented an attack recovering the secret key efficiently for practical parameters of the Gabidulin variant [15]. Although the parameters could be modified to make this attack infeasible, the advantages of using Gabidulin codes over Goppa codes are lost with larger parameter sizes.

Using algebraic geometry codes was proposed by Janwa and Morenoin 1995 [17], but a recent paper from 2014 presents an efficient attack against this scheme as well [10].

Convolutional codes were used to construct a new variant of McEliece PKC by Löndahl and Johansson in 2012 [24], but an efficient attack was found the next year by Landais and Tillich [20].

4.7 Post-quantum security

The McEliece cryptosystem is immune to Shor's algorithm [35], which will break other public-key cryptosystems such as RSA and ECDSA if an efficient quantum computer is built. In [5], Bernstein presents the currently most efficient attack against McEliece using Grover's quantum algorithm. Bernstein shows that for the McEliece cryptosystem to retain post-quantum security, the key sizes would have to be quadrupled.

5 Conclusion

In this report, we presented a survey of the McEliece public-key cryptosystem, the security of which is based on the hardness of decoding general linear codes. The McEliece cryptosystem today presents a viable alternative to RSA due to very efficient encryption and decryption. We have shown that the best-known attacks against the original construction using binary Goppa codes are all exponential in the size of the code. Furthermore, the McEliece cryptosystem is also post-quantum secure when quadrupling the key sizes.

Acknowledgment

The author of this report has received the Skype and IT Academy Master's Scholarship for the academic year 2014/15, funded by Estonian Information Technology Foundation and Skype Technologies OÜ.

References

[1] M. Baldi, M. Bodrato, and F. Chiaraluce. A new analysis of the McEliece cryptosystem based on QC-LDPC codes. In R. Ostrovsky, R. D. Prisco, and I. Visconti, editors, *Security and Cryptography for Networks, 6th International Conference, SCN 2008, Amalfi, Italy, September 10-12, 2008. Proceedings*, volume 5229 of *Lecture Notes in Computer Science*, pages 246–262. Springer, 2008.

[2] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In D. E. Denning, R. Pyle, R. Ganesan, R. S. Sandhu,

and V. Ashby, editors, *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993.*, pages 62–73. ACM, 1993.

[3] T. P. Berger and P. Loidreau. How to mask the structure of codes for a cryptographic use. *Des. Codes Cryptography*, 35(1):63–79, 2005.

[4] E. Berlekamp, R. McEliece, and H. Van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, May 1978.

[5] D. J. Bernstein. Grover vs. McEliece. In N. Sendrier, editor, *Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings*, volume 6061 of *Lecture Notes in Computer Science*, pages 73–80. Springer, 2010.

[6] D. J. Bernstein, T. Chou, and P. Schwabe. McBits: Fast constant-time code-based cryptography. In G. Bertoni and J. Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, volume 8086 of *Lecture Notes in Computer Science*, pages 250–272. Springer, 2013.

[7] D. J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. In J. A. Buchmann and J. Ding, editors, *Post-Quantum Cryptography, Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17-19, 2008, Proceedings*, volume 5299 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2008.

[8] A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.

[9] N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a mceliece-based digital signature scheme. In C. Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 157–174. Springer, 2001.

[10] A. Couvreur, I. M. Corbella, and R. Pellikaan. A polynomial time attack against algebraic geometry code based public key cryptosystems. In *2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, June 29 - July 4, 2014*, pages 1446–1450. IEEE, 2014.

[11] N. Döttling, R. Dowsley, J. Müller-Quade, and A. C. A. Nascimento. A CCA2 secure variant of the McEliece cryptosystem. *IEEE Transactions on Information Theory*, 58(10):6672–6680, 2012.

[12] R. Dowsley, J. van de Graaf, J. Müller-Quade, and A. C. A. Nascimento. Oblivious transfer based on the McEliece assumptions. In R. Safavi-Naini, editor, *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, volume 5155 of *Lecture Notes in Computer Science*, pages 107–117. Springer, 2008.

[13] D. Engelbert, R. Overbeck, and A. Schmidt. A summary of McEliece-type cryptosystems and their security. *IACR Cryptology ePrint Archive*, 2006:162, 2006.

[14] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and their applications in cryptology. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 482–489. Springer, 1991.

[15] K. Gibson. The security of the Gabidulin public key cryptosystem. In U. M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 1996.

[16] V. D. Goppa. A new class of linear correcting codes. *Problems of Information Transition*, 6(3):207–212, 1970.

[17] H. Janwa and O. Moreno. McEliece public key cryptosystems using algebraic-geometric codes. In *Information Theory, 1995. Proceedings., 1995 IEEE International Symposium on*, pages 484–, Sep 1995.

[18] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[19] K. Kobara and H. Imai. Semantically secure McEliece public-key cryptosystems—conversions for McEliece PKC. In K. Kim, editor, *Public Key Cryptography, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001, Cheju Island, Korea, February 13-15, 2001, Proceedings*, volume 1992 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2001.

[20] G. Landais and J. Tillich. An efficient attack of a McEliece cryptosystem variant based on convolutional codes. In P. Gaborit, editor, *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings*, volume 7932 of *Lecture Notes in Computer Science*, pages 102–117. Springer, 2013.

[21] P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In C. G. Günther, editor, *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques,*

Davos, Switzerland, May 25-27, 1988, Proceedings, volume 330 of *Lecture Notes in Computer Science*, pages 275–280. Springer, 1988.

[22] Y. Li, R. H. Deng, and X. Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, 1994.

[23] P. Loidreau and N. Sendrier. Weak keys in the McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3):1207–1211, 2001.

[24] C. Löndahl and T. Johansson. A new version of McEliece PKC based on convolutional codes. In T. W. Chim and T. H. Yuen, editors, *Information and Communications Security - 14th International Conference, ICICS 2012, Hong Kong, China, October 29-31, 2012. Proceedings*, volume 7618 of *Lecture Notes in Computer Science*, pages 461–470. Springer, 2012.

[25] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116, Jan. 1978.

[26] L. Minder and A. Shokrollahi. Cryptanalysis of the Sidelnikov cryptosystem. In M. Naor, editor, *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*, volume 4515 of *Lecture Notes in Computer Science*, pages 347–360. Springer, 2007.

[27] R. Misoczki, J. Tillich, N. Sendrier, and P. S. L. M. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *Proceedings of the IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12, 2013*, pages 2069–2073.

[28] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory. *Problemy Upravlenija i Teorii Informacii*, 15(2):159–166, 1986.

[29] 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.

[30] N. Patterson. The algebraic decoding of Goppa codes. *IEEE Transactions on Information Theory*, 21(2):203–207, Mar 1975.

[31] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[32] R. M. Roth. *Introduction to coding theory.* Cambridge University Press, 2006.

[33] N. Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Transactions on Information Theory*, 46(4):1193–1203, 2000.

[34] 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.

[35] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[36] V. M. Sidelnikov and S. O. Shestakov. On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 2(4):439–444, 1992.

[37] J. Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988.

[38] C. Wieschebrink. Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In N. Sendrier, editor, *Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings*, volume 6061 of *Lecture Notes in Computer Science*, pages 61–72. Springer, 2010.

[39] Wikipedia. Ciphertext indistinguishability — Wikipedia, the free encyclopedia, 2015. [Online; accessed 08-May-2015].