# Reducing Key Length of the McEliece Cryptosystem

Thierry Pierre Berger, Pierre-Louis Cayrel, Philippe Gaborit, Ayoub Otmani

# Reducing Key Length of the McEliece Cryptosystem

Thierry P. Berger[1], Pierre-Louis Cayrel[1], Philippe Gaborit[1], and Ayoub Otmani[2]

[1] Université de Limoges, XLIM-DMI,
123, Av. Albert Thomas
87060 Limoges Cedex, France.
{thierry.berger,pierre-louis.cayrel,philippe.gaborit}@xlim.fr
[2] GREYC - Ensicaen - Université de Caen,
Boulevard Maréchal Juin, 14050 Caen Cedex, France.
Ayoub.Otmani@info.unicaen.fr

**Abstract.** The McEliece cryptosystem is one of the oldest public-key cryptosystem ever designated. It is also the first public-key cryptosystem based on linear error-correcting codes. The main advantage of the McEliece cryptosystem is to have a very fast encryption and decryption functions but suffers from a major drawback. It requires a very large public key which makes it very difficult to use in many practical situations. In this paper we propose a new general way to reduce the public key size through quasi-cyclic codes. Our construction introduces a new method of hiding the structure of the secret generator matrix by first choosing a subfield subcode of a quasi-cyclic code that is defined over a large alphabet and then by randomly shortening the chosen subcode. The security of our variant is related to the hardness of decoding a random quasi-cyclic code. We introduce a new decisional problem that is associated to the decoding of an arbitrary quasi-cyclic code. We prove that it is an NP-complete problem. Starting from subfield subcodes of quasi-cyclic generalized Reed-Solomon codes, we propose a system with several size of parameters from 6,000 to 11,000 bits with a security ranging from $2^{80}$ to $2^{107}$. Implementations of our proposal show that we can encrypt at a speed of 120 Mbits/s (or one octet for 120 cycles). Hence our new proposal represents the most competitive public-key cryptosystem.

**Keywords :** public-key cryptography, McEliece cryptosystem, alternant code, quasi-cyclic.

## 1 Introduction

The McEliece cryptosystem [18] represents one of the oldest public-key cryptosystem ever designated. It is also the first public-key cryptosystem based on linear error-correcting codes. The principle is to select a linear code of length $n$ and dimension $t$ that is able to efficiently correct $t$ errors. The core idea is to transform it to a random-looking linear code. A description of the original code and the transformations can serve as the private key while a description of the modified code serves as the public key. McEliece's original proposal uses a generator matrix of a binary Goppa code. The encryption function encodes a message according to the public code and adds an error vector of weight $t$. The decryption function basically decodes the ciphertext by recovering the secret code through the trapdoor which consists of the transformations. Niederreiter [20] also proposed a public-key cryptosystem based on linear codes in which the public key is a parity-check matrix. It is proved in [16] that these two systems are equivalent in terms of security. It relies upon two kinds of attacks that seek from the public data to either totally break the system, or to decrypt an arbitrary ciphertext. Any cryptosystem that is resistant to these attacks is to said to be *one-way secure under a chosen plaintext attack* (OW-CPA). The first category of attacks which are also called *structural attacks* in code-based cryptography aims at recovering the secret code or alternatively, constructing an equivalent code that can be efficiently decoded. The other class of attacks try to design decoding algorithms for arbitrary linear codes in order to decrypt a given cipher text. Such

an attack is called a *decoding attack*. The most efficient algorithms used to decode arbitrary linear codes are based on the information set decoding. A first analysis was done by McEliece in [18] then in [14, 15, 27] and lastly in [8] which is the best refinement up to now. All these algorithms solve the famous search problem of decoding random linear code. It was proved in [5] that decoding an arbitrary linear code is NP-Hard problem. However the security of the McEliece cryptosystem is not equivalent to the general problem of decoding random linear code due to the following reasons: 1) inverting the McEliece encryption function is a special case of the general problem of decoding where the error weight $t$ is set to a certain value and (2) binary Goppa codes form a subclass of linear codes. Therefore, McEliece cryptosystem is secure as long as there is no efficient algorithm that distinguishes between binary Goppa codes and random binary codes. Nobody has managed to solve this challenging problem for the last thirty years and if ever a solution appears towards that direction, this would toll the knell of the original McEliece cryptosystem. Note that this assumption is not always true for any class of codes that has an efficient decoding algorithm. For instance, Sidel'nikov and Shestakov proved in [26] that the structure of Generalised Reed-Solomon codes of length $n$ can be recovered in $O\left(n^4\right)$. Sendrier proved that the (permutation) transformation can be extracted for concatenated codes. Minder and Shokrollahi presented in [19] a structural attack that creates a private key against a cryptosystem based on Reed-Muller codes [25]. Despite these attacks on these variants of the McEliece cryptosystem, the original scheme still remains unbroken. The other main advantages of code-based cryptosystems are twofold:

1. high-speed encryption and decryption compared with other public-key cryptosystems which involve for instance modular exponentiations (even faster than the NTRU cryptosystem),
2. resistance to a putative quantum computer.

Unfortunately its major weakness is a huge public key of several hundred thousand bits in general. Currently, the McEliece public key cryptosystem satisfies OW-CPA for $n \geq 2048$ with appropriate values for $t$ and $k$ such that $W_f(2, n, k, t) \geq 2^{100}$ where $W_f(q, n, k, t)$ is the work factor of the best algorithm that decodes $t$ errors with any linear code of length $n$, dimension $k$ over the finite $\mathbb{F}_q$. For example the work factor is around $2^{106}$ if we choose $(n, k, t) = (2048, 1278, 70)$. For such parameters, the public key size is about 2.5 Mbits. It is therefore tempting to enhance the McEliece cryptosystem by finding a way to reduce the representation of a linear code as well as the matrices of the transformations.

A possible solution is to take very sparse matrices. This idea has been applied in [7] which examined the implications of using low density parity-check (LDPC) codes. The authors showed that taking sparse matrices for the linear transformations is an unsafe solution. Another recent trend first appeared in [11] tries to use quasi-cyclic codes [11, 1, 13, 12, 9]. This particular family of codes offers the advantage of having a very simple and compact description. The first proposal [11] uses subcodes of a primitive BCH cyclic code. The size of the public key for this cryptosystem is only 12Kbits. The other one [1] tries to combine these two positive aspects by requiring quasi-cyclic LDPC codes. The authors propose a public key size that is about 48Kbits. A recent work shows [21] that these two cryptosystems [11, 1] can be totally broken. The main drawbacks of [11] are: (1) the permutation that is supposed to hide the secret generator matrix is very constrained, (2) the use of sub-codes of a *completely known* BCH code. Combining these two flaws lead to a structural attack that recovers the secret permutation by basically solving an over-constrained linear system. The unique solution reveals the secret key.

This present work generalizes and strengthens the point of view developped in [11] on the use of quasi-cyclic code to reduce the size of the public key in the MacEliece scheme. In particular it develops new ideas which permit to overcome the weaknesses of this latter protocol. Our proposal is based on a general construction that starts from a family of MDS quasi-cyclic codes. We instantiate

our general approach through generalised Reed-Solomon codes which represent an important family of cyclic codes equipped with an efficient decoding algorithm. The two main threats that may undermine the security of our variant are the attack of [26], which exploits the structure of generalised Reed-Solomon codes, and the attack of [21] which makes profit of the quasi-cyclic structure. Our first improvement over [11] consists in taking subfield subcodes of generalised Reed-Solomon codes rather than considering subcodes of a BCH code. Subfield subcodes of generalised Reed-Solomon codes are also called *alternant* codes. This approach respects in a sense the McEliece's proposal since binary Goppa codes are a special case of alternant codes. The first positive effect for using quasi-cyclic alternant codes is the high number of codes that share the same parameters. The second positive effect is to be immune to the structural attack of [26] which strictly requires generalised Reed-Solomon codes to successfully operate. Consequently, the use of alternant codes permits to break the Reed-Solomon structure and avoids the classical Sidelnikov-Shestakov attack [26]. The second improvement consists in resisting to the attack of [21] by randomly shortening a very long quasi-cyclic alternant codes. For instance, one constructs codes of lengths of order $1,000$ from codes of length $2^{16}$ or $2^{20}$. Note that the idea of randomly shortening a code is a new way of hiding the structure of code which exploits the recent result of [28] in which it is proved that deciding whether a code is permutation equivalent to a shortened code is NP-complete. Hence the introduction of the random shortening operation makes harder the recovery of the code structure in two complementary ways. First, it worsens the chances of the Sidelnikov-Shestakov attack because the original generalised Reed-Solomon code is even more "degraded". Second, the use a random shortened code permits to avoid the attack [21] that exploits the quasi-cyclic structure since it requires the public code to be permutation equivalent to a subcode of a *known* code. The fact of considering a random shorter code makes it inapplicable because the shortened code to which the public code is equivalent is unknown to any attacker. These two improvements underly the assumption that there is no efficient algorithm that is able to distinguish between a randomly shortened quasi-cyclic alternant code and random quasi-cyclic linear code.

The main achievement of this paper is to derive a construction which drastically reduces the size of the public key of the McEliece and Niederreiter cyrptosystem to about thousands bits. The security of our new variant assumes that there is no efficient algorithm that distinguishes between a randomly shortened quasi-cyclic alternant code and random quasi-cyclic linear code. Assuming that is computationally impossible to distinguish between a randomly shortened quasi-cyclic alternant code and random quasi-cyclic linear code, the one-wayness under chosen plaintext attack is guaranteed by the hardness to decode an arbitrary quasi-cyclic linear code. We prove that the associated decisional problem is NP-complete as it is the case for arbitrary linear codes. This important result makes it reasonable to assume that there is no efficient algorithm that decode any arbitrary quasi-cyclic code. This important fact establishes the security of our variant.

The paper is organized as follows. In Section 2 we recall basic facts for code-based cryptography. In Section 3 we summarize the coding tools we use for our new algorithm: quasi-cyclic codes, generalized Reed-Solomon codes and alternant codes. In Section 4 we deal with a description of the protocol and in Sections 5 we give parameters for our scheme and study its performance. In Section 6 we consider the security of the scheme.

## 2 Code-Based Cryptography

### 2.1 Public-Key Cryptography

A public-key cryptosystem uses a *trapdoor one-way function* $E$ that will serve as the *encryption function*. The calculation of the inverse $E^{-1}$ also called the *decryption function* is only possible

thanks to a secret (the trapdoor) $K$. This concept of trapdoor one-way function forms the basis of the public-key cryptography in which the *private key* is $K$ and the public key is $E$. More precisely a public-key cryptosystem should provide three algorithms namely KeyGen, Encrypt and Decrypt algorithms. KeyGen is a probabilistic polynomial-time algorithm which given an input $1^\kappa$, where $\kappa \geq 0$ is a security parameter, outputs a pair $(\mathsf{pk}, \mathsf{sk})$ of public/private key. The KeyGen also specifies a finite *message space* $M_{\mathsf{pk}}$. The Encrypt is a probabilistic polynomial-time algorithm that on inputs $1^\kappa$, pk and a word $\boldsymbol{x}$ in $M_{\mathsf{pk}}$ outputs a word $\boldsymbol{c}$. The decryption is a deterministic polynomial-time algorithm that on inputs $1^\kappa$, sk and a word $\boldsymbol{c}$ outputs a word $\boldsymbol{x}$. The cryptosystem should satisfy the *correctness* property which means that the decryption must undo the encryption.

### 2.2 McEliece Cryptosystem

The McEliece cryptosystem [18] utilizes error-correcting codes that have an efficient decoding algorithm in order to build trapdoor one-way functions. We refer the reader to [17] for a detailed treatment of coding theory. McEliece proposed binary Goppa codes as the underlying family of codes. The parameters of a binary Goppa code are $[2^m, 2^m - mt, \geq 2t + 1]$ where $m$ and $t$ are non-negative integers. Additionally, a binary Goppa code can be decoded by an efficient $t$-bounded decoding algorithm [17]. The principle of the McEliece cryptosystem is to randomly pick a code $\mathscr{C}$ among the family of binary Goppa codes. The private key is the Goppa polynomial of $\mathscr{C}$. The public key will be a generator matrix that is obtained from the private key and by two random linear transformations: the *scrambling* transformation $S$, which sends the secret matrix $G$ to another generator matrix, and a permutation transformation which reorders the columns of the secret matrix. Figure 1 and Figure 2 give details of the three algorithms. The role of these transformation is to hide any visible structure that may characterise the code $\mathscr{C}$. Therefore, McEliece encryption scheme resists to a total break as long as it is impossible to distinguish between a permuted Goppa code and a random linear code. Based on this fact, the other security assumption is the One-Wayness of the McEliece encryption function. This is due to two facts: it is proven in [5] that decoding a random linear code is NP-Hard, and second the best known algorithms, which are based on information set decoding, [8] and [23, Volume I, Chapter 7] operate exponentially with the length $n$ and the rate of the underlying code (see also [10]). We denote by $W_f(q, n, k, t)$ the number of binary operations (also called the *work factor*) of the best known algorithm for decoding $t$ errors with a random linear code of length $n$ and dimension $k$ over $\mathbb{F}_q$. Currently, the McEliece public key cryptosystem satisfies OW-CPA for $n \geq 2048$ with appropriate values for $t$ and $k$ such that $W_f(2, n, k, t) \geq 2^{100}$. An example of values is $(n, k, t) = (2048, 1278, 70)$ because in that case the work factor is around $2^{106}$. For such parameters, the public key size is about 2.5 Megabits.

**Fig. 1.** Key generation algorithm of the McEliece cryptosystem

$\mathsf{KeyGen}(1^\kappa)$

1. Choose $n$, $k$ and $t$ such that $W_f(2, n, k, t)) \geq 2^\kappa$
2. Randomly pick a generator matrix $\boldsymbol{G}_0$ of an $[n, k, 2t + 1]$ binary Goppa code $\mathscr{C}$
3. Randomly pick a $n \times n$ permutation matrix $\boldsymbol{P}$
4. Randomly pick a $k \times k$ invertible matrix $\boldsymbol{S}$
5. Calculate $\boldsymbol{G} = \boldsymbol{S} \times \boldsymbol{G}_0 \times \boldsymbol{P}$
6. Output $\mathsf{pk} = (\boldsymbol{G}, t)$ and $\mathsf{sk} = (\boldsymbol{S}, \boldsymbol{G}_0, \boldsymbol{P}, \gamma)$ where $\gamma$ is a $t$-bounded decoding algorithm of $\mathscr{C}$

**Fig. 2.** Encryption and decryption algorithms of the McEliece cryptosystem

| $\mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{m} \in \mathbb{F}_2^k)$ | $\mathsf{Decrypt}(\mathsf{sk}, \boldsymbol{c} \in \mathbb{F}_2^n)$ |
|---|---|
| 1. Randomly pick $\boldsymbol{e}$ in $\mathbb{F}_2$ of weight $t$ | 1. Calculate $\boldsymbol{z} = \boldsymbol{c} \times \boldsymbol{P}^{-1}$ |
| 2. Calculate $\boldsymbol{c} = \boldsymbol{m} \times \boldsymbol{G} + \boldsymbol{e}$ | 2. Calculate $\boldsymbol{y} = \gamma(\boldsymbol{z})$ |
| 3. Output $\boldsymbol{c}$ | 3. Output $\boldsymbol{m} = \boldsymbol{y} \times \boldsymbol{S}^{-1}$ |

### 2.3 Niederreiter Cryptosystem

A dual encryption scheme is the Niederreiter cryptosystem [20] which is equivalent in terms of security [16] to the McEliece cryptosystem. The main difference between McEliece and Niederreiter cryptosystems lies in the description of the codes. The Niederreiter encryption scheme describes codes through parity-check matrices. But both schemes has to hide any structure through a scrambling transformation and a permutation transformation. The encryption algorithm takes as input words of weight $t$ where $t$ is the number of errors that can be decoded. We denote by $\mathcal{W}_{q,n,t}$ the words of $\mathbb{F}_q^n$ of weight $t$. Figure 3 and Figure 4 give details of the three algorithms.

**Fig. 3.** Key generation algorithm of the Niederreiter cryptosystem

$\mathsf{KeyGen}(1^\kappa)$

1. Choose $n$, $k$ and $t$ such that $W_f(2, n, k, t) \geq 2^\kappa$
2. Randomly pick a $(n-k) \times n$ parity-check matrix $\boldsymbol{H}_0$ of an $[n, k, 2t+1]$ binary Goppa code $\mathscr{C}$
3. Randomly pick a $n \times n$ permutation matrix $\boldsymbol{P}$
4. Randomly pick a $(n-k) \times (n-k)$ invertible matrix $\boldsymbol{S}$
5. Calculate $\boldsymbol{H} = \boldsymbol{S} \times \boldsymbol{H}_0 \times \boldsymbol{P}$
6. Output $\mathsf{pk} = (\boldsymbol{H}, t)$ and $\mathsf{sk} = (\boldsymbol{S}, \boldsymbol{H}_0, \boldsymbol{P}, \gamma)$ where $\gamma$ is a $t$-bounded decoding algorithm of $\mathscr{C}$

**Fig. 4.** Encryption and decryption algorithms of the Niederreiter cryptosystem

| $\mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{m} \in \mathcal{W}_{2,n,t})$ | $\mathsf{Decrypt}(\mathsf{sk}, \boldsymbol{c} \in \mathbb{F}_2^{n-k})$ |
|---|---|
| 1. Calculate $\boldsymbol{c} = \boldsymbol{H} \times \boldsymbol{m}^T$ | 1. Calculate $\boldsymbol{z} = \boldsymbol{S}^{-1} \times \boldsymbol{c}$ |
| 2. Output $\boldsymbol{c}$ | 2. Calculate $\boldsymbol{y} = \gamma(\boldsymbol{z})$ |
| | 3. Output $\boldsymbol{m} = \boldsymbol{y} \times \boldsymbol{P}$ |

### 2.4 Security of the McEliece Cryptosystem

Any public-key cryptosystem primarily requires to be resistant against an attacker that manages either to totally break the cryptosystem which consists in extracting the private data given only public data, or is able to invert the trapdoor encryption function given the ciphertexts of his choice and public data. This second security notion is also named OW-CPA for *One-Wayness under Chosen Plaintext Attack*.

**Total Break.** If we consider irreducible binary Goppa codes then there is no efficient algorithm that extracts the secret key from the public key in the McEliece or the Niederreiter cryptosystem provided that weak keys are avoided. Additionally, there is no efficient algorithm which is able to distinguish between matrices defined by the public keys and random matrices. In order words, there is no algorithm which can efficiently solve the Goppa code distinguishing.

**Definition 1 (Goppa code distinguishing).** *Given a binary $r \times n$ matrix $\boldsymbol{H}$, does $\boldsymbol{H}$ defines a binary Goppa codes?*

In order to define another important decisional problem, namely the code equivalence problem, we need some notations. We denote by $\mathsf{Diag}(\boldsymbol{v})$ the square matrix whose diagonal is $\boldsymbol{v} = (v_1, \ldots, v_n)$ and all the other entries are zero. The symmetric group $\mathcal{S}_n$ of order $n$ is the set of permutations of $J_n = \{1, \ldots, n\}$. For any $\sigma \in \mathcal{S}_n$, we denote by $\boldsymbol{P}_\sigma = [p_{i,j}]$ the $n \times n$ matrix such that $p_{i,j} = 1$ if $\sigma(i) = j$ and $p_{i,j} = 0$ otherwise. We define another important decisional problem.

**Definition 2 (Code equivalence).** *Given two $k \times n$ matrices $G$ and $G'$ over $\mathbb{F}_q$, does there exist a permutation $\sigma \in \mathcal{S}_n$, an $n$-tuple $\boldsymbol{\lambda}$ in $\mathbb{F}_q^n$ and an invertible $k \times k$ matrix $S$ such that $G' = S \times G \times \mathsf{Diag}(\boldsymbol{\lambda}) \times P_\sigma$?*

In the special case where $\mathsf{Diag}(\boldsymbol{\lambda})$ is equal to $\boldsymbol{I}_n$, the *code equivalence* problem is called the *permutation equivalence* problem. This latter problem actually reduces in the worst case to the graph isomorphism [22] which is conjectured to be in NP\P. The *support splitting* algorithm [24] was designed to solve it for the special case where the monomial transformation is exactly a permutation. Even if its complexity depends exponentially on the dimension of $\mathscr{C} \cap \mathscr{C}^\perp$, this algorithm efficiently finds the permutation in practise. Based upon these facts, the best algorithm up to now that solves the Goppa code distinguishing consists in enumerating Goppa polynomials of degree $\leq 2t$ over $\mathbb{F}_q$ and, by means of the support splitting algorithm, checking whether the corresponding Goppa code is permutation equivalent to the code defined by the public key. This strategy has time complexity $O\left(n^{2t}(1 + o(1))\right)$.

**On the One-Wayness.** The *one-wayness* property is the weakest security notion that any public-key cryptosystem must satisfy. It essentially states that inverting the encryption function is computationally impossible without a secret called the *trapdoor*. In the case of McEliece (or Niederreiter) cryptosystem, it consists in decoding a given word into a codeword of the public code. Another equivalent way of stating this problem is the *syndrome decoding problem* which constitutes an important algorithmic problem that we encounter in coding theory. It was proved NP-Complete in [5].

**Definition 3 (Syndrome decoding).** *Given an $r \times n$ matrix $H$ over $\mathbb{F}_q$, a positive integer $w \leq n$ and an $r$-tuple $\boldsymbol{z}$ in $\mathbb{F}_q^r$, does there exist $\boldsymbol{e}$ in $\mathbb{F}_q^n$ such that $\mathsf{wt}(\boldsymbol{e}) \leq w$ and $H \times \boldsymbol{e}^T = \boldsymbol{z}$?*

This important result means that decoding an arbitrary linear code is difficult (in the worst case). The issue of decoding can also be translated into the problem of finding a low-weight codeword in an appropriate code. We assume that we encrypt by means of the McEliece cryptosystem. We have a public generator $G$ and a given ciphertext $\boldsymbol{z}$. We know that there exist a codeword $\boldsymbol{c}$ in $\mathscr{C}$ and a vector $\boldsymbol{e} \in \mathbb{F}_q^n$ of weight $t$ such that $\boldsymbol{z} = \boldsymbol{c} + \boldsymbol{e}$. By hypothesis, the minimum distance of $\mathscr{C}$ is $2t + 1$. Therefore the linear code $\tilde{\mathscr{C}}$ defined by the generator matrix $\tilde{G} = \begin{bmatrix} G \\ \boldsymbol{c} \end{bmatrix}$ contains a codeword of weight $t$ namely $\boldsymbol{e}$. Thus inverting the encryption amounts to find codewords of low weight in a given code. We know that this problem is NP-Hard for an arbitrary linear code. Moreover, all the practical existing algorithms operates exponentially with length and the rate of the considered code. Currently, the most efficient one is the Canteaut-Chabaud's algorithm [8] which is an improvement upon Stern's algorithm [27]. We denote by $W_f(q, n, k, t)$ the work factor of this algorithm. Note that we can approximate $W_f(q, n, k, t) \leq \log_2^2(q) \times W_f(2, n, k, t)$.

## 3 Coding Theory Background

### 3.1 Quasi-cyclic Codes

**Definition 4.** *A* circulant *matrix $M$ is an $\ell \times \ell$ matrix obtained by cyclically right shifting the first row:*

$$\boldsymbol{M} = \begin{pmatrix} m_0 & m_1 & \cdots & m_{\ell-1} \\ m_{\ell-1} & m_0 & \cdots & m_{\ell-2} \\ \vdots & \ddots & \ddots & \vdots \\ m_1 & \cdots & m_{\ell-1} & m_0 \end{pmatrix}. \tag{1}$$

**Definition 5.** *A linear code $\mathscr{C}$ of length $n = \ell n_0$ is a* quasi-cyclic code of order $\ell$ *(and* index $n_0$*) if $\mathscr{C}$ is generated by a parity-check matrix $\boldsymbol{H} = [\boldsymbol{H}_{i,j}]$ where each $\boldsymbol{H}_{i,j}$ is an $\ell \times \ell$ circulant matrix.*

Note that a code $\mathscr{C}$ is *cyclic* if for any $\boldsymbol{c}$ in $\mathscr{C}$ the cyclic shift $(c_n, c_1, \ldots, c_{n-1})$ also belongs to $\mathscr{C}$. A cyclic code is therefore a quasi-cyclic code of any order $\ell$ which divides $n$. The interest of circulant matrices comes from the fact they are completely described by their first row. Hence, through such matrices, quasi-cyclic codes admit a very compact representation which is almost linear with the code length. We now describe from [2, 4, 3] several linear transformations that will serve us to obtain a family of quasi-cyclic codes starting from a unique quasi-cyclic code.

**Definition 6 (Shortened code).** *Let $\mathscr{C}$ be a code of length $n$ defined by a parity-check matrix $H$. Consider a subset $J'$ of $J_n$ of cardinality $n' < n$ and define $H(J')$ to be the matrix obtained by deleting columns of $H$ that are* not *in $J'$. The parity-check matrix $H(J')$ defines the* shortened *code $\mathscr{C}(J')$ over $J'$.*

**Remark:** Recall that for $T$ a set of coordinates, if one denotes by $C_T$ the code $C$ shortened in $T$ and by $C_T$ the code punctured on $T$ (deleting columns corresponding to coordinates of $T$) then $(C^\perp)_T = (C^T)^\perp$. This equality relates a punctured code with its shortened dual.

The following lemma sums up the transformation we will apply on our codes:

**Lemma 1.** *Let $\mathscr{C}$ be a quasi-cyclic code of length $n = \ell n_0$ and of order $\ell$ defined by a parity-check matrix $\boldsymbol{H} = [\boldsymbol{H}_{i,j}]$ where each $\boldsymbol{H}_{i,j}$ is an $\ell \times \ell$ circulant matrix.*

1. *Let $J$ be a subset of $J_{n_0}$ and let $\boldsymbol{H}'$ be the parity-check matrix obtained after deleting circulant matrices $\boldsymbol{H}_{i,j}$ such that $j$ does not belong to $J$. The code defined by $\boldsymbol{H}'$ is a quasi-cyclic code of order $\ell$.*
2. *Define the Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$ of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ as the matrix $[a_{i,j}\boldsymbol{B}]$. Let $\boldsymbol{a}$ be an $n_0$-tuple of elements in $\mathbb{F}_q$. The parity-check matrix $\boldsymbol{H} \times (\mathsf{Diag}(\boldsymbol{a}) \otimes I_\ell)$ defines a quasi-cyclic code of order $\ell$.*
3. *Assume that there exists $\alpha$ in $\mathbb{F}_q$ such that $\alpha^\ell = 1$, then $\boldsymbol{H} \times \left(I_{n_0} \otimes \mathsf{Diag}(1, \alpha, \ldots, \alpha^{\ell-1})\right)$ defines the parity-check matrix of a quasi-cyclic code of order $\ell$.*
4. *Let $P_\pi$ be a permutation in $\mathcal{S}_{n_0}$, then the code defined by the parity-check matrix $\boldsymbol{H} \times (\boldsymbol{P}_\pi \otimes I_\ell)$ is quasi-cyclic of order $\ell$.*
5. *Set $T_\ell$ as the cyclic shift defined by $T_\ell(\boldsymbol{v}) = (v_\ell, v_1, \ldots, v_{\ell-1})$ for any $\boldsymbol{v} \in \mathbb{F}_q^\ell$. Denote by $\mathbb{T}_\ell$ the cyclic group generated by $T_\ell$. Let $P_{\sigma_1}, \ldots, P_{\sigma_{n_0}}$ be $n_0$ permutations of $\mathbb{T}_\ell$. The code defined by the following parity-check matrix is quasi-cyclic of order $\ell$:*

$$\boldsymbol{H} \times \begin{pmatrix} P_{\sigma_1} & & 0 \\ & \ddots & \\ 0 & & P_{\sigma_{n_0}} \end{pmatrix}.$$

**Definition 7.** *Any code obtained by successively applying transformations of Lemma 1 to a code $\mathscr{C}$ is denoted by $\mathscr{C}(J, \boldsymbol{a}, \alpha, P_\pi, P_{\sigma_1}, \ldots, P_{\sigma_{n_0}})$.*

### 3.2 Generalised Reed-Solomon Codes

**Definition 8.** *Let $\alpha$ be a primitive element of $\mathbb{F}_q$ and assume that $n = q - 1$. The Reed-Solomon code $\mathscr{R}_{k,n}$ is the code of length $n$ and dimension $k$ over $\mathbb{F}_q$ defined by the parity-check matrix*

$$\boldsymbol{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \cdots & (\alpha^2)^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{d-1} & (\alpha^{d-1})^2 & & (\alpha^{d-1})^{n-1} \end{bmatrix} \tag{2}$$

*where $d = n - k + 1$.*

Reed-Solomon codes represent an important class of cyclic codes. It is well-known that $d = n - k + 1$ is actually its minimum distance. Moreover, Reed-Solomon codes admit a $t$-bounded decoding algorithm as long as $2t \leq d - 1$. They can also be seen as a sub-family of Generalised Reed-Solomon codes.

**Definition 9.** *Let $\boldsymbol{\lambda}$ be an $n$-tuple of nonzero elements in $\mathbb{F}_q$ where $n < q$ and let $\boldsymbol{v}$ be an $n$-tuple of distinct elements in $\mathbb{F}_q$. The Generalised Reed-Solomon $\mathscr{G}_k(\boldsymbol{v}, \boldsymbol{\lambda})$ code over $\mathbb{F}_q$ of length $n$ and dimension $k$ is the code defined by the following parity-check matrix where $d = n - k + 1$*

$$\boldsymbol{H}_{\boldsymbol{v},\boldsymbol{\lambda}} = \begin{bmatrix} \lambda_0 & \lambda_1 & \cdots & \lambda_{n-1} \\ \lambda_0 v_0 & \lambda_1 v_1 & \cdots & \lambda_{n-1} v_{n-1} \\ \vdots & \vdots & & \vdots \\ \lambda_0 v_0^{d-1} & \lambda_1 v_1^{d-1} & \cdots & \lambda_{n-1} v_{n-1}^{d-1} \end{bmatrix}.$$

Generalised Reed-Solomon codes are decoded in the same way as Reed-Solomon codes by means of the classical Berlekamp-Massey algorithm.

### 3.3 Subfield Subcode

The subfield subcode construction is another very important way to obtain a new class of codes. For instance Goppa codes are subfield subcodes of generalised Reed-Solomon codes. In that particular case, they are also called *alternant* codes.

**Definition 10.** *Let $\mathbb{F}_q$ be subfield of $\mathbb{F}_{q_0}$ and let $\mathscr{C}$ be a code of length $n$ over $\mathbb{F}_{q_0}$. The* subfield subcode *$\mathscr{C}'$ of $\mathscr{C}$ over $\mathbb{F}_q$ is the vector space $\mathscr{C} \cap \mathbb{F}_q^n$.*

Obviously, any subfield subcode of quasi-cyclic code of order $\ell$ is also a quasi-cyclic code of the same order. Another advantage of restricting ourselves to such subcodes is the possibility to obtain codes with a better minimum distance which hence enables to correct more errors.

**Definition 11 (Alternant code).** *Let $\mathbb{F}_q$ be a subfield of $\mathbb{F}_{q_0}$ and let $\mathscr{G}_k(\boldsymbol{v}, \boldsymbol{\lambda})$ be a Generalised Reed-Solomon code of dimension $k$ over $\mathbb{F}_q$. The* alternant *code $\mathscr{A}(\boldsymbol{v}, \boldsymbol{\lambda})$ over $\mathbb{F}_q$ of $\mathscr{G}_k(\boldsymbol{v}, \boldsymbol{\lambda})$ is the subfield subcode of $\mathscr{G}_k(\boldsymbol{\lambda}, \boldsymbol{v})$ over $\mathbb{F}_q$.*

## 4 Our New Variant

### 4.1 Description of the New Cryptosystem

The new variant of the MacEliece (or Niederreiter) cryptosystem we propose is not based on classical binary Goppa codes but rather on quasi-cyclic codes as in [11]. These codes, which are a generalization of cyclic codes, offer the advantage of having a very compact representation. However it was recently proved in [21] that replacing quasi-cyclic codes in the McEliece cryptosystem does not resist to a total break. This was made possible mainly because the secret permutation that hides the structure of the secret quasi-cyclic code can be described with much less coefficients. The secret permutation is then recovered by simply solving an over-constrained linear system. In our new approach, we overcome this weakness by considering a more general kind of linear transformations which not only reorder the columns of the secret generator matrix but also delete most of them making their retrieval very hard in practice. We assume first that we have at our disposal a family of MDS quasi-cyclic codes that we know how to decode (in practice we will consider Generalized reed-Solomon codes). (We recall that a $[n, k, d]_q$ code is Maximum Distance separable code if $d = n - k + 1$.)

The construction starts by randomly picking such a quasi-cyclic codes $\mathscr{C}_0$ of length $N = \ell N_0$, and dimension $K$ and order $\ell$ over a finite field $\mathbb{F}_{q_0}$. The code $\mathscr{C}_0$ is equipped by $t$-bounded decoding algorithm $\gamma$. We also assume that there exists an element $\beta$ of *order* $\ell$ in $\mathbb{F}_{q_0}$. Figure 5 gives KeyGen algorithm of our variant. Encrypt and Decrypt algorithms are not given because they are exactly the same as those of the original Niederreiter (or MacEliece) encryption scheme. The goal of KeyGen algorithm is to construct a new quasi-cyclic code of length $n = \ell n_0$ (with $n_0 << N_0$) and order $\ell$ over a finite subfield $\mathbb{F}_q \subset \mathbb{F}_{q_0}$ with $q_0 = q^r$. The subfield-subcode has dimension $N - r(N - K)$ where $r$ and $N - K$ are chosen such that $r(N - K) \sim n_r \ell$, for $n_r$ an integer which represents the number of rows needed to describe the quasi-cyclic matrix, in practice $n_r = 1, 2$ or 3. This is done thanks to four operations.

1. First, deleting $(N_0 - n_0)$ random block columns of circulant matrices of $\boldsymbol{H}'$. We get hence a shortened code $\mathscr{C}_1$ of length $n = n_0 l$.
2. Next, transforming $\mathscr{C}_1$ by means of the transformations given in Lemma 1. We obtain a new code $\mathscr{C}_2$ of length $n$.
3. Taking a subfield subcode of $\mathscr{C}_2$ over $\mathbb{F}_q$ in order to get the public quasi-cyclic code $\mathscr{C}$ of length $n = n_0 \ell$ and dimension $n - r(N - K)$ (with $r(N - K) \sim n_r \ell$).
4. The quasi-cyclic code obtained is described through its dual matrix of dimension $r(N-K)$ which can be described through the $\ell$-order quasi-cyclicity as a matrix $\boldsymbol{H} = [\boldsymbol{H}_{i,j}]$ with $1 \leq i \leq n_r$ and $1 \leq j \leq n_0$ where $\boldsymbol{H}_{i,j}$ is an $\ell \times \ell$ circulant matrix.

As we can see, the proposed cryptosystem uses circulant matrices to describe the codes. The public key is formed by an $n_r \ell \times n$ matrix over $\mathbb{F}_q$, where $n = \ell n_0$, formed by $n_r \times n_0$, $\ell \times \ell$ circulant matrices. It only needs $n_r \ell (n_0 - n_r) \log_2 q$ bits to be completely described in systematic form (See Section 5 for the obtained values).

### 4.2 Application to Generalised Reed-Solomon Codes

We have presented in Section 4.1 a general method to build a public-key cryptosystem from a family of quasi-cyclic codes. We give in this section how to effectively obtain quasi-cyclic codes. The idea is to select as in [11] a family of cyclic codes equipped with an efficient decoding algorithm. A good candidate is the family of Reed-Solomon codes. From this family, we generate by means of the transformations defined in Lemma 1 many new quasi-cyclic codes. Recall that the resulting public

**Fig. 5.** KeyGen algorithm of the new variant of the Niederreiter cryptosystem

1. Choose a security parameter $\kappa$
2. Choose $N_0$, $\ell$, $q$, $n_0$, $r$, $n_r$ and $t$ with $n_0 < N_0$ such that $W_f(q^r, N, K, t) \geq 2^\kappa$ and $W_f(q, n, k, t) \geq 2^\kappa$ where $N = \ell N_0$, $(N-K)r \sim n_r \ell$, $n = \ell n_0$ and $k = n - (N-K)r \sim n - n_r\ell$. Set $q_0 = q^r$.
3. Randomly pick a parity-check matrix $H_0$ of a quasi-cyclic code $\mathscr{C}_0$ of length $N$, dimension $K$ and order $\ell$ over $\mathbb{F}_{q_0}$
4. Randomly pick a $n_0$-subset $J \subset J_{N_0}$
5. Delete from $\boldsymbol{H}'$ the circulant matrices $\boldsymbol{H}'_{i,j}$ such that $i \notin J$. Set $\boldsymbol{H}_0$ the resulting matrix.
6. Randomly pick an $n_0$-tuple $\boldsymbol{a}$ of non zero elements of $\mathbb{F}_{q_0}$. Set $\boldsymbol{H}_1 = \boldsymbol{H}_0 \times (\mathsf{Diag}(\boldsymbol{a}) \otimes I_\ell)$
7. Let $\beta$ be an element of $\mathbb{F}_{q_0}$ of *order* $\ell$. Randomly pick an integer $1 \leq m \leq \ell - 1$ and set
$$\boldsymbol{H}_2 = \boldsymbol{H}_1 \times \left(I_{n_0} \otimes \mathsf{Diag}(1, \beta^m, \beta^{2m} \ldots, \beta^{(\ell-1)m})\right)$$
8. Randomly pick $P_\pi \in \mathcal{S}_{n_0}$. Set $\boldsymbol{H}_3 = \boldsymbol{H}_2 \times (\boldsymbol{P}_\pi \otimes I_\ell)$
9. Randomly pick $P_{\sigma_1}, \ldots, P_{\sigma_{n_0}}$ in $\mathbb{T}_\ell$. Set
$$\boldsymbol{H}_4 = \boldsymbol{H}_3 \times \begin{pmatrix} P_{\sigma_1} & & 0 \\ & \ddots & \\ 0 & & P_{\sigma_{n_0}} \end{pmatrix}$$
10. Choose a subfield $\mathbb{F}_q \subset \mathbb{F}_{q_0}$ Let $\Omega = \{\omega_1, \ldots, \omega_r\}$ be an arbitrary basis of $\mathbb{F}_{q_0}$ over $\mathbb{F}_q$. Decompose each entry of $\boldsymbol{H}_4$ according to $\Omega$. Set $\boldsymbol{H}_5$ the resulting $n_r\ell \times n$ matrix with $n = \ell n_0$
11. Transform $\boldsymbol{H}_5$ into an $n_r\ell \times n$ matrix $\boldsymbol{H}^* = [\boldsymbol{H}^*_{i,j}]$ where $\boldsymbol{H}^*_{i,j}$ is an $\ell \times \ell$ circulant matrix
12. Randomly pick an invertible $n_r\ell \times n_r\ell$ matrix $\boldsymbol{S} = [\boldsymbol{S}^*_{i,j}]$ for $1 \leq i, j \leq n_r$ and $S_{i,j}$ a circulant $\ell \times \ell$ matrix. Set $\boldsymbol{H} = \boldsymbol{S} \times \boldsymbol{H}^*$. (Alternatively $H$ can be put in standard form to decrease the size of the but the encryption algorthim has to be slighlty modified (see [6]).
13. Output $\mathsf{pk} = (\boldsymbol{H}, t)$ and $\mathsf{sk} = (\boldsymbol{S}, \boldsymbol{H}_0, J, \boldsymbol{a}, m, P_\pi, P_{\sigma_1}, \ldots, P_{\sigma_{n_0}})$

key is a randomly shortened alternant code denoted by $\mathscr{G}(J, \boldsymbol{a}, \beta^m, \pi, \sigma_1, \ldots, \sigma_{n_0})$ (Definiton 7) when the underlying Reed-Solomon code is $\mathscr{G}$. The matrix $\boldsymbol{H}_0$ is therefore defined by Equation (2).

## 5 Suggested Parameters and Performances

### 5.1 Suggested Parameters

We illustrate our construction with different sets of parameters in Table 1. The table sums up the different parameters, the security entry is computed from the Canteaut-Chabaud algorithm ($\log_2(W_f(q, n, k, t))$), also taking account of the use of the order of quasi-cyclicity which can be used with this attack simply by shifting the vector to decode (notice it only improves the attack linearly by the order of quasi-cyclicity $l$ and that until now this is the only way to use quasi-cyclicity for the decoding attack - we also point the fact that for NTRU no real attack has been found based on the quasi-cyclicity of the lattice). We give parameters with decoding security from $2^{80}$ to $2^{116}$ with for instance public key size from $6,500$ bits for a $2^{80}$ security to $11,160$ bits for a $2^{107}$ security. These parameters show the adaptability of our system with a size of key which increases moderatly relatively to the security of the system.

Remark that as usual for McEliece cryptosystem the parameters are easily scalable. The two cases we consider are generalised Reed-Solomon codes over $\mathbb{F}_{2^{20}}$ and $\mathbb{F}_{2^{16}}$. In the following cases the codes are generated by quasi-cyclicity from one, two or three rows.

**Example:** Consider set of parameters $A_{16}$: $2^{16} - 1 = 51 \times 1285$, hence we can take $l = 51$, $N_0 = (2^{16} - 1)/51 = 1285$. We consider the subfield $\mathbb{F}_{2^8}$ of $\mathbb{F}_{2^{16}}$, hence $r = 2$. We take a GRS cyclic code $C_0$ with $t = 25$, which gives $N = 2^{16} - 1$ and $K = 2^{16} - 1 - 50$, hence we get $r(N-K) = 100 \sim 2.\ell$ with $n_r = 2$ (the number of row eventually needed). Then we take $n_0 = 13$ and keep only 13 blocks of

size $\ell$ among the 1285 blocks possible. We then obtain a quasi-cyclic GRS $[13.51, 13.51-50, t=25]_{2^{16}}$ code of order 51. We then apply linear transformations of Lemma 1 to obtain a code $C_1$, and take the subfield subcode of $C_1$ from $\mathbb{F}_{2^{16}}$ to $\mathbb{F}_{2^8}$. We hence obtain a code $[13.51, 13.51-100, t=25]_{2^8} = [663, 561, t=25]_{2^8}$ code, we can check from the Canteaut-Chabaud algorithm that the security (taking account of the quasi-cyclic order) is $2^{80}$ and the size of the public key in systematic form is $8,980$ bits.

**Table 1.** Suggested parameters of generalised Reed-Solomon codes for different security levels

| $N = N_0\ell, K = N-2t, d = 2t+1$ | | | | Public code $\mathscr{C}[n, k, \geq 2t+1]$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $q_0$ | $\ell$ | $N_0$ | $t$ | Parameter | n | k | $q$ | $n_0$ | $n-k$ | security | Public key size (bits) |
| $2^{16}$ | 51 | 1,285 | 25 | $A_{16}$ | 663 | 561 | $2^8$ | 13 | $2\times 2\times 2t$ | 80 | 8,980 |
| | 51 | 1,285 | 37 | $B_{16}$ | 663 | 510 | $2^8$ | 13 | $3\times 2\times 2t$ | 95 | 12,240 |
| | 51 | 1,285 | 37 | $C_{16}$ | 1020 | 867 | $2^8$ | 20 | $3\times 2\times 2t$ | 116 | 20,800 |
| | 85 | 771 | 42 | $D_{16}$ | 1105 | 935 | $2^4$ | 13 | $2\times 4\times 2t$ | 83 | 14,960 |
| $2^{20}$ | 93 | 11,275 | 23 | $A_{20}$ | 744 | 651 | $2^{10}$ | 8 | $2\times 2t$ | 80 | 6,510 |
| | 93 | 11,275 | 46 | $B_{20}$ | 744 | 558 | $2^{10}$ | 8 | $2\times 2\times 2t$ | 107 | 11,160 |
| | 75 | 13,981 | 36 | $C_{20}$ | 750 | 675 | $2^{10}$ | 10 | $2\times 2t$ | 100 | 12,120 |
| | 165 | 6,355 | 40 | $D_{20}$ | 1320 | 1155 | $2^5$ | 8 | $4\times 2t$ | 90 | 11,480 |

*Remark 1.* The number of possible codes by our construction is very large: $\binom{N_0}{n_0} 2^{pn_0} n_0!$ for $\mathbb{F}_{2^p}$ the field of the larger code. For instance for parameters $A_{16}$ we obtain $2^{526}$ possible codes. The quasi-cyclic structure of the codes permits to go easily from a McEliece type public key (a generator matrix of code) to a Niedereiter type public key (a dual matrix of the code).

### 5.2 Performance

We implemented the encryption with the system $A_{20}$ on $GF(2^8)$ on 2.4 Ghz computer with 64 bits architecture. The multiplication over $GF(2^8)$ was tabulated in cache memory and all operations were done octet by octet as a matrix vector product so that eventually the 64 bits structure was not really enhanced. Overall the encryption speed was 15Mo/s, which corresponds to about 128 cycle per octet.

This speed compares well with the implementation of [6] which speed was inferior with a factor 2 to what we obtain. Moreover our implementation can still be improved (probably by a factor 2 or 3) by taking account of the cyclic structure. Indeed rather than tabulating the multiplication over the field ($GF(2^8)$ in this case) it is possible to put in cache memory the multiplication of the first row of the code by all elements of the base field. Since all the rows of the generator matrix are obtained as cyclic shift from the first row it then possible to deduce all multiplied rows from shifts of the multiplied first rows in cache. So that it is possible to profit by the 64 bits structure by summation (but in 64 bits) of the multiplied shifted first rows and even enhance our performance. For decryption we obtain similar speed of a few hundred cycle per octet as in [6] although we did not yet optimized our implementation.

Our implementation speed compares of course very well with RSA-1024 and Elliptic Curves. An interesting question is the comparison with NTRU. Although exact performance results are hard to find, in term of encryption speed our system seems better with a factor 10 by comparison to known performance of NTRU [6]. These good results comes from the fact that besides the matrix-vector product in NTRU, one also needs to encrypt vectors with given weight which becomes in fact the

main cost (notice that the same problem arises for Niederreiter version of the scheme, but in our case we only considered the McEliece scheme).

## 6 Security Analysis

In this section we analyze the security of the new variant proposed in Section 4.1. We show that this new system based on Reed-Solomon codes is not only resistant to a total break but also satisfies the OW-CPA property.

### 6.1 Total Break

We review now all the existing attacks that aim at recovering the private key from public data. These attacks which are also called *structural attacks* are listed below.

**Brute force attack** Recall that the private key is $\mathsf{sk} = (J, \boldsymbol{a}, m, \pi, \sigma_1, \ldots, \sigma_{n_0})$ The set $J$ which is a subset of $J_{N_0}$ of cardinality $n_0$. The vector $\boldsymbol{a}$ is $n_0$-tuple of nonzero elements in $\mathbb{F}_q$, the permutation $\pi$ belongs to $\mathcal{S}_{n_0}$ and each $\sigma_i$ is in $\mathbb{T}_\ell$. Note that $a_0$ can be chosen to be equal to 1. Thus the private key space contains $n_0! q^{n_0-1} \ell^{n_0+1} \binom{N_0}{n_0}$ elements. This implies that the private key is out of reach by exhaustively enumerating all the elements.

**Attack exploiting the code equivalence.** Another possible attack is to exploit the fact that the public code is equivalent (Definition 2) to a subcode of a Reed-Solomon code. One may think to employ the *support splitting* algorithm. However, one has first to find out the vector $\boldsymbol{a}$ and the integer $m$. Moreover, one has also to guess the set $J$ before applying this strategy. But the number of such sets is again too high for being at reach by brute force search. Actually, we can show that this issue forms an intractable problem called the *equivalent punctured code* problem. It was first proposed in [28]. Surprisingly, it shows that if one allows to delete some random columns then the problem becomes an NP-complete problem [28].

**Definition 12 (Equivalent shortened code problem).** *Given an $r \times n$ matrix $H$ over $\mathbb{F}_q$ and an $r \times n'$ matrix $H'$ over $\mathbb{F}_q$ with $n' < n$, does there exist $\sigma$ in $\mathcal{S}_{n'}$ and $J \subset J_n$ of cardinality $n'$ such that $H' = H(J) \times P_\sigma$?*

Unlike the permutation equivalence problem which is efficiently solved in practise by the support splitting algorithm, there is no efficient algorithm that solves the equivalent shortened code problem. The security of our new McEliece variant is related to the existence of such an algorithm.

**Attack exploiting the generalised Reed-Solomon structure.** Sidelnikov and Shestakov proved in [26] that it is possible to completely recover the structure of Generalised Reed-Solomon codes with time complexity in $O(n^3)$ where $n$ is the length of the public code. This attack can be used against any type of generalized Reed-Solomon code but uses the fact that the underlying matrix of the Reed-Solomon code is completely known. In our case we do not directly use a GRS code but a randomly shortened subfield subcode, which hence makes this attack unfeasible. It is worthwhile remarking that if such an efficient (or a generalized attack) was to exist for alternant codes, then it could be potentially used to break the McEliece cryptosystem since Goppa codes are a special case of binary alternant codes.

**Attack exploiting the quasi-cyclic structure.** Recently a new structural attack appeared in [21] that extracts the private key of the variant presented in [11]. This cryptosystem takes a binary quasi-cyclic subcode of a BCH code of length $n$ as the secret code. The structure is hidden by a strongly constrained permutation in order to produce a quasi-cyclic public code. This implies that the permutation transformation is completely described with $n_0^2$ binary entries where $n_0$ is the quasi-cyclic index rather than $n^2$ entries. The attack consists in taking advantage of the fact that the secret is a subcode of completely known BCH code. One generates linear equations by exploiting the public generator matrix and a known parity-check matrix of the BCH code so that one gets an over-constrained linear system satisfied by the unknown permutation matrix.

We show how to adapt this attack to our variant. We start from a parity-check matrix $\boldsymbol{H}_0$ of the Reed-Solomon code $\mathscr{R}_{K,N}$. We add $(N_0 - n_0)\ell$ zero columns to the public parity-check matrix $\boldsymbol{H}$ in order to form another parity-check matrix $\tilde{\boldsymbol{H}}$. By definition, we know that there exist an integer $1 \leq m \leq \ell - 1$ and an $N_0$-tuple $\boldsymbol{a}$ of elements in $\mathbb{F}_{q_0}$ (not necessarily nonzero) such that the code $\tilde{\mathscr{C}}$ defined by $\tilde{\boldsymbol{H}}$ is permutation equivalent to a subcode of the generalised Reed-Solomon code defined by the parity-check matrix $\boldsymbol{H}_2$

$$\begin{aligned}\boldsymbol{H}_2 &= \boldsymbol{H}_0 \times (\mathsf{Diag}(\boldsymbol{a}) \otimes I_\ell) \times \Big(I_{n_0} \otimes \mathsf{Diag}(1, \beta^m, \beta^{2m} \ldots, \beta^{(\ell-1)m})\Big) \\ &= \boldsymbol{H}_1 \times (\mathsf{Diag}(\boldsymbol{a}) \otimes I_\ell)\end{aligned}$$

where $\boldsymbol{H}_1 = \boldsymbol{H}_0 \times \big(I_{n_0} \otimes \mathsf{Diag}(1, \beta^m, \beta^{2m} \ldots, \beta^{(\ell-1)m})\big)$. In other words, given a (quasi-circulant) generator matrix $\tilde{G}$ of $\tilde{\mathscr{C}}$, we would like to find $\theta$ in $\mathcal{S}_{N_0}$ and $\gamma_1, \ldots, \gamma_{n_0}$ in $\mathbb{T}_\ell$ such that the following equation holds:

$$\boldsymbol{H}_1 \times \Big(\Big(\mathsf{Diag}(\boldsymbol{a}) \times P_\theta\Big) \otimes I_\ell\Big) \times \begin{pmatrix} P_{\gamma_1} & & 0 \\ & \ddots & \\ 0 & & P_{\gamma_{n_0}} \end{pmatrix} \times \tilde{G} = 0. \tag{3}$$

For the sake of simplicity, we assume that $m$ is known and $P_{\gamma_i} = \boldsymbol{I}_\ell$. If we set $\Gamma = P_\theta \times \mathsf{Diag}(\boldsymbol{\lambda})$ then an attacker has to solve the linear system given in Equation (3) where the unknowns are the $N_0^2$ entries of $\Gamma$. It is important to note that there do exist a solution to Equation (3) that reveals both $\boldsymbol{a}$ and the set $J$. Each row of $\tilde{G}$ must satisfy $N - K = \ell(N_0 - K_0)$ parity equations over $\mathbb{F}_{q_0}$. But if we decompose each row of $\boldsymbol{H}_1$ according to a basis $\Omega$ of $\mathbb{F}_{q_0}$ over $\mathbb{F}_q$, we actually get $r(N-K) = r\ell(N_0 - K_0)$ parity equations. So the total number of equations is $r(N-K)k = r\ell^2(N_0 - K_0)k_0$ where we assume that $k = k_0\ell$. For instance if we consider the parameters $\mathsf{A}_{16}$ we obtain $594,441$ unknowns and $61,600$ equations.

An attacker has therefore to reduce the number of unknowns in order to guess $\Gamma$. However, with $r\ell^2(N_0 - K_0)k_0$ linear equations, an attacker can potentially guess any secret $\delta \times \delta$ matrix where $\delta = \ell\sqrt{r(N_0 - K_0)k_0}$. Another strategy is then to set $N_0 - \delta$ random columns of $\Gamma$ to zero and then to solve the resulting linear system. This method would give the right solution if only if the $n_0$ columns that intervene in the construction of the shortened alternant code are not set to zero. The success probability of this method is therefore $\frac{\binom{N_0 - n_0}{\delta - n_0}}{\binom{N_0}{\delta}}$. Additionally, for each random choice, one has to solve a linear system with a time complexity $O(s'^2\delta^3)$ with coefficients in $\mathbb{F}_{2^{s'}}$. In practice this attack does not give better results than direct decoding attack.

### 6.2 On the One-Wayness

We prove in this section that the primitive of our variant is also OW-CPA under the assumption that it is computationally impossible to distinguish a random quasi-cyclic code from an alternant quasi-cyclic code. We show that decoding an arbitrary quasi-cyclic code is also an NP-Hard problem. For

doing so, we propose another decisional problem called *quasi-cyclic syndrome decoding*. We prove that this new problem is also NP-Complete.

**Definition 13 (Quasi-cyclic syndrome decoding).** *Given $\ell > 1$ (we avoid the case $l = 1$ which corresponds to a degenerate case) matrices $A_1, \ldots, A_\ell$ of size $r^* \times n^*$ over $\mathbb{F}_q$, an integer $w < \ell n^*$ and a word $\boldsymbol{z}$ in $\mathbb{F}_q^{\ell r^*}$. Let $A$ be the $\ell r^* \times \ell n^*$ matrix:*

$$A = \begin{bmatrix} A_1 & \cdots & \cdots & A_\ell \\ A_\ell & A_1 & \cdots & A_{\ell-1} \\ \vdots & \ddots & \ddots & \vdots \\ A_2 & \cdots & A_\ell & A_1 \end{bmatrix}$$

*Does there exist $\boldsymbol{e}$ in $\mathbb{F}_q^{\ell n^*}$ of weight $\mathsf{wt}(\boldsymbol{e}) \leq w$ such that $A \times \boldsymbol{e}^T = \boldsymbol{z}$?*

**Proposition 1.** *The quasi-cyclic syndrome decoding problem is NP-Complete.*

*Proof.* We consider an instance $H$, $w$ and $\boldsymbol{z}$ of the syndrome decoding problem. We define $w^* = 2w$, the $2r$-tuple $\boldsymbol{z}^* = (\boldsymbol{z}, \boldsymbol{z})$ and the following $2r \times 2n$ matrix $A$:

$$A = \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix}.$$

Clearly $A$, $\boldsymbol{z}^*$ and $w^*$ are constructed in polynomial time. Assume now that there exist $\boldsymbol{e}$ in $\mathbb{F}_q^n$ of weight $\mathsf{wt}(\boldsymbol{e}) \leq w$ such that $H \times \boldsymbol{e}^T = \boldsymbol{z}$. Then $\mathsf{wt}(\boldsymbol{e}^*) \leq w^*$ and $A \times \boldsymbol{e}^{*T} = \boldsymbol{z}^*$.

Conversely assume that there exists $\boldsymbol{e}^*$ in $\mathbb{F}_q^{2n}$ of weight $\mathsf{wt}(\boldsymbol{e}^*) \leq w^*$ such that $A \times \boldsymbol{e}^{*T} = \boldsymbol{z}^*$. If we denote $\boldsymbol{e}^* = (\boldsymbol{e}_1, \boldsymbol{e}_2)$ where $\boldsymbol{e}_1$ is formed by the first $n$ symbols and $\boldsymbol{e}_2$ by the last $n$ symbols. Obviously we have either $\boldsymbol{e}_1$ or $\boldsymbol{e}_2$ of weight $\leq w^*/2$ and for both of them $H \times \boldsymbol{e}_j{}^T = \boldsymbol{z}$.

This result ensures that in the worse decoding random quasi-cyclic codes is a difficult problem. One may think that classical algorithms can be modified by taking into account the quasi-cyclic structure of the codes. This fact can be favorably used to indeed improve performances of generic algorithms, and one can reasonably expect to decrease by a factor the order of quasi-cyclicity $\ell$ the work factor of the general decoding algorithms. Note that the proposed parameters in Table 5.1 take into account this fact.

## 7 Conclusion

In this paper we presented a new way to reduce the size of the public key for code-based cryptosystems like McEliece or Niederreiter schemes by the use of quasi-cyclic alternant codes and very strongly punctured codes. The use of quasi-cyclic codes permits to reach a public key of as low as 6500 bits for a security of $2^{80}$ or 11,000 bits for $2^{107}$. We prove the NP-completeness of decoding quasi-cyclic codes. An implementation of our scheme ran at 120 Mb/s for encryption speed which makes it far better than RSA or NTRU cryptosystems (see [6] for comparisons). Such low parameters together with the high speed of the system open the doors to new potential applications for code-based cryptography in smart cards for instance, where such an algorithm can be used for key exchange or authentication.

## References


1. M. Baldi and G. F. Chiaraluce. Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In *IEEE International Symposium on Information Theory*, pages 2591–2595, Nice, France, March 2007.
2. T. P. Berger. Cyclic alternant codes induced by an automorphism of a GRS code. In R. Mullin and G. Mullen, editors, *Finite fields: Theory, Applications and Algorithms*, volume 225, pages 143–154, Waterloo, Canada, 1999. AMS, Contemporary Mathematics.
3. T. P. Berger. Goppa and related codes invariant under a prescribed permutation. *IEEE Trans. Inform. Theory*, 46(7):2628, 2000.
4. T. P. Berger. On the cyclicity of Goppa codes, parity-check subcodes of Goppa codes and extended Goppa codes. *Finite Fields and Applications*, 6:255–281, 2000.
5. E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *Information Theory, IEEE Transactions on*, 24(3):384–386, May 1978.
6. Bhaskar Biswas and Nicolas Sendrier. Mceliece cryptosystem implementation: theory and practice. *PQCrypto 2008 - Lecture Notes in Computer Science*, 2008.
7. A. Shokrollahi C. Monico, J. Rosenthal. Using low density parity check codes in the McEliece cryptosystem. In *IEEE International Symposium on Information Theory (ISIT 2000)*, page 215, Sorrento, Italy, 2000.
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. P.L. Cayrel, A. Otmani, and D. Vergnaud. On Kabatianskii-Krouk-Smeets Signatures. In *Proceedings of the first International Workshop on the Arithmetic of Finite Fields (WAIFI 2007)*, Springer Verlag Lecture Notes, pages 237–251, Madrid, Spain, June 21–22 2007.
10. D. Engelbert, R. Overbeck, and A. Schmidt. A summary of McEliece-type cryptosystems and their security. volume 1, pages 151–199, 2007.
11. P. Gaborit. Shorter keys for code based cryptography. In *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*, pages 81–91, Bergen, Norway, March 2005.
12. P. Gaborit and M. Girault. Lightweight code-based authentication and signature. In *IEEE International Symposium on Information Theory (ISIT 2007)*, pages 191–195, Nice, France, March 2007.
13. P. Gaborit, C. Lauradoux, and N. Sendrier. Synd: a fast code-based stream cipher with a security reduction. In *IEEE International Symposium on Information Theory (ISIT 2007)*, pages 186–190, Nice, France, March 2007.
14. P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In *Advances in Cryptology - EUROCRYPT'88*, volume 330/1988 of *Lecture Notes in Computer Science*, pages 275–280. Springer, 1988.
15. J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, 1988.
16. 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.
17. F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North–Holland, Amsterdam, fifth edition, 1986.
18. R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.
19. L. Minder and A. Shokrollahi. Cryptanalysis of the Sidelnikov cryptosystem. In *Eurocrypt 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 347–360, Barcelona, Spain, 2007.
20. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems Control Inform. Theory*, 15(2):159–166, 1986.
21. A. Otmani, J.P. Tillich, and L. Dallot. Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. preprint, 2008.
22. E. Petrank and R.M. Roth. Is code equivalence easy to decide? *Information Theory, IEEE Transactions on*, 43(5):1602–1604, Sep 1997.
23. V.S. Pless and W.C. Huffman, editors. *Handbook of coding theory*. North Holland, 1998.
24. N. Sendrier. Finding the permutation between equivalent linear codes: the support splitting algorithm. *Information Theory, IEEE Transactions on*, 46(4):1193–1203, Jul 2000.
25. V.M. Sidelnikov. A public-key cryptosystem based on binary Reed-Muller codes. *Discrete Mathematics and Applications*, 4(3), 1994.
26. V.M. Sidelnikov and S.O. Shestakov. On the insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 1(4):439–444, 1992.
27. J. Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988.
28. Christian Wieschebrink. Two NP-complete problems in coding theory with an application in code based cryptography. *Information Theory, 2006 IEEE International Symposium on*, pages 1733–1737, July 2006.