

Reducing Key Length of the McEliece Cryptosystem

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. Its main advantage is to have very fast encryption and decryption functions. However it suffers from a major drawback. It requires a very large public key which makes it very difficult to use in many practical situations. A possible solution is to advantageously use quasi-cyclic codes because they have a compact representation. In an other hand, for a fixed level of security, the use of optimal codes like Maximum Distance Separable ones allows to use smaller codes. The almost only known family of MDS codes with an efficient decoding algorithm is the class of Generalized Reed-Solomon (GRS) codes. However, it is well-known that GRS codes and quasi-cyclic codes do not represent secure solutions.

In this paper we propose a new general method to reduce the public key size by constructing quasi-cyclic Alternant codes over smaller fields.. We introduce a new method of hiding the structure of a quasi-cyclic GRS code. The idea is to start from a Reed-Solomon code in quasi-cyclic form defined over a large field. We then apply three transformations that preserve the quasi-cyclic feature. First, we randomly block shorten the RS code. Next, we transform it to get a Generalised Reed Solomon, and lastly we take the subfield subcode over a smaller field. We show that all existing structural attacks are infeasible. We also introduce a new decisional problem called quasi-cyclic syndrome decoding. We prove that it is NP-complete. This result suggest that decoding attacks against our variant have little chances to be better than the general ones against the classical McEliece cryptosystem. We propose a system with several size of parameters from 6,800 to 20,000 bits with a security ranging from 2^{80} to 2^{120} . 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 [19] 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 [21] also proposed a public-key cryptosystem based on linear codes in which the public key is a parity-check matrix. It is proved in [17] 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 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 [19] then

in [15, 16, 25, 6–8] and lastly in [3] 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 [2] 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 [24] that the structure of Generalised Reed-Solomon codes of length n can be recovered in $O(n^4)$. Sendrier proved that the (permutation) transformation can be extracted for concatenated codes. Minder and Shokrollahi presented in [20] a structural attack that creates a private key against a cryptosystem based on Reed-Muller codes [23]. 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 the OW-CPA criteria for $n \geq 2048$ with appropriate values for t and k such that $W_{2,n,k,t} \geq 2^{100}$ where $W_{q,n,k,t}$ is the work factor of the best algorithm that decodes t errors in a linear code of length n , dimension k over the finite field \mathbb{F}_q . For example (See [3]) the work factor is around 2^{128} if we choose $(n, k, t) = (2960, 2288, 56)$. For such parameters, the public key size is about 6.5 Mbits. It is therefore tempting to enhance the McEliece scheme 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 [5] 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 [12] tries to use quasi-cyclic codes [12, 1, 14, 13, 10]. This particular family of codes offers the advantage of having a very simple and compact description. The first proposal [12] 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 [22] that these two cryptosystems [12, 1] can be totally broken. The main drawbacks of [12] 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 developed in [12]. It makes use of quasi-cyclic code over a relatively small field (like $\mathbb{F}_{2^{10}}$ or \mathbb{F}_{2^8}) to reduce the size of the public keys. In particular it develops new ideas which permit to overcome the weaknesses of [12]. Our proposal is based on a general construction that starts from a family of Maximum Distance Separable (MDS) quasi-cyclic codes defined over a very large field (like $\mathbb{F}_{2^{16}}$ or $\mathbb{F}_{2^{20}}$). An excellent candidate is the family of Generalised Reed-Solomon codes. It represents an important class of MDS cyclic codes

equipped with an efficient decoding algorithm. The two main threats that may undermine the security of our variant are the attack of [24], which exploits the structure of generalised Reed-Solomon codes, and the attack of [22] which makes profit of the quasi-cyclic structure. Our first improvement over [12] 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 Sidelnikov-Shestakov [24] 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 attack [24]. The second improvement consists in resisting to the attack of [22] by randomly shortening a very long quasi-cyclic Alternant codes. For instance, one constructs codes of lengths of order 800 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 [27] 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 [22] that exploits the quasi-cyclic structure because it requires that 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.

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 cryptosystem to about thousands bits. The security of our new variant relies upon two assumptions. First, it is impossible to recover the secret shortened quasi-cyclic Alternant code. We prove that all the existing structural attacks require a prohibitive amount of computation. Assuming that is computationally impossible to structurally recover the secret code, the one-wayness under chosen plaintext attack is guaranteed by the hardness of decoding an *arbitrary quasi-cyclic* linear code. We prove in this paper 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 decodes any arbitrary quasi-cyclic code. This important assumption establishes the security of our variant.

The paper is organized as follows. In Section 2 we give useful notions of coding theory. We refer the reader to [18] for a detailed treatment of the coding theory. In Section 3.2 we recall basic facts about code-based cryptography (See [11] for more details). Section 4 deals with a description of our new variant. Sections 5 we provide different parameters for our scheme. In Section 6 we analyze the security of our scheme. Section 7 treats performances of our variant.

2 Coding Theory Background

2.1 Generalised Reed-Solomon Codes

Definition 1. *Let m be a positive integer and let q be a power of a prime number. Let α be a primitive element of the finite field \mathbb{F}_{q^m} and assume that $n = q^m - 1$. Let d and k be integers such that where $d = n - k + 1$.*

A code $\mathcal{R}_{n,k}$ of length n and dimension k over \mathbb{F}_{q^m} is a Reed-Solomon code if, up to a column permutation, it is defined by the parity-check matrix

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

Reed-Solomon codes represent an important class of Maximum Distance Separable (MDS) cyclic codes. It is well-known that d 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 2. Let $\boldsymbol{\lambda}$ be an n -tuple of nonzero elements in \mathbb{F}_{q^m} where $n = q^m - 1$. Let $\mathcal{R}_{n,k}$ be the Reed-Solomon code of length n and dimension k . The Generalised Reed-Solomon $\mathcal{G}_{n,k}(\boldsymbol{\lambda})$ code over \mathbb{F}_{q^m} of length n and dimension k is the set:

$$\mathcal{G}_{n,k}(\boldsymbol{\lambda}) = \left\{ (\lambda_1 v_1, \dots, \lambda_n v_n) \mid (v_1, \dots, v_n) \in \mathcal{R}_{n,k} \right\}. \quad (2)$$

Remark 1. Generalised Reed-Solomon codes are decoded in the same way as Reed-Solomon codes by means of the classical Berlekamp-Massey algorithm.

2.2 Subfield Subcode Construction

We describe an operation that takes as input a code over \mathbb{F}_{q^m} and outputs a linear code over \mathbb{F}_q . In our applications $q = 2^r$ for a positive integer $r \geq 1$.

Definition 3. Let \mathcal{C} be a code over \mathbb{F}_{q^m} . The subfield subcode $\tilde{\mathcal{C}}$ of \mathcal{C} over \mathbb{F}_q is the vector space $\mathcal{C} \cap \mathbb{F}_q^n$.

The construction of a subfield subcode from a code \mathcal{C} is easy by using the dual code. Indeed, it is well-known that the dual of $\tilde{\mathcal{C}}$ is the trace of the dual of \mathcal{C} [18]. Concretely this means that from a basis of \mathbb{F}_{q^m} over \mathbb{F}_q , at each element $a \in \mathbb{F}_{q^m}$, we can associate the column vector composed by its coordinates and denoted by $[a]$. From an $(n - k) \times n$ generator matrix $\mathbf{H} = (h_{i,j})$ of the dual of \mathcal{C} over \mathbb{F}_{q^m} , we can construct an $m(n - k) \times n$ matrix $\tilde{\mathbf{H}}$ over \mathbb{F}_q . The rows of $\tilde{\mathbf{H}}$ form a system of generators for the dual of $\tilde{\mathcal{C}}$. Note that $\tilde{\mathbf{H}}$ is not necessary of full rank and must be reduced by a Gaussian elimination.

Example 1. Let us consider $q = 2$, $m = 2$ and $(1, \alpha)$ a basis of \mathbb{F}_4 over \mathbb{F}_2 with $\alpha^2 = \alpha + 1$. Set $\mathbf{H} = \begin{pmatrix} 1 & \alpha & \alpha^2 & 1 \\ 0 & \alpha^2 & 1 & \alpha \end{pmatrix}$. Since $[0] = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $[1] = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $[\alpha] = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $[\alpha^2] = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, we obtain

$$\tilde{\mathbf{H}} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}. \text{ After a row elimination, a generator matrix of } \mathcal{C}'^\perp \text{ is } \mathbf{H}' = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

The subfield subcode construction is a very important tool to obtain a new class of codes. 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. Note that Goppa codes are subfield subcodes of generalised Reed-Solomon codes. They are also called *Alternant* codes.

Definition 4 (Alternant code). Let m be a positive and let \mathbb{F}_q be the field with q elements. Let $\mathcal{G}_{n,k}(\boldsymbol{\lambda})$ be a Generalised Reed-Solomon code of dimension k over \mathbb{F}_{q^m} . The Alternant code $\mathcal{A}_n(\boldsymbol{\lambda})$ over \mathbb{F}_q is the subfield subcode of $\mathcal{G}_{n,k}(\boldsymbol{\lambda})$ over \mathbb{F}_q .

Remark 2. Recall that the dimension of $\mathcal{A}_n(\boldsymbol{\lambda})$ is $\geq n - m(n - k)$.

2.3 Quasi-cyclic Codes

Definition 5. Let $N = \ell N_0$. The quasi-cyclic permutation σ_ℓ on $\{0, \dots, N-1\}$ of order ℓ (and index N_0) is the permutation defined by the orbits $\{(0, \dots, \ell-1), (\ell, \dots, 2\ell-1), \dots, ((N_0-1)\ell, \dots, N-1)\}$.

Definition 6. A linear code \mathcal{C} of length $N = \ell N_0$ is a quasi-cyclic code of order ℓ (and index N_0) if it is globally invariant under the action of σ_ℓ .

Note that, up to a change of order, this definition is equivalent to the more classical invariance under σ^ℓ , where σ is the cyclic shift permutation defined by $\sigma(\mathbf{v}) = (v_{N-1}, v_0, \dots, v_{N-2})$ for any vector $\mathbf{v} \in \mathbb{F}^N$ where \mathbb{F} is a finite field. However this definition corresponds to the description of quasi-cyclic codes with block circulant matrices which leads to a compact description (See [12] for more details).

Example 2. Let \mathbf{M} be the following 6×15 binary matrix:

$$\mathbf{M} = \begin{pmatrix} 100 & 000 & 101 & 110 & 011 \\ 010 & 000 & 110 & 011 & 101 \\ 001 & 000 & 011 & 101 & 110 \\ 000 & 100 & 111 & 101 & 000 \\ 000 & 010 & 111 & 110 & 000 \\ 000 & 001 & 111 & 011 & 000 \end{pmatrix}.$$

It is a generator matrix of a $[15, 6]$ quasi-cyclic code of order 3 (as it can be easily checked by its block circulant matrices form). It is completely described by the first row of each block:

$$\begin{bmatrix} [100] & [000] & [101] & [110] & [011] \\ [000] & [100] & [111] & [101] & [000] \end{bmatrix}.$$

Moreover, using the systematic form of the generator matrix, this description can be reduced to its redundancy part:

$$\begin{bmatrix} [101] & [110] & [011] \\ [111] & [101] & [000] \end{bmatrix}.$$

Remark 3. Since the dual of a quasi-cyclic code is also a quasi-cyclic code, it is possible to define a quasi-cyclic code through block circulant parity-check matrices.

Remark 4. A cyclic code is obviously quasi-cyclic of any order.

2.4 Reed-Solomon Codes in Quasi-Cyclic Form

We have seen that Reed-Solomon codes are cyclic codes. We now propose to reorder the support in order to obtain quasi-cyclic Reed-Solomon codes. Let \mathbb{F}_q the finite field where q is power of a prime number p (actually $p = 2$ and $q = 2^r$). Let m be an positive integer. Let us define α as a

primitive element of \mathbb{F}_{q^m} and set $N = q^m - 1$. Assume that $N = N_0\ell$ and define $\beta = \alpha^{N_0}$. Note that β is of order ℓ . We denote by β_ℓ the ℓ -tuple $(1, \beta, \dots, \beta^{\ell-1})$. Let t be a positive integer and let $\mathbf{U}_{2t} = (\mathbf{A}_0 \mid \dots \mid \mathbf{A}_{N_0-1})$ be the following block parity-check matrix where for $0 \leq j \leq N_0 - 1$:

$$\mathbf{A}_j = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha^j & \alpha^j \beta & \dots & \alpha^j \beta^{\ell-1} \\ \vdots & \vdots & & \vdots \\ (\alpha^j)^{2t-1} & (\alpha^j \beta)^{2t-1} & \dots & (\alpha^j \beta^{\ell-1})^{2t-1} \end{pmatrix}. \quad (3)$$

Proposition 1. *The code defined by \mathbf{U}_{2t} is a Reed-Solomon code $\mathcal{R}_{N,K}$ with $N - K = 2t + 1$ that is quasi-cyclic of order ℓ .*

3 Code-Based Cryptography

3.1 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^κ , where $\kappa \geq 0$ is a security parameter, outputs a pair (pk, sk) of public/private key. The **KeyGen** also specifies a finite *message space* M_{pk} . The **Encrypt** is a probabilistic polynomial-time algorithm that on inputs 1^κ , pk and a word \mathbf{x} in M_{pk} outputs a word \mathbf{c} . The decryption is a deterministic polynomial-time algorithm that on inputs 1^κ , sk and a word \mathbf{c} outputs a word \mathbf{x} . The cryptosystem should satisfy the *correctness* property which means that the decryption must undo the encryption.

3.2 McEliece Cryptosystem

Algorithm 1: McEliece.KeyGen(1^κ)

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

The McEliece cryptosystem [19] utilizes error-correcting codes that have an efficient decoding algorithm in order to build trapdoor one-way functions. 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 [18]. The principle of the McEliece cryptosystem is to randomly pick a code \mathcal{C} among the family of binary Goppa codes. The private key is the Goppa polynomial of \mathcal{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. In the rest of the paper we denote by $W_{q,n,k,t}$ the work factor of the best attack against the McEliece cryptosystem when the code is of length n , dimension k over \mathbb{F}_q and the number of added errors is t .

Algorithm 2: McEliece.Encrypt(pk, $\mathbf{m} \in \mathbb{F}_2^k$)

- 1 Randomly pick \mathbf{e} in \mathbb{F}_2 of weight t
 - 2 Calculate $\mathbf{c} = \mathbf{m} \times \mathbf{G} + \mathbf{e}$
 - 3 Output \mathbf{c}
-

Algorithm 3: McEliece.Decrypt(sk, $\mathbf{c} \in \mathbb{F}_2^n$)

- 1 Calculate $\mathbf{z} = \mathbf{c} \times \mathbf{P}^{-1}$
 - 2 Calculate $\mathbf{y} = \gamma(\mathbf{z})$
 - 3 Output $\mathbf{m} = \mathbf{y} \times \mathbf{S}^{-1}$
-

3.3 Niederreiter Cryptosystem

Algorithm 4: Niederreiter.KeyGen(1^κ)

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

A dual encryption scheme is the Niederreiter cryptosystem [21] which is equivalent in terms of security [17] 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 .

3.4 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*.

Algorithm 5: Niederreiter.Encrypt(pk, $\mathbf{m} \in \mathcal{W}_{2,n,t}$)

- 1 Calculate $\mathbf{c} = \mathbf{H} \times \mathbf{m}^T$
 - 2 Output \mathbf{c}
-

Algorithm 6: Niederreiter.Decrypt(sk, $\mathbf{c} \in \mathbb{F}_2^{n-k}$)

- 1 Calculate $\mathbf{z} = \mathbf{S}^{-1} \times \mathbf{c}$
 - 2 Calculate $\mathbf{y} = \gamma(\mathbf{z})$
 - 3 Output $\mathbf{m} = \mathbf{y} \times \mathbf{P}$
-

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.

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 in coding theory. It was proved NP-Complete in [2].

Definition 7 (Syndrome decoding). *Given an $r \times n$ matrix H over \mathbb{F}_q , a positive integer $w \leq n$ and an r -tuple \mathbf{z} in \mathbb{F}_q^r , does there exist \mathbf{e} in \mathbb{F}_q^n such that $\text{wt}(\mathbf{e}) \leq w$ and $H \times \mathbf{e}^T = \mathbf{z}$?*

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 \mathbf{G} and a given ciphertext \mathbf{z} . We know that there exist a codeword \mathbf{c} in \mathcal{C} and a vector $\mathbf{e} \in \mathbb{F}_q^n$ of weight t such that $\mathbf{z} = \mathbf{c} + \mathbf{e}$. By hypothesis, the minimum distance of \mathcal{C} is $2t + 1$. Therefore the linear code $\tilde{\mathcal{C}}$ defined by the generator matrix $\tilde{\mathbf{G}} = \begin{bmatrix} \mathbf{G} \\ \mathbf{c} \end{bmatrix}$ contains a codeword of weight t namely \mathbf{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.

The best practical algorithms for searching low weight codewords of any linear code are all derived from the *Information Set Decoding*. An information set for a generator matrix \mathbf{G} of a code of dimension k is a set J of k positions such that the restriction \mathbf{G}_J of \mathbf{G} over J is invertible. The redundancy part is then the complementary set. The algorithm looks for a codeword of weight w . The principle is to randomly pick a permutation matrix \mathbf{P} and a matrix \mathbf{S} such that $\mathbf{G}' = \mathbf{SGP} = (\mathbf{I}_k | \mathbf{B})$ is a row reduced echelon form. It then computes all the sums of g rows or less (where g is small) of \mathbf{G}' and stops as soon as a codeword of weight w is obtained. Otherwise the algorithm picks another permutation matrix \mathbf{P} and continues again. Lee and Brickell [15] were the first to use to McEliece cryptosystem. Leon [16] proposed to look only for codewords that are zero over a window of size σ in the redundancy part. Stern [25] improved the algorithm by dividing the information set into parts and by looking for codewords that are identical over a window of size σ in the redundancy part. Another improvement proposed by van Tilburg in [26] consists in not choosing a completely new information set J at each new iteration but rather permuting only a small number c of columns of \mathbf{G}' and keeping intact the $(k - c)$ other columns of the information set. This permits to decrease the amount of computation of the row-reduction operation. Canteaut and

Chabaud gathered all these refinements with $c = 1$ (also suggested in [6] and [7]) and studied the complexity of the algorithm in [8]. This last attempt represented the most effective attack against the McEliece cryptosystem (in the case of *binary* linear codes [9]). Recently, Bernstein *et al.* in [3] further enhanced the attack to a level that makes it practically possible to break the original parameters of the McEliece cryptosystem.

It is worthwhile remarking that the improvements of [8] and [3] are essentially applied to binary codes. In particular both analyze the complexity of the algorithms specifically in the binary case. It would be interesting to study the behavior of [3] when non-binary codes are considered.

4 A New Variant of the McEliece Cryptosystem

The new variant of the McEliece (or Niederreiter) cryptosystem we propose is not based on classical binary Goppa codes but rather on shortened quasi-cyclic Alternant codes. For a cryptographic purpose, we need a large family of codes with the following requirement: a description as compact as possible, a secret efficient decoding algorithm (i.e. a hidden structure) and a resistance to known (and unknowns...) attacks. As seen previously, quasi-cyclic codes offer the advantage of having a very compact representation which can lead to shorter keys. In an other hand, for a fixed level of security, the use of optimal codes (in particular, large minimum distance for fixed length and dimension) allows to use smaller codes. The so-called Maximum Distance Separable (MDS) codes are good candidates. The almost only known family of MDS codes with an efficient decoding algorithm is the class of Generalized Reed-Solomon (GRS) codes. However, it is well-known (see eg [24]) that GRS codes cannot be used directly in cryptography.

The aim is to obtain a family of quasi-cyclic Alternant code defined over relatively small field \mathbb{F}_q (like $\mathbb{F}_{2^{10}}$ or \mathbb{F}_{2^5}). The idea is to start from a Reed-Solomon code in quasi-cyclic form defined over a large alphabet \mathbb{F}_{q^m} (for instance $\mathbb{F}_{2^{20}}$) in quasi-cyclic form. Then we randomly delete the majority of circulant blocks to counter known attacks against the recovery of the quasi-cyclic structure. We transform this code into a quasi-cyclic shortened Generalized Reed Solomon code. Finally, we construct the subfield subcode over an intermediate field \mathbb{F}_q to mask the GRS structure. Recall that subfield subcodes of GRS codes constitute the family of Alternant codes. Note that the strategy of subfield subcodes in order to mask the structure of GRS code is not new since the class of Goppa codes used in the original McEliece cryptosystem is a subclass of Alternant codes. In our case, we use a large subfield in order to increase the performance of our codes.

4.1 Construction of a Family of Quasi-Cyclic Alternant Codes

We keep the notation of Section 2.4 throughout this section. Let t be a positive integer and let $\mathcal{R}_{N,K}$ be a Reed-Solomon in quasi-cyclic form over \mathbb{F}_{q^m} of length $N = \ell N_0$ and such that $N - K = 2t + 1$. The idea is to start from a Reed-Solomon code $\mathcal{R}_{N,K}$ in quasi-cyclic form defined over \mathbb{F}_{q^m} and successively applying three operations that preserve the quasi-cyclic feature, namely (1) randomly block shortening the code in order to obtain a code of length $n = n_0 \ell$ with $n_0 < N_0$, (2) transforming it to get a Generalised Reed Solomon code of length n and (3) taking the subfield subcode over \mathbb{F}_q .

In term of security the first operation (1) adds a combinatorial complexity by hiding our code into a larger code. Thus recovering the correct blocks requires to search for a relatively small number n_0 (say n_0 is greater than 6) of blocks, among a relatively large number N_0 of blocks (say 1000 to 100.000 blocks) which makes it very difficult in practice. This operation permits to resist to a recent cryptanalysis on quasi-cyclic codes by the combinatorial masking of the dual code. Operation (2) is a classical algebraic choice of blocks multiplication or cyclic permutations which increases the number of possible generalised Reed-Solomon. At last operation (3) scrambles the structure of the

code. In particular it permits to resist to the Sidelnikov-Shestakov attack which can be used on generalised Reed-Solomon codes. But it is inefficient on subfield subcodes even if they are derived from generalised Reed-Solomon codes. Notice that we could also have considered another hiding procedure by considering a subcode as in [12]. But this operation would increase the size of the key by a factor 2 we omitted it here. We present now the details of our strategy.

(1) Random Block Shortening. This step consists in randomly choosing n_0 (with $n_0 < N_0$) circulant blocks of the parity-check matrix \mathbf{U}_{2t} . More precisely, let $\mathbf{j} = (j_1, \dots, j_{n_0})$ be an n_0 -tuple of different non-negative integers less than or equal to N_0 . Note that we do not have necessarily that $j_1 \leq j_2 \leq \dots \leq j_{n_0}$. We then consider the code $\mathcal{R}_{n,K}(\mathbf{j})$ of length $n = \ell n_0$ over \mathbb{F}_{q^m} defined by the following parity-check matrix $\mathbf{U}_{2t}(\mathbf{j}) = (\mathbf{A}_{j_1} \mid \mathbf{A}_{j_2} \mid \dots \mid \mathbf{A}_{j_{n_0}})$.

Proposition 2. $\mathcal{R}_{n,K}(\mathbf{j})$ is a quasi-cyclic code of order ℓ .

Remark 5. One can easily check that random block shortening consists in randomly block permuting and puncturing the parity-check matrix \mathbf{U}_{2t} .

(2) GRS Transformation. The random block shortened Reed-Solomon $\mathcal{R}_{n,K}(\mathbf{j})$ code is transformed to obtain a quasi-cyclic Generalised Reed-Solomon code. Let $\mathbf{D}_{\beta_\ell} = (d_{i,j})$ be the $\ell \times \ell$ diagonal matrix such $d_{i,i} = \beta^{i-1}$. For any integer s , we have that:

$$\mathbf{D}_{\beta_\ell}^s = \begin{pmatrix} 1 & & & \\ & \beta^s & & \\ & & \ddots & \\ & & & (\beta^s)^{\ell-1} \end{pmatrix}. \quad (4)$$

We consider an n_0 -tuple $\mathbf{a} = (a_1, \dots, a_{n_0})$ of nonzero elements of \mathbb{F}_{q^m} and an integer s with $1 \leq s \leq \ell - 1$. Let $\mathcal{R}_{n,K}(\mathbf{j}, \mathbf{a}, s)$ be the code defined by the block parity-check matrix $\mathbf{U}_{2t}(\mathbf{j}, \mathbf{a}, s) = (\mathbf{B}_1 \mid \dots \mid \mathbf{B}_{n_0})$ where $\mathbf{B}_i = a_i \mathbf{A}_{j_i} \times \mathbf{D}_{\beta_\ell}^s$ for $1 \leq i \leq n_0$, or equivalently:

$$\mathbf{B}_i = \begin{pmatrix} a_i & a_i \beta^s & \dots & a_i (\beta^s)^{\ell-1} \\ a_i \alpha^{j_i} & a_i \beta^s \alpha^{j_i} \beta & \dots & a_i (\beta^s)^{\ell-1} \alpha^{j_i} \beta^{\ell-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_i (\alpha^{j_i})^{2t-1} & a_i \beta^s (\alpha^{j_i} \beta)^{2t-1} & \dots & a_i (\beta^s)^{\ell-1} (\alpha^{j_i} \beta^{\ell-1})^{2t-1} \end{pmatrix}. \quad (5)$$

Proposition 3. $\mathcal{R}_{n,K}(\mathbf{j}, \mathbf{a}, s)$ is a quasi-cyclic code of order ℓ .

Proof. It is easy to see that if we apply the quasi-cyclic permutation σ_ℓ to the i -th row \mathbf{u}_i of $\mathbf{U}_{2t}(\mathbf{j}, \mathbf{a}, s)$ then we have $\sigma_\ell(\mathbf{u}_i) = (\beta^{\ell-1})^{i+s} \mathbf{u}_i$, which proves that $\mathcal{R}_{n,K}(\mathbf{j}, \mathbf{a}, s)$ is globally invariant under the action of σ_ℓ .

(3) Subfield Subcode Operation. We then consider the subfield subcode over \mathbb{F}_q of $\mathcal{R}_{n,K}(\mathbf{j}, \mathbf{a}, s)$. We denote it by $\mathcal{A}_n(\mathbf{j}, \mathbf{a}, s)$ and is defined by a block $2tm \times n$ parity-check matrix $\tilde{\mathbf{H}}$ that is the trace matrix of $\mathbf{U}_{2t}(\mathbf{j}, \mathbf{a}, s)$ (See Section 2.2 for more details).

4.2 Description of the New Variant

We have seen in the previous section how to get a family of quasi-cyclic Alternant codes over \mathbb{F}_q from a unique reed-Solomon code $\mathcal{R}_{N,K}$ over \mathbb{F}_{q^m} . It is easy to see that the number of possible codes is equal to $n_0! \binom{N_0}{n_0} \times (q^m - 1)^{n_0} \times (\ell - 1)$. We now describe the KeyGen algorithm of our variant for a security parameter κ . We assume that q and m are given as input to the algorithm. In particular N_0 and ℓ are also known.

In Step 6, we further hide the structure of our codes by applying column permutations as it is done in the original McEliece cryptosystem. However we need to take a special kind of permutations that preserve the quasi-cyclic feature. This can be achieved by permuting columns inside each circulant block by means of a power σ^i of the cyclic shift σ of $\{0, \dots, \ell - 1\}$. Therefore we randomly pick an n_0 -tuple $\mathbf{i} = (i_1, \dots, i_{n_0})$ of non-negative integer smaller that ℓ . Note that \mathbf{M}^π is the result of a column permutation π over a matrix \mathbf{M} .

The goal of Step 7 is to choose a compact representation. This can be done by transforming \mathbf{H}' into a $\delta\ell \times n$ block circulant matrix in systematic form \mathbf{H} where δ is the smallest integer such that $\delta\ell \geq 2mt$. This means that there exists then $\delta\ell \times \delta\ell$ matrix \mathbf{S} such that $\mathbf{H} = \mathbf{S} \times \mathbf{H}'$. \mathbf{H} is then completely described by $(n_0 - \delta)\delta$ vectors of \mathbb{F}_q^ℓ , that is to say $(n_0 - \delta)\delta\ell r$ bits (recall that $q = 2^r$). We refer to [12] for a more detailed algorithm.

Algorithm 7: KeyGen(1^κ)

- 1 Choose t and n_0 such that $W_{2,n,k,t} \geq 2^\kappa$ where $n = \ell n_0$ and $k = n - 2mt$
 - 2 Randomly pick an n_0 -tuple $\mathbf{j} = (j_1, \dots, j_{n_0})$ of distinct non-negative integers $\leq N_0 - 1$.
 - 3 Randomly pick an n_0 -tuple \mathbf{a} of non zero elements of \mathbb{F}_{q^m} .
 - 4 Let α be a primitive element of \mathbb{F}_{q^m} and set $\beta = \alpha^{N_0}$. Randomly pick an integer $1 \leq s \leq \ell - 1$.
 - 5 Let $\tilde{\mathbf{H}} = (\tilde{\mathbf{B}}_1 \mid \dots \mid \tilde{\mathbf{B}}_{n_0})$ be the trace matrix of $U_{2t}(\mathbf{j}, \mathbf{a}, s)$.
 - 6 Randomly pick an n_0 -tuple $\mathbf{i} = (i_1, \dots, i_{n_0})$ of non-negative integer smaller that ℓ . Compute $\mathbf{H}' = (\tilde{\mathbf{B}}_1^{\sigma^{i_1}} \mid \dots \mid \tilde{\mathbf{B}}_{n_0}^{\sigma^{i_{n_0}}})$.
 - 7 Transform \mathbf{H}' into a block circulant matrix in systematic form \mathbf{H} .
 - 8 Output $\text{pk} = (\mathbf{H}, t)$ and $\text{sk} = (\mathbf{j}, \mathbf{a}, s, \mathbf{i})$
-

5 Suggested Parameters

We illustrate our construction with different sets of parameters in Table 1. Notice that the parameters n and k corresponds to the generator matrix G , the size of the public is computed from a matrix G or dual matrix H in systematic form, knowing that it is possible to use a systematic form for encryption if a hash function is also used (cf N. Sendrier's habilitation document).

The table also sums up the security entry that is computed from the Stern's algorithm applied over *non-binary* field. We give parameters with security from 2^{80} to 2^{120} with for instance public key sizes ranging from 6,500 bits for a 2^{80} security to 20,000 bits for a 2^{120} security. These parameters show the adaptability and the scalability of our system with a key size that moderately increases according to the security level. The security is taken to be the best attack between structural attack and decoding attack, but in all cases we considered the structural have a greater complexity than decoding attacks.

Example 3. 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 $m = 2$. We take a GRS

cyclic code C_0 with $t = 50$, which gives $N = 2^{16} - 1$ and $K = 2^{16} - 1 - 2 \times 50$, hence we get $m(N - K) = 100 \sim 2 \cdot \ell$ with $\delta = 4$ (the number of row eventually needed). Then we take $n_0 = 9$ and keep only 9 blocks of size ℓ among the 1285 possible blocks, this operation (1) of section 4. We then obtain a quasi-cyclic GRS $[9.51, 9.51 - 100, t = 50]_{2^{16}}$ code of order 51. We then apply linear transformations (2) and (3) of section 4 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 $[9.51, 9.51 - 200, t = 25]_{2^8} = [459, 255, t = 50]_{2^8}$ code, we can check from decoding attack algorithms that the security (taking account of the quasi-cyclic order) is 2^{80} and the size of the public key in systematic form on the dual is 8,100 bits.

Table 1. Suggested parameters for different security levels

$q^m - 1 = N = N_0 \ell$				Public code $\mathcal{C}[n, k]$ over \mathbb{F}_q							
q^m	ℓ	N_0	t	Code	n	k	q	n_0	δ	Security	Public key size (bits)
2^{16}	51	1,285	50	A_{16}	459	255	2^8	9	4	80	8,100
	51	1,285	50	B_{16}	510	306	2^8	10	4	90	9,700
	51	1,285	50	C_{16}	612	408	2^8	12	4	100	13,000
	51	1,285	50	D_{16}	765	510	2^8	15	5	120	20,000
2^{20}	75	13,981	56	A_{20}	450	225	2^{10}	6	3	80	6,800
	93	11,275	63	B_{20}	558	279	2^{10}	6	3	90	8,500
	93	11,275	54	C_{20}	744	372	2^{10}	8	4	110	14,600

6 Security Analysis

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 $\mathbf{sk} = (\mathbf{j}, \mathbf{a}, s, \mathbf{i})$ where \mathbf{j} is an n_0 -tuple of distinct non-negative integers $\leq N_0 - 1$, $\mathbf{a} = (a_1, \dots, a_{n_0})$ is an n_0 -tuple of non zero elements of \mathbb{F}_{q^m} , s is a non-negative integer $\leq \ell - 1$, and \mathbf{i} is an n_0 -tuple of non negative integer $\leq \ell - 1$. Note that a_1 can be chosen to be equal to 1. Thus the private key space contains $n_0! \binom{N_0}{n_0} \times (q^m - 1)^{n_0 - 1} \times (\ell - 1)^{n_0}$ elements. This implies that the private key is out of reach by an exhaustive search.

Recall also that the public parity-check matrix \mathbf{H} is obtained by putting $\mathbf{H}' = (\mathbf{B}'_1 \mid \dots \mid \mathbf{B}'_{n_0})$ in systematic form. There exists therefore a $\delta \ell \times \delta \ell$ matrix \mathbf{S} such that $\mathbf{H} = \mathbf{S} \times \mathbf{H}'$. A possible strategy is to guess the δ first blocks $\mathbf{\Delta} = (\mathbf{B}'_1 \mid \dots \mid \mathbf{B}'_\delta)$. This matrix $\mathbf{\Delta}$ is reduced in row echelon form. This achieved by left multiplying $\mathbf{\Delta}$ with a matrix $\mathbf{\Gamma}$. If one guesses the correct blocks then $\mathbf{\Gamma} = \mathbf{S}$. This can be checked if there exists at least a column \mathbf{c} of \mathbf{H}_{2t} and an nonzero element a in \mathbb{F}_{q^m} such that the column vector $a\mathbf{\Gamma} \times \mathbf{c}$ appears in the public matrix. If it is not the case, one choose another guess for $\mathbf{\Delta}$. The number of candidates for $\mathbf{\Delta}$ is $\delta! \binom{N_0}{\delta} (q^m - 1)^{\delta - 1} \ell^{\delta + 1}$. The cost to transform $\mathbf{\Delta}$ in row-reduced echelon form is $O(mr(\delta \ell)^3)$. Since \mathbf{H}_{2t} has N columns, the overall complexity of the attack is then $O\left(\delta! \binom{N_0}{\delta} (q^m - 1)^{\delta - 1} \ell^{\delta + 1} (mr(\delta \ell)^3 + N(q^m - 1)mr(\delta \ell)^2)\right)$.

Attack exploiting the generalised Reed-Solomon structure. Sidelnikov and Shestakov proved in [24] that it is possible to completely recover the structure of Generalised Reed-Solomon codes

with time complexity in $O(n^4)$ 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 (random 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 [22] that extracts the private key of the variant presented in [12]. 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 the parity-check matrix U_{2t} of the Reed-Solomon code $\mathcal{R}_{K,N}$ in quasi-cyclic form as defined in Section 2.4. We consider the matrix $\mathbf{G} = (\mathbf{G}_p \mid \mathbf{0})$ where \mathbf{G}_p is a generator matrix of the public code of length n and $\mathbf{0}$ is a zero matrix with $N - n$ columns. Clearly, there exists an $N \times N$ matrix \mathbf{X} such that:

$$U_{2t} \times \mathbf{X} \times \mathbf{G}^T = \mathbf{0}. \quad (6)$$

where \mathbf{G}^T is the transpose of \mathbf{G} . The matrix \mathbf{X} has a block structure where each block is either the $\ell \times \ell$ zero matrix or a matrix of the form $a\mathbf{D}_{\beta_\ell}^s$ where a is a nonzero element of \mathbb{F}_{q^m} and s is an integer smaller than ℓ . The matrix \mathbf{X} actually gathers all the secret operations that have been made to obtain the public quasi-cyclic Alternant code. Thus the integer s is the *same* for any nonzero block whereas a may vary. Therefore solving the linear system given by Equation (6) reveals the secret key. Note that \mathbf{X} is thus totally determined by N_0^2 matrices of size $\ell \times \ell$. Additionally, as s is very small, one may assume that $\mathbf{D}_{\beta_\ell}^s$ is known. Therefore the number of unknowns to totally describe \mathbf{X} is *exactly* N_0^2 . On the other hand, Equation 6 provides $(n - 2mt) \times 2t$ equations (each row of \mathbf{G} gives $2t$ check equations from U_{2t}). But if we consider the trace matrix U'_{2t} instead of U_{2t} , then m times more parity equations for each row of \mathbf{G} . We can get a total of $2mt(n - 2mt)$ linear equations for N_0^2 unknowns.

An attacker has therefore to reduce the number of unknowns in order to guess \mathbf{X} . Another strategy is to set $N_0 - \tilde{N}$ random block columns of \mathbf{X} to zero where $\tilde{N}^2 = 2mt(n - 2mt)$. 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 $\frac{\binom{N_0 - n_0}{\tilde{N} - n_0}}{\binom{N_0}{\tilde{N}}}$. Additionally, for each random choice, one has to solve a linear system with a time complexity $O((rm)^2 \tilde{N}^3)$ with coefficients in $\mathbb{F}_{2^{rm}}$. 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 recover the secret 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 8 (Quasi-cyclic syndrome decoding). *Given $\ell > 1$ (we avoid the case $\ell = 1$ which corresponds to a degenerate case) matrices $\mathbf{A}_1, \dots, \mathbf{A}_\ell$ of size $r^* \times n^*$ over \mathbb{F}_q , an integer $w < \ell n^*$ and a word \mathbf{z} in $\mathbb{F}_q^{\ell n^*}$. Let \mathbf{A} be the $\ell r^* \times \ell n^*$ matrix:*

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_\ell \\ \mathbf{A}_\ell & \mathbf{A}_1 & \cdots & \mathbf{A}_{\ell-1} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{A}_2 & \cdots & \mathbf{A}_\ell & \mathbf{A}_1 \end{bmatrix}$$

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

Proposition 4. *The quasi-cyclic syndrome decoding problem is NP-Complete.*

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

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

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

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

This result ensures that (in the worse case) decoding an arbitrary quasi-cyclic code is a difficult task. One may think that classical algorithms can be modified by taking into account the quasi-cyclic structure. 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 ℓ the work factor of the general decoding algorithms. Note that the proposed parameters in Table 5 take into account this fact. However our proof shows that if ever an algorithm appears that efficiently solves the quasi-cyclic syndrome decoding problem, then it will be give another efficient one for the more general problem of syndrome decoding. This result would represent a major breakthrough in the field of coding theory. This result also suggest that decoding attacks against our variant have little chances to be better than the general ones against the classical McEliece cryptosystem.

Discussion on the practical difficulty of syndrome decoding for QC codes

In fact up to now, although decoding a general random QC codes may seem easier than decoding a random code with no such quasi-cyclic structure, the best known algorithm remains the general algorithms for random codes (up to the order of quasi-cyclicity which is a small factor). The situation is in some sense very similar to the case of lattice based cryptography and the LLL algorithm. For instance the NTRU lattice has a similar quasi-cyclic structure but no attack related to the LLL algorithm did really took advantage of this structure except by a very small improvement of a constant, although the NTRU cryptosystem has been known for more than 10 years and was really scrutinized. For coding theory the same situation seems to occur and general random QC codes seem to be as hard to decode as general random codes (up to the size of the order of quasi-cyclicity).

7 Performance

We implemented the encryption with the system A_{20} on \mathbb{F}_{2^8} on 2.4 Ghz computer with a 64-bit architecture. The multiplication over \mathbb{F}_{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 cycles per octet.

This speed compares well with the implementation of [4] 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 (\mathbb{F}_{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 [4] although we did not yet optimized our implementation.

Our implementation speed compares of course very well with RSA-1024 and Elliptic Curve cryptosystems. 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 [4]. 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).

8 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. We use quasi-cyclic Alternant codes over a non-binary small field. We introduced new methods to hide the structure of quasi-cyclic Alternant codes. We showed that all structural attack cannot cope with our parameters. We prove that decoding quasi-cyclic codes is an NP-Hard problem. This result makes decoding attack inappropriate. Our scheme permits to reach a public key size as low as 6,500 bits for a security of 2^{80} and 20,000 bits for 2^{120} . Lastly, an implementation of our scheme ran at 120 Mb/s for encryption speed which makes it far better than RSA or NTRU cryptosystems (see [4] for comparisons). Such low parameters together with the high speed of the system open the doors to new potential applications for code-based cryptography like smart cards, 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. 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.
3. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the mceliece cryptosystem. In *PQCrypto*, pages 31–46, 2008.
4. Bhaskar Biswas and Nicolas Sendrier. Mceliece cryptosystem implementation: theory and practice. *PQCrypto 2008 - Lecture Notes in Computer Science*, 2008.
5. 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.

6. A. Canteaut and H. Chabanne. A further improvement of the work factor in an attempt at breaking McEliece's cryptosystem. In *EUROCODE 94*, pages 169–173. INRIA, 1994.
7. A. Canteaut and F. Chabaud. Improvements of the attacks on cryptosystems based on error-correcting codes. Technical Report 95–21, INRIA, 1995.
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. A. Canteaut and N. Sendrier. Cryptanalysis of the original McEliece cryptosystem. In *Advances in Cryptology - ASIACRYPT'98*, number 1514 in LNCS, pages 187–199. Springer-Verlag, 1998.
10. 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.
11. D. Engelbert, R. Overbeck, and A. Schmidt. A summary of McEliece-type cryptosystems and their security. volume 1, pages 151–199, 2007.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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.
18. F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, fifth edition, 1986.
19. R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.
20. 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.
21. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems Control Inform. Theory*, 15(2):159–166, 1986.
22. A. Otmani, J.P. Tillich, and L. Dallot. Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. preprint, 2008.
23. V.M. Sidelnikov. A public-key cryptosystem based on binary Reed-Muller codes. *Discrete Mathematics and Applications*, 4(3), 1994.
24. 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.
25. 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.
26. Johan van Tilburg. On the mceliece public-key cryptosystem. In *CRYPTO '88: Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*, pages 119–131, London, UK, 1990. Springer-Verlag.
27. 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.