Compact McEliece Keys from Goppa Codes

Rafael Misoczki[1], Paulo S. L. M. Barreto[1⋆]

Departamento de Engenharia de Computação e Sistemas Digitais (PCS),
Escola Politécnica, Universidade de São Paulo, Brazil.
{rmisoczki,pbarreto}@larc.usp.br

Abstract. The classical McEliece cryptosystem is built upon the class of Goppa codes, which remains secure to this date in contrast to many other families of codes but leads to very large public keys. Recently, Berger et al. proposed a method to construct short McEliece keys by replacing Goppa codes by quasi-cyclic, block-shortened alternant codes. In this paper we describe a simple alternative way to reduce significantly the size of McEliece keys using Goppa codes, and to improve the efficiency of cryptographic operations to subquadratic time.

Keywords: post-quantum cryptography, syndrome decoding, efficient algorithms.

1 Introduction

Quantum computers can potentially break most if not all conventional cryptosystems actually deployed in practice, namely, all systems based on the integer factorization problem (like RSA) or the discrete logarithm problem (like traditional or elliptic curve Diffie-Hellman and DSA, and also all of pairing-based cryptography).

Certain classical cryptosystems based on NP-complete or NP-hard problems remain largely unaffected by the threat of quantum computing, and have thus been called quantum-resistant or, more suggestively, 'post-quantum' cryptosystems. These include lattice-based cryptosystems and syndrome-based cryptosystems like McEliece [12] and Niederreiter [15]. Such systems usually have even a speed advantage over conventional schemes; for instance, McEliece or Niederreiter encryption over a code of length n has time complexity $O(n^2)$, while RSA and DSA with n-bit keys have time complexity $O(n^3)$. On the other hand, they are plagued by very large keys compared to their conventional counterparts.

It is therefore of utmost importance to seek ways to reduce the key sizes for post-quantum cryptosystems while keeping their security level. The first steps toward this goal were taken by Monico et al. using low density parity-check codes [14], by Gaborit using quasi-cyclic codes [8], and by Baldi and Chiaraluce using a combination of both [1]. However, these proposals were all shown to contain weaknesses [17].

In a recent breakthrough, Berger et al. [2] showed how to circumvent the drawbacks of Gaborit's original scheme by means of two techniques:

1. Extracting block-shortened public codes from very large private codes, adapting an idea proposed by Wieschebrink [22];
2. Working with subfield subcodes over an intermediate subfield between the base field and the extension field of the original code.

⋆ Supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under research productivity grant 312005/2006-7 and universal grant 485317/2007-9, and by the Science Foundation Ireland (SFI) as E. T. S. Walton Award fellow under grant 07/W.1/I1824.

These two techniques were successfully applied to quasi-cyclic codes, yet we will see that their applicability is not restricted to that class.

Our contribution in this paper is a new method to generate compact keys for efficient syndrome-based cryptosystems using a subclass of Goppa codes, as a simple and viable alternative to cyclic and quasi-cyclic codes. The keys our method produces are up to a factor t smaller than keys produced from generic t-error correcting binary Goppa codes, while retaining the capability of actually correcting the designed number of errors (for other classes of alternant codes, no decoding algorithm is known capable of correcting more than $t/2$ errors). Due to the structure of public keys, the complexity of all operations become no more than $O(n^{\lg 3})$; specifically, code generation (excluding conversion to systematic form) has complexity $O(n \lg n)$, while systematic formatting, encryption and decryption take no more than $O(n^{\lg 3})$ time.

The remainder of this paper is organized as follows. Section 2 introduces some basic concepts of syndrome-decoding based cryptography. In section 3 we describe our proposal of using binary Goppa codes in dyadic form, and how to build them. We consider security issues in Section 4, and efficiency issues, including guidelines on how to choose parameters, in Section 5. We conclude in Section 6.

2 Preliminaries

Definition 1. *Given $z = (z_0, \ldots, z_{t-1}) \in \mathbb{F}_q^t$ and $L = (L_0, \ldots, L_{n-1}) \in \mathbb{F}_q^n$, such that $z_i \neq z_j$, $L_k \neq L_\ell$, and $z_i \neq L_k$ for all $0 \leqslant i \neq j < t$ and $0 \leqslant k \neq \ell < n$, the* Cauchy matrix *$C(z, L)$ is the $t \times n$ matrix with elements $C_{ij} = 1/(z_i - L_j)$, i.e.*

$$C(z,L) = \begin{bmatrix} \frac{1}{z_0 - L_0} & \frac{1}{z_0 - L_1} & \cdots & \frac{1}{z_0 - L_{n-1}} \\ \frac{1}{z_1 - L_0} & \frac{1}{z_1 - L_1} & \cdots & \frac{1}{z_1 - L_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{z_{t-1} - L_0} & \frac{1}{z_{t-1} - L_1} & \cdots & \frac{1}{z_{t-1} - L_{n-1}} \end{bmatrix}.$$

Cauchy matrices have the property that all of their submatrices are nonsingular [20].

Definition 2. *Given $h = (h_0, \ldots, h_{n-1}) \in \mathbb{F}_q^n$, the* dyadic *matrix $\Delta(h)$ is the matrix with components $\Delta_{ij} = h_{i \oplus j}$ where $\oplus$ stands for bitwise exclusive-or. The sequence h is called its* signature*. Given $t > 0$, $\Delta(t, h)$ denotes $\Delta(h)$ truncated to its first t rows. A* quasi-dyadic *matrix is either a dyadic matrix whose signature consists of (possibly non-dyadic) submatrices rather than field elements, or equivalently a (possibly non-dyadic) matrix whose elements are dyadic submatrices. A* dyadic permutation *is a dyadic matrix π^i characterized by the signature $\pi^i_j = \delta_{ij}$, where δ_{ij} is the Kronecker delta.*

Dyadic matrices form a subgroup as they are associated to the ring of dyadic convolution polynomials [10]. Notice that, in general, Cauchy matrices are not dyadic and vice-versa, although the intersection of these two classes is non-empty in characteristic 2.

Definition 3. *Given $t > 0$ and a sequence $L = (L_0, \ldots, L_{n-1}) \in \mathbb{F}_q^n$ of distinct elements, the* Vandermonde matrix *$V(t, L)$ is the $t \times n$ matrix with elements $V_{ij} = L_j^i$.*

Definition 4. *Given a sequence $L = (L_0, \ldots, L_{n-1}) \in \mathbb{F}_q^n$ of distinct elements and a sequence $D = (D_0, \ldots, D_{n-1}) \in \mathbb{F}_q^n$ of nonzero elements, the* Generalized Reed-Solomon code $GRS_d(L, D)$ *is an $[n, k, d]$ linear error-correcting code defined by the parity-check matrix*

$$H = V(d-1, L) \cdot \mathrm{diag}(D).$$

The classical Reed-Solomon code itself is $GRS_d(\boldsymbol{\alpha}, \boldsymbol{\alpha})$ for $\boldsymbol{\alpha} = (1, \alpha, \alpha^2, \ldots, \alpha^{n-1})$ and some primitive element $\alpha \in \mathbb{F}_q$. Its parity-check matrix can also be written as $H = V^{\mathsf{T}}(n - 1, (\alpha, \alpha^2, \ldots, \alpha^{d-1}))$.

Definition 5. *An* alternant code *of length n and dimension k over $\mathbb{F}_q$ for $q = p^m$ is a subfield subcode of a Generalized Reed-Solomon code over $\mathbb{F}_p$.*

Subfield subcodes are easily derived from a given code by means of the trace construction:

Definition 6. *The* trace construction *produces an $\mathbb{F}_p$-subfield subcode of a given code over $\mathbb{F}_{p^m} = \mathbb{F}_p[x]/r(x)$ (for some irreducible polynomial $r(x) \in \mathbb{F}_p[x]$ of degree m) by writing the $\mathbb{F}_p$ coefficients of each $\mathbb{F}_{p^m}$ component of a parity-check matrix $H_{t\times n}$ of the original code onto m successive rows of a parity-check matrix $H'_{mt\times n}$ of the subcode. The related* co-trace *parity-check matrix $H''_{mt\times n}$, equivalent to H' by a left permutation, is obtained from H by writing the $\mathbb{F}_p$ coefficients of terms of equal degree from all components on a column of H onto successive rows of H''.*

Thus, given $\mathbb{F}_{p^m}$ elements $u_i(x) = u_{i,0} + \cdots + u_{i,m-1}x^{m-1}$, the trace construction maps a column $(u_0, \ldots, u_{t-1})^{\mathsf{T}}$ from H to the column $(u_{0,0}, \ldots, u_{0,m-1}; \ldots; u_{t-1,0}, \ldots, u_{t-1,m-1})^{\mathsf{T}}$ on the trace matrix H', and to the column $(u_{0,0}, \ldots, u_{t-1,0}; \ldots; u_{0,m-1}, \ldots, u_{t-1,m-1})^{\mathsf{T}}$ on the co-trace matrix H''.

Finally, one of the most important families of linear error-correcting codes for cryptographic purposes is that of Goppa codes:

Definition 7. *Given $q = p^m$, a sequence $L = (L_0, \ldots, L_{n-1}) \in \mathbb{F}_q^n$ of distinct elements and a polynomial $g(x) \in \mathbb{F}_q[x]$ such that $g(L_i) \neq 0$ for $0 \leqslant i < n$, the* Goppa code *is the alternant code over $\mathbb{F}_p$ defined by the parity-check matrix $H = GRS_d(L, D)$ where $D = (g(L_0)^{-1}, \ldots, g(L_{n-1})^{-1})$.*

A binary irreducible Goppa code with design distance $2t + 1$ can correct up to t bits using Patterson's algorithm [18], or slightly more using Bernstein's list decoding method [4], and t errors can still be corrected by suitable decoding algorithms if the binary generator $g(x)$ is not irreducible[1]. In all other cases no more than $t/2$ errors can in general be corrected.

3 Goppa codes in Cauchy and dyadic form

A property of Goppa codes that is central to our proposal is that they admit a parity-check matrix in Cauchy form:

Theorem 1 ([21]). *The Goppa code defined by a polynomial $g(x) = (x - z_0) \ldots (x - z_{t-1})$ without multiple zeros admits a parity-check matrix of the form $H = C(z, L)$, i.e. $H_{ij} = 1/(z_i - L_j)$, $0 \leqslant i < t$, $0 \leqslant j < n$.*

[1] For instance, one can equivalently view the Goppa code as the alternant code defined by the generator polynomial $g^2(x)$, in which case any generic alternant decoder will decode t errors. We are grateful to Nicolas Sendrier for pointing this out.

This theorem (also appearing in [11, Ch. 12, §3, Pr. 5]) is completely general when one considers the factorization of the Goppa polynomial over its split field, in which case a single root of g is enough to completely characterize the code. For simplicity, we will restrict our attention to the case where all zeroes of that polynomial are in the field $\mathbb{F}_q$ itself.

The McEliece cryptosystem cannot be securely defined on a Goppa code specified directly by a parity-check matrix in Cauchy form, since this would immediately reveal the Goppa polynomial $g(x)$: it suffices to solve the overdefined linear system $z_i - L_j = 1/H_{ij}$ consisting of tn equations in $t+n$ unknowns. Actually this system has rank $t+n-1$, but they are all equivalent in the sense that, for any fixed $\omega \in \mathbb{F}_q$, the polynomial defined by the roots $z'_i = z_i + \omega$ on the support $L'_j = L_j + \omega$ leads to the same parity-check matrix $H_{ij} = 1/(z'_i - L'_j)$ and hence to the same syndromes, decodable using any of those codes.

3.1 Building a binary Goppa code in dyadic form

We now show how to build a binary Goppa code that admits a parity-check matrix in dyadic form. We do so by constructing a sequence $h = (h_0, \ldots, h_{n-1}) \in \mathbb{F}_q^n$, with $q = 2^m$ for some m, of distinct nonzero values such that

$$\frac{1}{h_{i\oplus j}} = \frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0},$$

choosing an element $\omega \in \mathbb{F}_q$, and then defining the Goppa code $\Gamma(L, g)$ by assembling the polynomial g from its zeros $z_i = 1/h_i + \omega$ and the code support $L_j = 1/h_j + 1/h_0 + \omega$. Clearly the parity-check matrix in Cauchy form $H_{ij} = 1/(z_i - L_j)$ is actually a dyadic matrix $H_{ij} = h_{i\oplus j}$.

The technique we propose consists of simply choosing distinct nonzero h_0 and h_i at random where i scans all powers of two smaller than n, and setting all other values as

$$h_{i+j} \leftarrow \frac{1}{\dfrac{1}{h_i} + \dfrac{1}{h_j} + \dfrac{1}{h_0}}$$

for $0 < j < i$ (so that $i + j = i \oplus j$), as long as this value is well-defined. Algorithm 1, whose running time is $O(n \lg n)$ steps, captures this idea.

To complete the construction it is necessary to choose a compact generator matrix for the subfield subcode. Although the parity check matrix H built by Algorithm 1 is dyadic over $\mathbb{F}_q$, the usual trace construction leads to a generator of the dual code that most probably violates the dyadic symmetry. However, by representing each field element to a basis of $\mathbb{F}_q$ over the subfield $\mathbb{F}_p$, one can view H as a superposition of $m = [\mathbb{F}_q : \mathbb{F}_p]$ distinct dyadic matrices over $\mathbb{F}_p$, and each of them can be stored in a separate dyadic signature.

Theorem 2. *Algorithm 1 produces up to* $\prod_{i=0}^{\lceil \lg n \rceil} (q - 2^i)$ *Goppa codes in dyadic form.*

Proof. Each dyadic signature produced by Algorithm 1 is entirely determined by the values h_0 and h_{2^s} for $s = 0, \ldots, \lceil \lg n \rceil - 1$ chosen at steps 2 and 6 (ω only produces equivalent codes). Along the loop at line 4, exactly $2i = 2^{s+1}$ elements are erased from U, corresponding to the choices of $h_{2^s} \ldots h_{2^{s+1}-1}$. At the end of that loop, $2 + 2\sum_{\ell=0}^{s} 2^\ell = 2^{s+2}$ elements have been erased in total. Hence at the beginning of each step of the loop only 2^{s+1} elements had been erased from U, i.e. there are $q - 2^{s+1}$ elements in U to choose h_{2^s} from, and $q - 1$ possibilities for h_0. Therefore this construction potentially yields up to $(q-1)\prod_{s=0}^{\lceil \lg n \rceil - 1} (q - 2^{s+1}) = \prod_{i=0}^{\lceil \lg n \rceil} (q - 2^i)$ possible codes. □

Algorithm 1 Constructing a binary Goppa code in dyadic form

INPUT: q (a power of 2), $n \leqslant q/2$, t.

OUTPUT: Support L, generator polynomial g and dyadic parity-check matrix H for a binary Goppa code $\Gamma(L, g)$ of length n and design distance $2t + 1$ over $\mathbb{F}_q$.

$U \leftarrow \mathbb{F}_q \setminus \{0\}$

▷ Choose the dyadic signature $(h_0, \ldots, h_{n-1})$. N.B. Whenever h_j with $j > 0$ is taken from U, so is $1/(1/h_j + 1/h_0)$ to prevent a potential spurious intersection between z and L.

$h_0 \xleftarrow{\$} U$

$U \leftarrow U \setminus \{h_0\}$

for $s \leftarrow 0$ **to** $\lceil \lg n \rceil - 1$ **do**

    $i \leftarrow 2^s$

    $h_i \xleftarrow{\$} U$

    $U \leftarrow U \setminus \{h_i,\, 1/(1/h_i + 1/h_0)\}$

    for $j \leftarrow 1$ **to** $i - 1$ **do**

        $h_{i+j} \leftarrow 1/(1/h_i + 1/h_j + 1/h_0)$

        $U \leftarrow U \setminus \{h_{i+j},\, 1/(1/h_{i+j} + 1/h_0)\}$

    end for

end for

$\omega \xleftarrow{\$} \mathbb{F}_q$

▷ Assemble the Goppa polynomial:

for $i \leftarrow 0$ **to** $t - 1$ **do**

    $z_i \leftarrow 1/h_i + \omega$

end for

$g(x) \leftarrow \prod_{i=0}^{t-1} (x - z_i)$

▷ Compute the support:

for $j \leftarrow 0$ **to** $n - 1$ **do**

    $L_j \leftarrow 1/h_j + 1/h_0 + \omega$

end for

$H \leftarrow \Delta(t, (h_0, \ldots, h_{n-1}))$

return L, g, H

The actual number of possible Goppa codes that can be put in dyadic form (independently of how they are constructed) is currently unknown. Also, we will see that the number of permuted subfield subcodes that we describe next and propose for cryptographic applications is even larger.

3.2 Constructing quasi-dyadic, permuted subfield subcodes

Algorithm 1 generates fully dyadic codes. We now show how to integrate the techniques of Berger et al. with Algorithm 1 so as to build quasi-dyadic subfield subcodes whose parity-check matrix is a non-dyadic matrix composed of blocks of dyadic submatrices. The principle to follow here is to *select, permute, and scale* the columns of the original parity-check matrix so as to preserve quasi-dyadicity in the target subfield subcode and the distribution of introduced errors in cryptosystems. A similar process yields a generator matrix in convenient quasi-dyadic, systematic form.

For the desired security level (see the discussion in Section 5.1), choose $r = 2^m$ for some m, $q = r^d$ for some d, a code length n and a design number of correctable errors t such that $n = \ell t$ for some $\ell > d$. For simplicity we assume that t is a power of 2, but the following construction method can be modified to work with other values.

Run Algorithm 1 to produce a code over $\mathbb{F}_q$ whose length $N \gg n$ is a large multiple of t not exceeding the largest possible length $q/2$, so that the constructed parity-check matrix $H_{t\times N}$ can be viewed as a sequence of N/t dyadic blocks $[B_0 \mid \cdots \mid B_{N/t-1}]$ of size $t \times t$ each. Select ℓ distinct blocks $B_{i_0}, \ldots, B_{i_{\ell-1}}$ in any order from H together with ℓ random dyadic permutations $(\pi^{j_0}, \ldots, \pi^{j_{\ell-1}})$ of size $t \times t$ and ℓ nonzero scale factors $(\sigma_0, \ldots, \sigma_{\ell-1}) \in \mathbb{F}_r^\ell$. Let $H''_{dt\times \ell t}$ be the co-trace matrix over $\mathbb{F}_r$ constructed from $[\sigma_0 B_{i_0}\pi^{j_0} \mid \cdots \mid \sigma_{\ell-1}B_{i_{\ell-1}}\pi^{j_{\ell-1}}]$. By construction, H'' consists of d layers of ℓ dyadic submatrices each. In Section 4.1 we discuss the security properties of this quasi-dyadic structure of H''. The private information consisting of the signature h (which gives rise to the polynomial g and the support L), the sequence $(i_0, \ldots, i_{\ell-1})$ of blocks, the sequence $(j_0, \ldots, j_{\ell-1})$ of dyadic permutation identifiers, and the sequence of scale factors $(\sigma_0, \ldots, \sigma_{\ell-1})$, relates the public code defined by (the systematic form of) H'' with the private code defined by H. The space occupied by the private information is thus about $ndm + \ell \lg(N/t) + \ell \lg t + \ell m$ bits.

Writing H'' in systematic form involves a Gaussian elimination incurring some ℓ multiplications of such submatrices per layer to put each column of submatrices in echelon form; since there are d columns to process and d layers, the overall cost is about $d^2\ell$ products of dyadic $t \times t$ submatrices, implying a complexity $O(d^2\ell t^{\lg 3})$. For typical cryptosystems one has $t = O(n/\lg n)$ and hence the cost of formatting is $O(d^2 n^{\lg 3}/(\lg n)^{\lg 3-1})$. In the worst case where $\mathbb{F}_r$ is the base field itself, one has $d = O(\lg n)$ and the complexity deteriorates to $O(n^{\lg 3}(\lg n)^{3-\lg 3})$, but since maximum size reduction is achieved when $\mathbb{F}_p$ is a large proper subfield of $\mathbb{F}_q$ (see below), in practice d is a small constant, and the complexity of systematic formatting stays at the $O(n^{\lg 3})$ level.

The resulting parity-check matrix defines a code of length n and dimension $k = n - dt$, and still consists of dyadic submatrices which can be represented by a signature of length t, and hence the whole matrix can be stored in an area a factor t smaller than a general matrix. During the process the dyadic submatrices are not guaranteed to be nonsingular as they are not associated to a Cauchy matrix any longer; should all the submatrices on a column be found to be singular during the Gaussian elimination (above or below the diagonal, according to the direction of this process) so that no pivot is possible, the whole column containing may be replaced by another block $B_{j'}$ chosen at random from H' as above.

The total space occupied by the essential part of the resulting generator (or parity-check) matrix over $\mathbb{F}_p$ is $dt \times (n - dt)/t = dk$ $\mathbb{F}_p$ elements, or mk bits – a factor t better than plain Goppa codes,

which occupy $k(n-k) = mkt$ bits. Had t not been chosen to be a power of 2, say, $t = 2^u v$ where $v > 1$ is odd, the cost of multiplying $t \times t$ matrices would be in general $O(v^3 3^u)$ rather than simply $O(3^u)$, and the final parity-check matrix would be compressed by only a factor 2^u.

For each code produced by Algorithm 1, the number of codes generated by this construction is $\binom{N/t}{\ell} \times \ell! \times t^\ell \times (r-1)^\ell$, hence $\binom{N/t}{\ell} \times \ell! \times t^\ell \times (r-1)^\ell \times \prod_{i=0}^{\lceil \lg N \rceil} (q - 2^i)$ codes are possible in principle.

As a final observation, notice that an $r \times n$ parity-check matrix consisting of $t \times t$ dyadic submatrices defines a code equivalent to another one defined by a $t \times t$ parity-check matrix consisting of $k \times n$ submatrices, the two matrices being related to each other by left and right permutations. We will make use of this observation when dealing with security issues in Section 4.1.

3.3 Decoding

While there are algorithms to efficiently decode a Goppa syndrome computed with a parity-check matrix H in alternant form, no such algorithm is known to directly decode a syndrome computed with a parity-check matrix Δ in systematic quasi-dyadic form.

However, knowledge of z, L and the overall scaled permutation P resulting from the process described in Section 3.2 is enough to build the easily decodable parity-check matrix H, which is linearly related to Δ as $H = S\Delta P$ for some nonsingular matrix S, which can be retrieved as $S = HP^\mathsf{T}\Delta^\mathsf{T}(\Delta\Delta^\mathsf{T})^{-1}$. Thus, given a syndrome $s = \Delta\, c^\mathsf{T}$ for some (corrupted) codeword c, compute $s' = Ss = Hc^\mathsf{T}$ and use this syndrome for decoding.

3.4 A toy example

Let $\mathbb{F}_{2^5} = \mathbb{F}_2[u]/(u^5 + u^2 + 1)$. The dyadic signature

$$h = (u^{20}, u^3, u^6, u^{28}, u^9, u^{29}, u^4, u^{22}, u^{12}, u^5, u^{10}, u^2, u^{24}, u^{26}, u^{25}, u^{15})$$

and the offset $\omega = u^{21}$ define a 2-error correcting binary Goppa code of length $N = 16$ with $g(x) = (x - u^{12})(x - u^{15})$ and support $L = (u^{21}, u^{29}, u^{19}, u^{26}, u^6, u^{16}, u^7, u^5, u^{25}, u^3, u^{11}, u^{28}, u^{27}, u^9, u^{22}, u^2)$. The associated parity-check matrix built according to Theorem 1 is

$$H' = \left[\begin{array}{cc|cc|cc|cc|cc|cc|cc|cc} u^{20} & u^3 & u^6 & u^{28} & u^9 & u^{29} & u^4 & u^{22} & u^{12} & u^5 & u^{10} & u^2 & u^{24} & u^{26} & u^{25} & u^{15} \\ u^3 & u^{20} & u^{28} & u^6 & u^{29} & u^9 & u^{22} & u^4 & u^5 & u^{12} & u^2 & u^{10} & u^{26} & u^{24} & u^{15} & u^{25} \end{array}\right],$$

with eight 2×2 blocks $B_0, \ldots, B_7$ as indicated. From this we extract the shortened, rearranged and permuted sequence $H'' = [B_7\pi^0 \mid B_5\pi^1 \mid B_1\pi^0 \mid B_2\pi^1 \mid B_3\pi^0 \mid B_6\pi^1 \mid B_4\pi^0]$ (for simplicity we choose all scale factors equal to 1), i.e.:

$$H'' = \left[\begin{array}{cc|cc|cc|cc|cc|cc|cc|cc} u^{25} & u^{15} & u^2 & u^{10} & u^6 & u^{28} & u^{29} & u^9 & u^4 & u^{22} & u^{26} & u^{24} & u^{12} & u^5 \\ u^{15} & u^{25} & u^{10} & u^2 & u^{28} & u^6 & u^9 & u^{29} & u^{22} & u^4 & u^{24} & u^{26} & u^5 & u^{12} \end{array}\right],$$

whose co-trace matrix over $\mathbb{F}_2$ has the systematic form:

$$H = \left[\begin{array}{cccc|cccccccccc} 0&1&0&1&1&0&0&0&0&0&0&0&0&0 \\ 1&0&1&0&0&1&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0&0&1&0&0&0&0 \\ 0&1&1&0&0&0&0&0&0&0&1&0&0&0 \\ 1&0&0&1&0&0&0&0&0&0&0&1&0&0 \\ 1&1&0&0&0&0&0&0&0&0&0&0&1&0 \\ 1&1&0&0&0&0&0&0&0&0&0&0&0&1 \end{array}\right] = [M^{\mathsf{T}} \mid I_{n-k}],$$

from which one readily obtains the $k \times n = 4 \times 14$ generator matrix in systematic form:

$$G = \left[\begin{array}{cccc|cccccccccc} 1&0&0&0&0&1&0&1&0&0&0&1&1&1 \\ 0&1&0&0&1&0&1&0&0&0&1&0&1&1 \\ 0&0&1&0&0&1&0&0&1&1&1&0&0&0 \\ 0&0&0&1&1&0&0&0&1&1&0&1&0&0 \end{array}\right] = [I_k \mid M],$$

where both G and H share the essential part M:

$$M = \begin{bmatrix} \mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{1} \\ 1&0&1&0&0&0&1&0&1&1 \\ \mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0} \\ 1&0&0&0&1&1&0&1&0&0 \end{bmatrix},$$

which is entirely specified by the elements in boldface and can thus be stored in 20 bits instead of, respectively, $4 \cdot 14 = 56$ and $10 \cdot 14 = 140$ bits.

4 Security considerations

The security of the original McEliece cryptosystem depends on the hardness of decoding permuted Goppa codes. Although our proposal is ultimately based on Goppa codes, one may wonder whether or not the highly composite nature of the Goppa polynomial $g(x)$ or the peculiar structure of the quasi-dyadic parity-check and generator matrices introduce any weakness.

Any alternant code can be written in Goppa-like fashion form by using the diagonal component of its default parity-check matrix to interpolate a generating polynomial (not necessarily of degree t) that is composite with high probability. We are not aware of any way this fact could be used to facilitate decoding without full knowledge of the code structure, and clearly any result in this direction would affect most of the alternant codes proposed for cryptographic purposes to date.

Otmani et al.'s attack against quasi-cyclic codes [17] could be modified to work against Goppa codes in dyadic form. For this reason we adopt the same countermeasure proposed by Berger et al. to thwart it for cyclic codes, namely, working with a block-shortened subcode of a very large code as described in Section 3.2. This idea also build upon the work of Wieschebrink [22] who proved that deciding whether a code is equivalent to a shortened code was proven NP-complete. In our case, the

result is to hide the Cauchy structure of the private code in a general dyadic structure, rather than disguising a quasi-cyclic code as another one with the same symmetry.

We now define a new computational problem whose hardness is related to the security of the quasi-dyadic matrices in our proposal, and we show that it is NP-complete.

4.1 A new NP-complete problem

The syndrome decoding problem (SDP) is classical in coding theory and known to be NP-complete [3]. Formally:

Definition 8 (Syndrome decoding). *Let $\mathbb{F}_q$ be a finite field, and let (H, w, z) be a triple consisting of a matrix $H \in \mathbb{F}_q^{r \times n}$, an integer $w < \ell n$, and a vector $z \in \mathbb{F}_q^r$. Does there exist a vector $e \in \mathbb{F}_q^n$ of Hamming weight $\mathsf{w}(e) \leqslant w$ such that $Ce^\mathsf{T} = z^\mathsf{T}$?*

One of the most elegant aspects of Berger et al.'s work [2] is a proof that the following problem, dubbed the quasi-cyclic syndrome decoding problem (QC-SDP) and to which the security of quasi-cyclic codes is related, is NP-complete:

Definition 9 (Quasi-cyclic syndrome decoding). *Let $\mathbb{F}_q$ be a finite field, and let (C, w, z) be a triple consisting of an $\ell \times \ell$ circulant block matrix C with elements $C_{ij} = M_{(j-i) \bmod \ell}$ for $\ell > 1$ matrices $M_0, \ldots, M_{\ell-1} \in \mathbb{F}_q^{r \times n}$, an integer $w < \ell n$, and a vector $z \in \mathbb{F}_q^{\ell r}$. Does there exist a vector $e \in \mathbb{F}_q^{\ell n}$ of Hamming weight $\mathsf{w}(e) \leqslant w$ such that $Ce^\mathsf{T} = z^\mathsf{T}$?*

Not surprisingly, one can formulate the corresponding problem for quasi-dyadic matrices, which are precisely the kind of parity-check (or generator) matrices produced by the method described in Section 3.2:

Definition 10 (Quasi-dyadic syndrome decoding). *Let $\mathbb{F}_q$ be a finite field, and let (Δ, w, z) be a triple consisting of an $\ell \times \ell$ quasi-dyadic matrix Δ with elements $\Delta_{ij} = M_{i \oplus j}$ for $\ell > 1$ matrices $M_0, \ldots, M_{\ell-1} \in \mathbb{F}_q^{r \times n}$, an integer $w < \ell n$, and a vector $z \in \mathbb{F}_q^{\ell r}$. Does there exist a vector $e \in \mathbb{F}_q^{\ell n}$ of Hamming weight $\mathsf{w}(e) \leqslant w$ such that $\Delta e^\mathsf{T} = z^\mathsf{T}$?*

Theorem 3. *The quasi-dyadic syndrome decoding problem (QD-SDP) is NP-complete.*

The proof is a straightforward extension of the proof for the QC-SDP and is given in appendix A. This theorem strengthens our claim that the family of codes we propose is secure for cryptographic applications.

5 Efficiency considerations

Due to their simple structure the matrices in our proposal can be held on a simple vector not only for long-term storage or transmission, but for processing as well.

The operation of multiplying a vector by a (quasi-)dyadic matrix is at the core of McEliece encryption. A Karatsuba-like approach for dyadic convolution[2] leads to the asymptotic complexity

[2] Regrettably the fast Walsh-Hadamard transform [10], which reduces the complexity of the dyadic convolution to $O(n \lg n)$, is only possible in fields of characteristic other than 2.

$O(n^{\lg 3})$ for this operation and hence also for encryption. Sarwate's decoding method [19] sets the asymptotic cost of decryption at roughly $O(n^{\lg 3})$ as well.

Inversion, on the other hand, can be carried out in $O(n)$ steps: one can show by induction that a binary dyadic matrix $\Delta(h)$ of dimension n satisfies $\Delta^{\mathsf{T}}\Delta = \Delta\Delta^{\mathsf{T}} = \Delta^2 = (\sum_i h_i)^2 I$, and hence its inverse, when it exists, is $\Delta^{-1} = (\sum_i h_i)^{-2}\Delta$, which can be computed in $O(n)$ steps since it is entirely determined by its first row.

The cost of converting a dyadic matrix to systematic (echelon) form is the same as the cost of a dyadic convolution: write the original matrix as $M_{k\times n} = [A_{k\times k} \mid B_{k\times(n-k)}]$, invert the square dyadic matrix A in linear time, and then compute the echelon form $[I_k \mid A^{-1}B]$, which involves the product of two dyadic matrices (one of them possibly truncated) and hence has the claimed $O(n^{\lg 3})$ overall complexity. If M is a parity-check matrix, write it as $M_{r\times n} = [A_{r\times(n-r)} \mid B_{r\times r}]$ and obtain the systematic form $[B^{-1}A \mid I_r]$. Notice that, contrary to systems based on quasi-circulant matrices [8, Proposition 3.4], our proposal does not require a lengthy process, involving expensive $O(n^3)$ matrix rank computations to construct a generator matrix in suitable form, often larger than one would expect for a code of the given dimension.

Table 1 summarizes the costs of key generation, systematic formatting, and encryption/decryption.

Table 1. Cryptographic operation complexity relative to the code length n.

operation	generic	ours
Key generation	$O(n^3)$	$O(n \lg n)$
Systematic form	$O(n^3)$	$O(n^{\lg 3})$
Encrypt/Decrypt	$O(n^2)$	$O(n^{\lg 3})$

5.1 Suggested parameters

Several trade-offs are possible when choosing parameters for a particular application. One may wish to minimize the key size, or increase speed, or simplify the underlying arithmetic, or attaining a balance between them. We present here some non-exhaustive combinations. The number of possible codes is large; for instance, it is about 2^{655} for an original code length of $N = 2^{15}$ even for the least secure example below, at the 80-bit level entry on table 3.

Table 2 shows the influence of varying the subfield degree while keeping the security level and the number of errors fixed. In general, codes over larger subfields allow for smaller keys as already indicated in [2].

Table 2. Sample parameters for 128-bit security, using a subcode over the subfield $\mathbb{F}_{2^s}$ of $\mathbb{F}_{2^{16}}$ with a fixed number of errors ($t = 32$).

s	n	k	key (bits)
8	1408	1280	20480
4	1856	1600	25600
2	3072	2560	40960
1	4864	3840	61440

Table 3 displays a different trade-off whereby the key size and the subfield are kept constant at the cost of varying the number of errors and the code length. The *level* column contains the target security level, while the *CC* column indicates the approximate computational cost of the Canteaut-Chabaud low-weight codeword attack [6] as a representative of the actual security of the suggested parameters; the method in [5] is more effective according to the authors, but its complexity is considerably more tricky to estimate and is nevertheless not too far from the values we list (a factor about $150 \approx 2^7$ more efficient). The number of errors is always a power of 2 to enable maximum reduction of the key size.

Table 3. Sample parameters for a fixed key size (8192 bits, corresponding to $k = 512$), using a subcode over the subfield $\mathbb{F}_{2^8}$ of $\mathbb{F}_{2^{16}}$.

n	t	CC
640	32	2^{90}
768	64	2^{122}
1024	128	2^{156}

Table 4 contains a variety of balanced parameters for practical security levels. The target security level is shown on column *level*, with a margin over the estimated actual cost of the Canteaut-Chabaud low-weight codeword attack (which is usually more effective than known alternatives) shown on column *CC*. The 'gain' column is the size ratio between a *generic* Goppa key and a quasi-dyadic Goppa key for roughly the same security level; to obtain it we compare the key sizes suggested in [5] with our proposal, and in both cases we take into account only the redundancy part of the key in systematic form (that it is safe to do so is proven in [7]). On the rightmost column we list the typical size of a (quantum-susceptible) RSA key at the specified security level (more accurate RSA security estimates can be found in [16]).

Table 4. Sample parameters for a subcode over the subfield $\mathbb{F}_{2^8}$ of $\mathbb{F}_{2^{16}}$.

level	CC	n	k	t	key size		gain	RSA
					ours	[5]		
72	2^{80}	512	256	64	4096	460647	112	1024
96	2^{104}	640	384	64	6144	1047600	170	1536
112	2^{122}	768	512	64	8192	–	–	2048
128	2^{138}	896	640	64	10240	1537536	150	3072
144	2^{152}	1024	768	64	12288	2093616	170	4096
192	2^{201}	1280	768	128	12288	4185415	340	7680
224	2^{238}	1536	1024	128	16384	–	–	11520
256	2^{268}	1792	1280	128	20480	7667855	374	15360

6 Conclusion and further research

We have described how to generate Goppa codes in quasi-dyadic form for cryptographic applications. Key sizes are roughly a factor $t = O(n/\lg n)$ smaller than generic Goppa codes, and keys can be kept in this compact size not only for storing and transmission but for processing as well, without

loosing the ability to correct the design number of errors, sometimes even more. This brings the size of cryptographic keys to within a factor 4 or less of equivalent RSA keys. Our work provides an alternative to conventional cyclic and quasi-cyclic codes, yet it benefits from the same security techniques proposed by Wieschebrink in general [22], and by Berger et al. for that family of codes [2].

The complexity of all cryptographic operations in the McEliece cryptosystem is reduced to no more than $O(n^{\lg 3})$. An obvious research problem is how to reduce the cost to $\tilde{O}(n)$ or to extend the proposed construction to higher characteristics for which $\tilde{O}(n)$ algorithms are immediately available. Another problem is that of finding the largest class of quasi-dyadic, efficiently decodable codes.

Other cryptosystems can also benefit from dyadic codes, e.g. entity identification and certain digital signatures for which double circulant codes have been proposed [9] could use dyadic codes instead, even random ones without a Goppa trapdoor.

Interestingly, it is equally possible to define *lattice*-based cryptosystems with short keys using dyadic lattices entirely analogous to ideal (cyclic) lattices as proposed by Micciancio [13], and achieving comparable size reduction. We leave this line of inquiry for future research since it falls outside the scope of this paper.

Acknowledgements

We are most grateful to Pierre-Louis Cayrel, Philippe Gaborit, Christiane Peters, and Nicolas Sendrier for their valuable comments during the preparation of this work.

References

1. M. Baldi and F. Chiaraluce. Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC code. In *IEEE International Symposium on Information Theory – ISIT'2007*, pages 2591–2595, Nice, France, 2007. IEEE.
2. T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani. Reducing key length of the McEliece cryptosystem. In *Progress in Cryptology – Africacrypt'2009*, Lecture Notes in Computer Science. Springer, 2009. To appear. Preliminary (2008) version at `http://www.unilim.fr/pages_perso/philippe.gaborit/reducing.pdf`.
3. E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
4. D. J. Bernstein. List decoding for binary Goppa codes. Preprint, 2008. `http://cr.yp.to/papers.html#goppalist`.
5. D. J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. In *Post-Quantum Cryptography Workshop – PQCrypto'2008*, volume 5299 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2008. `http://www.springerlink.com/content/68v69185x478p53g`.
6. A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: application to primitive narrow-sense BCH-codes of length 511. *IEEE Transactions on Information Theory*, 44:367–378, 1998.
7. D. Engelbert, R. Overbeck, and A. Schmidt. A summary of mceliece-type cryptosystems and their security. *Journal of Mathematical Cryptology*, 1:151–199, 2007.
8. P. Gaborit. Shorter keys for code based cryptography. In *International Workshop on Coding and Cryptography – WCC'2005*, pages 81–91, Bergen, Norway, 2005. ACM Press.
9. 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, 2007. IEEE.
10. M. N. Gulamhusein. Simple matrix-theory proof of the discrete dyadic convolution theorem. *Electronics Letters*, 9(10):238–239, 1973.
11. F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*, volume 16. North-Holland Mathematical Library, 1977.
12. R. McEliece. A public-key cryptosystem based on algebraic coding theory. The Deep Space Network Progress Report, DSN PR 42–44, 1978. `http://ipnpr.jpl.nasa.gov/progressreport2/42-44/44N.PDF`.

13. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.
14. C. Monico, J. Rosenthal, and A. Shokrollahi. Using low density parity check codes in the McEliece cryptosystem. In *IEEE International Symposium on Information Theory – ISIT'2000*, page 215, Sorrento, Italy, 2000. IEEE.
15. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.
16. European Network of Excellence in Cryptology. ECRYPT yearly report on algorithms and keysizes (2007-2008). D.SPA.28 Rev. 1.1, IST-2002-507932 ECRYPT, 07/2008, 2008. `http://www.ecrypt.eu.org/ecrypt1/documents/D.SPA.28-1.1.pdf`.
17. A. Otmani, J.-P. Tillich, and L. Dallot. Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. Preprint, 2008. `http://arxiv.org/abs/0804.0409v2`.
18. N. J. Patterson. The algebraic decoding of Goppa codes. *IEEE Transactions on Information Theory*, 21(2):203–207, 1975.
19. D. V. Sarwate. On the complexity of decoding Goppa codes. *IEEE Transactions on Information Theory*, 23:515–516, 1977.
20. S. Schechter. On the inversion of certain matrices. *Mathematical Tables and Other Aids to Computation*, 13(66):73–77, 1959. `http://www.jstor.org/stable/2001955`.
21. K. K. Tzeng and K. Zimmermann. On extending Goppa codes to cyclic codes. *IEEE Transactions on Information Theory*, 21:721–716, 1975.
22. C. Wieschebrink. Two NP-complete problems in coding theory with an application in code based cryptography. In *IEEE International Symposium on Information Theory – ISIT'2006*, pages 1733–1737, Seattle, USA, 2006. IEEE.

A Proof of the NP-completeness of the QD-SDP

Theorem 4. *The quasi-dyadic syndrome decoding problem (QD-SDP) is NP-complete.*

Proof. The QD-SDP is NP-complete for precisely the *same* reason that the quasi-cyclic syndrome decoding is, since a 2×2 circulant matrix is also dyadic. Thus we closely follow [2]. Consider an instance $(H_{r' \times n'}, w', z')$ of the SDP. Define $w = 2w'$, $z = (z', z')$, $r = 2r'$, $n = 2n'$, and the following quasi-dyadic matrix $\Delta_{r \times n}$:

$$\Delta = \begin{bmatrix} H & O \\ O & H \end{bmatrix}.$$

Clearly the instance (Δ, z, w) of the QD-SDP is constructed in polynomial time. Now suppose that there exists $e' \in \mathbb{F}_q^{n'}$ of Hamming weight $\mathsf{w}(e') \leqslant w'$ such that $He'^{\mathsf{T}} = z'^{\mathsf{T}}$, and let $e = (e', e')$. Then $\mathsf{w}(e) = 2\mathsf{w}(e') \leqslant 2w' = w$ and $\Delta e^{\mathsf{T}} = z^{\mathsf{T}}$. Conversely assume that there exists $e \in \mathbb{F}_q^n$ of Hamming weight $\mathsf{w}(e) \leqslant w$ such that $\Delta e^{\mathsf{T}} = z^{\mathsf{T}}$. Write $e = (e'_0, e'_1)$ as the concatenation of two n'-symbol vectors e'_0 and e'_1. Obviously at least one of e'_0 or e'_1 has Hamming weight not exceeding $w/2 = w'$, and either of them satisfies $He'^{\mathsf{T}}_i = z'^{\mathsf{T}}$. □