A Post-Quantum Digital Signature Scheme from QC-LDPC Codes

Alessio Meneghetti[*], Christian Picozzi[†], and Giovanni Tognolini[‡]

Abstract

We propose a novel post-quantum code-based digital signature algorithm whose security is based on the difficulty of decoding Quasi-Cyclic codes in systematic form, and whose trapdoor relies on the knowledge of a hidden Quasi-Cyclic Low-Density-Parity-Check (QC-LDPC) code. The utilization of Quasi-Cyclic (QC) codes allows us to balance between security and key size, while the LDPC property lighten the encoding complexity, thus the signing algorithm complexity, significantly.

1 Introduction

The *Syndrome Decoding Problem* (SDP) and the *Codeword Finding Problem* (CFP) are two important computationally-hard algebraic problems arising from Coding Theory, whose difficulty is underlying the security of several Post-Quantum Cryptographic schemes. A linear binary $[n, k]$ code C is a vector subspace of $\mathbb{F}_2^n$ of dimension k, where $(\mathbb{F}_2, +, \cdot)$ is the finite field with 2 elements. The values n and k are respectively called length and dimension of the code, and the elements of C are called codewords. In this work we adopt the well-established notation of considering codewords as row vectors, and thus any vector v is treated as a row vector so that its transpose $v^\top$ is a column vector. We consider $\mathbb{F}_2^n$ as a metric space together with the Hamming metric, i.e. two vectors u, v are at distance $\mathrm{d}(u, v)$ apart if they differ on exactly $\mathrm{d}(u, v)$ coordinates. A vector v has weight $w = \mathrm{w}(v)$ if the number of its non-zero coordinates is w. Notice that, since C is a vector subspace of $\mathbb{F}_2^n$, the zero-vector is always a codeword and, for any pair of codewords u and v, we have $\mathrm{d}(u, v) = \mathrm{w}(u + v)$. The minimum among the distances between codewords is a fundamental parameter of a code, called the minimum distance of the code and classically denoted with d. When the minimum distance d is known, the code is said to be an $[n, k, d]$ code. We remark that the problem of determining the minimum distance of a code is an NP-hard problem [36]. A $k \times n$ matrix whose rows form a basis for C is called a generator matrix, and an $(n - k) \times n$ generator matrix for the dual of C (i.e. the vector subspace $C^\perp$ of $\mathbb{F}_2^n$ whose elements are orthogonal to C) is called a parity-check matrix of C. Observe that a vector c is a codeword if and only if $cH^\top = 0$. In the general case, if we multiply a vector v by $H^\top$ we obtain a vector $s \in \mathbb{F}_2^{n-k}$ known as the syndrome of v. Notice that all vectors in the same coset $v + C$ possess the same syndrome.

The Syndrome Decoding Problem is the search version of the *Maximum Likelihood*

[*] alessio.meneghetti@unitn.it *Department of Mathematics, University of Trento*
[†] christianpicozzi98@gmail.com *Department of Mathematics, University of Trento*
[‡] giovanni.tognolini@unitn.it *Department of Mathematics, University of Trento*

Decoding Problem (MLD), which is the NP-complete problem [8] corresponding to the difficulty of decoding binary linear codes, and can be stated as follows:

Input: a positive integer w, a binary matrix H and a syndrome y.

Output: there exists a word x such that $\mathrm{w}(x) \leq w$ and $y^\top = Hx^\top$.

This problem has been proven to be in P/Poly by Bruck and Naor [10] and by Lobstein [24], namely, even by knowing in advance the parity-check matrix of a code the difficulty of decoding a received vector is still a hard problem. An interested reader can refer to [8] for the difficulty of decoding generic codes, to [29] for the link between MLD and the problem of solving quadratic Boolean polynomial systems, and to [11, 19, 20, 33] for specific classes of codes. The idea behind the SDP is that if we choose ω small enough, it is hard to find a vector with weight smaller than ω such that its syndrome is equal to y. On the other hand, the decisional CFP is as follows.

Input: a positive integer w and a binary matrix H.

Output: there exists a word x such that $\mathrm{w}(x) = w$ and $0 = Hx^\top$.

The idea behind the CFP is that if ω is small enough it is hard to find a codeword of the binary code C with parity-check matrix H such that its weight is ω, i.e. it is hard to find codeword with small weight in a given binary code.

Several post-quantum cryptosystems are build upon SDP (e.g. [1, 5, 9, 25, 27, 31]) or its variants obtained by adopting different metrics (e.g. the rank metric as in [26, 28]). In particular, the key-encapsulation mechanism Classic McEliece [9] is one of the three finalists in the NIST standardisation process for post-quantum primitives and its security is demonstrated by over forty years of cryptanalysis. Notably, no code-based primitive is present among the NIST finalists for post-quantum digital signature algorithms, even though some interesting proposals were submitted to the round 1 of the standardisation process [16, 17, 22]. These digital signature schemes, either due to security issues or efficiency problems, were not selected for the second round of the standardisation. Among code-based digital signature algorithms, the two most famous schemes are CFS [12, 14] and KKS [21]. The idea behind both schemes is the same: the digest of a message to be signed is considered as a corrupted codeword (more precisely, it is considered as a syndrome of a corrupted codeword) $c + e$ for a certain codeword $c \in C$ and a small-weight error e, and the owner of the private key is the only one capable of decoding, i.e. capable of finding e. The main difference between CFS and KKS is that in the first the digest is considered as a random syndrome, leading to a secure yet inefficient scheme, while the latter manipulates the digest to output a decodable syndrome, leading to a very efficient but insecure scheme [32]. In order to mitigate the efficiency issues of CFS, some authors have proposed CFS-like schemes in which the hash function is substituted with a map whose output is a syndrome with small-weight coset-leader, mixing some ideas behind both CFS and KKS. This approach has been adopted e.g. in [34], where the authors utilize a hash function based on the works of Augot, Finiasz and Sendrier [2] and on the Merkle-Damgard construction [15, 30]. This promising approach has however been proved to be insecure in [13].

In this document we propose a code-based digital signature based on different ideas, for which we need some preliminary results. Let us consider the quotient ring $R := \mathbb{F}_2[X]/(X^n - 1)$ of all polynomials over $\mathbb{F}_2$ of degree less than n. Given a polynomial $a \in R$ we denote with $\bar{a} \in \mathbb{F}_2^n$ the vector whose coordinates are the coefficients of a, namely, if $a = a_0 + a_1 X + \cdots + a_{n-1}X^{n-1}$ then $\bar{a} = (a_0, a_1, \ldots, a_{n-1})$. With a slight abuse of notation, we denote with both $\mathrm{w}(a)$ and $\mathrm{w}(\bar{a})$ the Hamming weight of the vector

$\bar{a}$. Consider two elements a and b in R. Their product $c \in R$ is the polynomial $a \cdot b$ and $\bar{c}$ is obtained by the formula $\bar{c}^\top = \mathrm{circ}(a) \cdot \bar{b}^\top$, where $\mathrm{circ}(a)$ is the $n \times n$ circulant matrix obtained from a. More precisely:

$$\bar{c}^\top = \underbrace{\begin{pmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{pmatrix}}_{\mathrm{circ}(a)} \cdot \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}.$$

Due to this relationship, we can link the description of our scheme to the *Syndrome Decoding Problem* and the *Codeword Finding Problem*.

In the following we work with 2-Quasi-Cyclic codes (2-QCC) [3] in systematic form over $\mathbb{F}_2$, which are a particular family of binary codes of length $2n$ and dimension n. For these codes the parity-check matrix is of the form $(\mathbb{1} \,||\, \mathrm{circ}(h))$ for some vector $h \in \mathbb{F}_2^n$. In particular, our scheme exploits the properties of 2-QCC whose parity-check matrices are sparse. Namely, 2-Quasi-Cyclic Low-Density-Parity-Check codes [3]. Notice that the two problems introduced above are efficiently solvable [3] if the underlying code is LDPC, but as we see in Section 5 this seems to be unrelated to the security of our scheme. More concretely, in this paper we propose a post-quantum digital signature scheme based on QC-LDPC codes. In the NIST Post-Quantum standardization process there were proposals for encryption schemes (e.g. BIKE [1], LEDAcrypt [6] and HQC [27]) based on the same coding problems on which we base our protocol, but to date there are still no proposals for a signature scheme which exploits these ideas. Using QC codes allows us to have smaller keys because our parity-check matrix, which is public of dimension $n \times 2n$, can be described using only n bits, while the LDPC property, on which our trapdoor relies, is also important from an implementation point of view, indeed the computation using sparse matrices speeds up the signing algorithm significantly.

In this work we introduce and discuss the main ideas behind our code-based digital signature scheme. In Section 2 we describe the setup, key generation, sign and verification phases of the scheme, dealing with its parameters and functions in Section 3. In Section 4 we give some results about the correctness of the signature, while in Section 5 we focus on the security of the scheme by discussing the hardness of some key recovery and forging attacks. At the end, in Section 6, we discuss the parameters choice according to the security levels requested by the NIST.

2 The Scheme

In the following we present our post-quantum digital signature scheme.

Setup. The parameters of the scheme are:

n	The dimension n of the vector space $\mathbb{F}_2^n$. It is a prime number such that 2 is a primitive root modulo n.
w, w_{pq}, w_r	The weights w, w_{pq}, w_r, where $\omega, \omega_{pq}, \omega_r$ are integers smaller than n, w_{pq} is odd and w_r is even.
I, I_t	The intervals I, I_t.
$\mathcal{H}_{\omega_r}$	The hash function $\mathcal{H}_{\omega_r}$ whose digests have weight ω_r over $\mathbb{F}_2^n$.

In the following sections we describe the Key-Generation algorithm KGen, the Signature algorithm Sign and the Verification algorithm Vf. We assume that the global parameters $(n, \omega, \omega_{pq}, \omega_r, I, I_t, \mathcal{H}_{\omega_r})$ are known to anyone and we do not specify them as input of the algorithms.

Key Generation. The Key Generation algorithm KGen is the following.

$\mathsf{KGen}(\emptyset)$

1. Generate randomly $x, y \in R := \mathbb{F}_2[X]/(X^n - 1)$, with $\mathrm{w}(x) = \mathrm{w}(y) = \omega$.
2. Generate randomly $p, q \in R$, with $\mathrm{w}(p) = \mathrm{w}(q) = \omega_{pq}$.
3. Define $h := pq^{-1}$.
4. Define $s := x + hy$.
5. Output the public key $\mathsf{pk} = (h, s)$ and the private key. $\mathsf{sk} = (y, q)$.

Note that actual secret key is $\mathsf{sk} = (y, q)$ but we want to keep (x, p) secret too, although they are ephemeral and their knowledge is not required in the signature phase, because if an attacker can retrieve p or x, then it can retrieve at least a part of the secret key sk.

Signature Algorithm. The signing protocol Sign is as following.

$\mathsf{Sign}(m, \mathsf{pk}, \mathsf{sk})$

1. Take as input a message m to be signed and the secret key sk. Generate the following values:
   - $r := \mathcal{H}_{\omega_r}(m \,||\, pk \,||\, \texttt{nonce})$, where `nonce` is a randomly chosen bitstring.
   - $t \in R$ such that $\mathrm{w}(t) \in I_t$.
   - $\alpha := qt + ry$ and $\beta := \alpha h + sr$. If $\mathrm{w}(\alpha)$ or $\mathrm{w}(\beta)$ do not lie in I then change the `nonce` and repeat the signing phase.
2. With this notation the signature is given by $(\alpha, \texttt{nonce})$.

Verification Algorithm. The verification algorithm Vf is as following.

$\mathsf{Vf}(m, \mathsf{pk})$

1. Compute $r := \mathcal{H}_{\omega_r}(m, pk, \texttt{nonce})$ and $\beta := \alpha h + sr$. Check that both $\mathrm{w}(\alpha)$ and $\mathrm{w}(\beta)$ lie in I.
2. If these conditions are satisfied the verifier accepts the signature, otherwise it rejects.

3 Parameters and Functions

During the setup phase we put some constrains on our parameters, namely we required n to be a prime such that 2 is a primitive root modulo n, and w_{pq} to be odd. These choices are linked to the invertibility of q in the ring R. In particular, with this choice of parameters, q will always be invertible, as a direct consequence of the following. It is well known that if K is a field of characteristic p and n is a positive integer not divisible by p, then

$$X^n - 1 = \prod_{d|n} \phi_d(X) \ ,$$

where $\phi_d(X)$ is the d-th cyclotomic polynomial. Furthermore, the following result [23] holds.

Proposition 1. *Let $\mathbb{F}_q$ be a finite field. If n is an integer such that $(n, q) = 1$, then $\phi_n(X)$ factors into $\varphi(n)/d$ distinct monic irreducible polynomials in $\mathbb{F}_q[X]$ of the same degree d, $\mathbb{F}_q^{(n)}$ is the splitting field of any such irreducible factor over $\mathbb{F}_q$, and $[\mathbb{F}_q^{(n)} : \mathbb{F}_q] = d$, where d is the least positive integer such that $q^d \equiv 1 \pmod{n}$.*

As we are working with $R = \mathbb{F}_2[X]/(X^n + 1) = \mathbb{F}_2[X]/(X^n - 1)$, since n is prime, $X^n - 1$ factors as $\phi_1(X)\phi_n(X)$ where both $\phi_1(X)$ and $\phi_n(X)$ are irreducible in $\mathbb{F}_2[X]$. To prove this, notice that the latter factors into $\phi(n)/d$ irreducible polynomials where d is the order of 2 modulo n. Since we have chosen n such that 2 is a primitive root modulo n, we have that $d = n-1 = \varphi(n)$. As a consequence, an element in R is invertible if and only if it is not divisible by $\phi_1(X)$ and by $\phi_n(X)$, but in R the only element divisible by $\phi_n(X)$ is itself, thus an element different from $\phi_n(X)$ is invertible in R if (and only if) it is not divisible by $\phi_1(X) = X + 1$. Notice that if a polynomial in R is divisible by $X + 1$ then its weight is even, so that, if we execute the protocol honestly, q will always be invertible in R.

Notice that it is possible to create hash functions which produce words of weight w_r, indeed $\mathcal{H}_{\omega_r}$ could work in the following way: let $\mathcal{H}$ be a cryptographically secure hash function with digest of 256 bit and let n, w_r be the parameters defined in the table above. The goal of this function is to output a list of w_r integer in the range $[1, n]$, these will be the positions of 1 of the digest message. First of all K compute $\bar{m} = (m \,||\, nonce)$, then let $H_1 = \mathcal{H}(\bar{m})$. In order to have a number less or equal than n we need $l = \lceil \log_2(n) \rceil$ bit, so the algorithm takes the first l bits of H_1 and, if the number associated with is in $[1, N]$, it is the first position, otherwise it discards the number. Consequently,it takes the second l bits, do the same as before, and so on. If you can no longer take l bits, the algorithm computes $H_2 = \mathcal{H}(H_1)$ and continue until it has generate w_r integer.

4 Correctness

To prove the correctness of our scheme we need the following preliminary result, whose proof can be found in [27].

Proposition 2. *Let $v = (v_0, \ldots, v_{n-1})$ be a random vector chosen uniformly among all binary vectors of weight ω_v and let $u = (u_0, \ldots, u_{n-1})$ be a random vector chosen uniformly among all vectors of weight ω_u and independently of v. Then, denoting $z = u \cdot v$, we have that for every $k \in \{0, \ldots, n-1\}$, the k-th coordinate z_k of z is Bernoulli distributed with parameter $\tilde{p} = P(z_k = 1)$ equal to:*

$$\tilde{p} = \frac{1}{\binom{n}{w_v}\binom{n}{w_u}} \sum_{\substack{1 \le l \le min(w_v, w_u) \\ l \ odd}} C_l \ ,$$

where $C_l = \binom{n}{l}\binom{n-l}{w_v-l}\binom{n-w_v}{w_u-l}$.

As a consequence, in the following corollary we derive the probability distribution of the polynomials α and β.

Corollary 3. *Using the previous notation, α and β has the same weight distribution. In particular, α and β are distributed as a Binomial with parameter $p^* = p_1(1-p_2) + p_2(1-p_1)$, where*

$$p_1 = \frac{1}{\binom{n}{w_{pq}}\binom{n}{w_t}} \sum_{\substack{1 \le l \le min(w_{pq}, w_t) \\ l \ odd}} \binom{n}{l}\binom{n-l}{w_{pq}-l}\binom{n-w_{pq}}{w_t-l},$$

$$p_2 = \frac{1}{\binom{n}{w_r}\binom{n}{w}} \sum_{\substack{1 \le l \le min(w_r, w) \\ l \ odd}} \binom{n}{l}\binom{n-l}{w_r-l}\binom{n-w_r}{w-l}.$$

Proof. Consider $\alpha = q \cdot t + r \cdot y$ and let $p_1 = \mathbb{P}((q \cdot t)_i = 1)$. Recall that $\mathrm{w}(q) = \omega_{pq}$ and $\mathrm{w}(t) = \omega_t \in I_t$. Using proposition 1 we have that:

$$p_1 = \frac{1}{\binom{n}{\omega_{pq}}\binom{n}{\omega_t}} \sum_{\substack{1 \le l \le \min(\omega_{pq}, \omega_t), \\ l \ odd}} \binom{n}{l}\binom{n-l}{\omega_{pq}-l}\binom{n-\omega_{pq}}{\omega_t-l}.$$

If we define $p_2 = \mathbb{P}((r \cdot y)_i = 1)$ with $\mathrm{w}(r) = \omega_r$ and $\mathrm{w}(y) = \omega$, using proposition 1 we have that:

$$p_2 = \frac{1}{\binom{n}{\omega_r}\binom{n}{\omega}} \sum_{\substack{1 \le l \le \min(\omega_r, \omega), \\ l \ odd}} \binom{n}{l}\binom{n-l}{\omega_r-l}\binom{n-\omega_r}{\omega-l}.$$

Let $p^* = \mathbb{P}(\alpha_i = 1)$:

$$\begin{aligned} p^* &= \mathbb{P}((q \cdot t + r \cdot y)_i = 1) \\ &= \mathbb{P}((q \cdot t)_i = 1)\mathbb{P}((r \cdot y)_i = 0) + \mathbb{P}((q \cdot t)_i = 0)\mathbb{P}((r \cdot y)_i = 1) \\ &= p_1(1-p_2) + (1-p_1)p_2. \end{aligned}$$

We can conclude that the weight distribution of α is a Binomial of parameter p^*. If we consider β we have the following:

$$\begin{aligned} \beta &= h \cdot \alpha + s \cdot r \\ &= h \cdot q \cdot t + h \cdot r \cdot y + r \cdot x + r \cdot h \cdot y \\ &= p \cdot q^{-1} \cdot q \cdot t + p \cdot q^{-1} \cdot r \cdot y + r \cdot x + r \cdot p \cdot q^{-1} \cdot y \\ &= p \cdot t + r \cdot x. \end{aligned}$$

We have just to observe that the β has the same form of α, $\mathrm{w}(p) = \mathrm{w}(q) = \omega_{pq}$ and $\mathrm{w}(x) = \mathrm{w}(y) = \omega$, then the thesis follows. □

According to Corollary 3, the public parameters n, w, w_{pq}, w_r and I_t determine the probability distribution of the weight of α and β, and thus we can find an interval I such that, if the scheme is executed honestly, the failure probability is negligible. In order to achieve a good balance between efficiency and security, a detailed discussion about the parameters' choice is discussed in Section 6.

5 Security

In this section we discuss some properties of the scheme. We start by analysing some key recovery and forgery attacks, ending up with a discussion about the role of t.

Key Recovery from s. Regarding the relationship between private and public keys, observe that $s = x + hy$ satisfies the linear equation

$$\bar{s}^\top = \left[\mathbb{1} \mid \mathrm{circ}(h)\right] \begin{pmatrix} \bar{x}^\top \\ \bar{y}^\top \end{pmatrix},$$

where x and y are two vectors of weight $\mathrm{w}(x) = \mathrm{w}(y) = \omega$. Therefore, obtaining x and y from the public key (h, s) is a particular instance of the Syndrome Decoding Problem, which is known to be NP-hard. Observe that if an attacker can retrieve (x, y) from s and it has also access to a valid signature $(\alpha = qt + ry, \texttt{nonce})$, then it can produce a forgery. Indeed, from α it can compute qt and then it can select a message m', compute r' such that $\alpha' = qt + r'y$ is a valid signature for m'. The problem of finding x, y from s is however linked to the computational *2-QCSD Problem*, where its decisional version can be stated as follows:

> *Input*: positive integers n, w_0, w_1, a random parity-check matrix H of a QC-code and a syndrome $y \in \mathbb{F}_2^n$.
>
> *Output*: there exists $x = (x_0, x_1) \in \mathbb{F}_2^n$ such that $\mathrm{w}(x_i) \leq w_i$ and $y^\top = Hx^\top$.

In our case the parity-check matrix is given by $H = (\mathbb{1} \,||\, \mathrm{circ}(h))$ and we have to take into consideration the sparsity of the matrix if we aim to fully understand the security of the scheme. Indeed, if H is a sparse matrix, then the dual of the code generated by H is a LDPC code and in that case it is well known that the SD Problem, and so the 2-QCSD Problem, can be solved in polynomial time [18]. However, under the assumption that h is indistinguishable from a randomly chosen vector (of odd weight) over $\mathbb{F}^n$, the SD Problem does not seem to be efficiently solvable. Notice that the indistinguishability of h from a random vector is assumed to be true also in well known works (e.g. BIKE [1] and LEDAcrypt [6]). The fact that with very high probability $H = (\mathbb{1}||\mathrm{circ}(h))$ is not a sparse matrix allows us to conclude that if we could solve the problem of finding (x, y) from $s = x + hy$ then we could solve an instance of the computational 2-QCSD which is believed to be difficult to solve.

Key Recovery from h. Here we consider the possibility to exploit the knowledge of h, together with the information $h = pq^{-1}$ to retrieve p and q. A way to try to retrieve p and q is the following: construct the matrix $M := \left(I_n \,||\, \mathrm{circ}(h)\right)$ and notice that $q \cdot M = (q, p)$, so that (q, p) is a codeword of low weight of the code with generator matrix M. As a consequence, (q, p) is a solution of an instance of the *Codeword Finding Problem*, which is NP-hard.

Key Recovery from a Valid Signature. Here we consider the case in which an attacker has access to a valid signature $(\alpha, \texttt{nonce})$ of a given message m. The attacker can compute r and try find the vector (qt, y) simply by solving $(\mathbb{1} \,||\, \mathrm{circ}(r))(qt, y)^\top = \alpha^\top$. Observe that the matrix $H = (\mathbb{1} \,||\, \mathrm{circ}(r))$ is a sparse matrix and so the dual of the code generated by H is a LDPC code. This tells us that the syndrome decoding related to this code is efficiently solvable. In particular, we know an efficient algorithm that takes as input a parity-check matrix H, a syndrome s, a weight w and it outputs a solution x such that $Hx^\top = s^\top$ and $\mathrm{w}(x) \leq w$. It is easy to observe that if someone succeeds in carrying this attack with non-negligible probability, then, in addition to be able to steal a part of the private key, it would be able to perform a forgery with non-negligible probability. In the following we prove that this attack can succeeds only with negligible probability. Notice that we know the weights of each polynomial involved in the expression (qt, y), and so we can compute a weight $w_{\max}$ such that with very high probability the weight of (qt, y) is less than $w_{\max}$. In other words, we have a weight which is reasonable for the SD

Problem with matrix $H = (\mathbb{1} \mid\mid \mathrm{circ}(r))$ and syndrome $\alpha^\top$. We show that, given H, α and w, the number of vector x which are solution of the associated SD Problem are just too many to hope to find (qt, y) among these. Let $\alpha \in \mathbb{F}_2^n \setminus \{0\}$ be a syndrome and define $C_\alpha := \{v \in \mathbb{F}_2^n \mid Hv^\top = \alpha^\top\}$. Assume that the weight distribution of the elements in C_α can be approximated with a binomial distribution with length $2n$ and probability $p = 1/2$ (this can be statistically shown through a goodness of fit test exploiting the fact that w_r is even). If we randomly extract a vector $v \in \mathbb{F}_2^{2n}$, then

$$\mathbb{P}(\mathrm{w}(v) = i) = \frac{\binom{2n}{i}}{2^{2n}}.$$

It is easy to show that $|C_\alpha| = 2^n$, therefore:

$$\big|\{v \mid v \in C_\alpha \text{ and } \mathrm{w}(v) \leq w_{\max}\}\big| = 2^n \cdot \sum_{i=0}^{w_{\max}} \mathbb{P}\big(\mathrm{w}(v) = i\big) = \frac{1}{2^n} \cdot \sum_{i=0}^{w_{\max}} \binom{2n}{i}.$$

In conclusion, with the parameters used to instantiate the scheme, the probability of finding the correct vector (qt, y) among these is negligible.

Forging a Signature of a Chosen Message. To produce a forgery of a given message, an adversary is required to determine two values α and β such that their weight lie in the interval I. According to the previous section, a valid user is capable of producing with non-negligible probability a signature due to the knowledge of the private key. On the other hand, an adversary capable of producing a forgery is also capable of solving the problem

$$\begin{cases} \mathrm{w}(\alpha h + sr) \in I \\ \mathrm{w}(\alpha) \in I \end{cases},$$

which can also be described in the following way:

$$\begin{cases} (\overline{sr})^\top = \Big[\mathbb{1} \mid \mathrm{circ}(h)\Big] \begin{pmatrix} \bar{\beta}^\top \\ \bar{\alpha}^\top \end{pmatrix} \\ \mathrm{w}(\alpha) \in I \\ \mathrm{w}(\beta) \in I \end{cases}.$$

This problem is however strictly related to the SDP, which is NP-complete.

The usage of t. Notice that it is mandatory to use the ephemeral value t just once. As usual for such signatures, it would be a dab idea to send different messages using the same ephemeral key. In our case we would end up with two signatures $(\alpha_1, \mathtt{nonce}_1), (\alpha_2, \mathtt{nonce}_2)$ such that

$$\begin{cases} \alpha_1 = qt_1 + r_1 y \\ \beta_1 = h\alpha_1 + sr_1 \end{cases} \quad \text{and} \quad \begin{cases} \alpha_2 = qt_1 + r_2 y \\ \beta_2 = h\alpha_2 + sr_2 \end{cases},$$

from which it immediately follows that $\alpha_1 + \alpha_2 = (r_1 + r_2)y$. If $(r_1 + r_2)$ was invertible than we could compute $(\alpha_1 + \alpha_2)(r_1 + r_2)^{-1}$ and find y. However, this attack is not possible because $r_1 + r_2$ is always non-invertible in our framework. Indeed, a simple argument on the weights of r_1, r_2 and their sum shows that $r_1 + r_2$ is invertible in R if an only if only one of the two addends is invertible. According to this, an attacker can still try to find y by solving the linear system of equation generated by $\mathrm{circ}(r_1 + r_2)y = \alpha_1 + \alpha_2$. Hence, we have to care about the number of possible solutions of that system. According

to this, we limit ourselves to consider the linear system described by $\mathrm{circ}(r)y+\alpha$. where α and r are known, r has even weight, and we ask for how many y this system admits a solution. Notice that the element that are solution of the previous system are exactly the elements that belong to the fiber of $\mathcal{L}_{r,\alpha}$ in zero, where

$$\begin{aligned} \mathcal{L}_{r,\alpha} : \mathbb{F}_2^n &\longrightarrow \mathbb{F}_2^n \\ y &\longmapsto \mathrm{circ}(r)y + \alpha \end{aligned}.$$

Since we are interested in computing $(\mathcal{L}_{r,\alpha})^{-1}(0)$ and the function $x \mapsto x + \alpha$ is a translation map, we can study the fiber of $\mathcal{L}_{r,0}$ in zero. It is easy to show that $\mathcal{L}_{r,0}$ is an homomorphism of vector spaces, so that the First Isomorphism Theorem guarantees that there is a bijection between the image space $\mathcal{L}_{r,0}(\mathbb{F}_2^n)$ and the quotient space $\mathbb{F}_2^n / \ker(\mathcal{L}_{r,0})$, from which it follows that

$$|\ker(\mathcal{L}_{r,0})| = \frac{|\mathbb{F}_2^n|}{|\mathcal{L}_{r,0}(\mathbb{F}_2^n)|}.$$

This computation is achievable, indeed the rank of a circulant matrix $\mathrm{circ}(r)$ is equal to $n - d$, where d is the degree of the polynomial (r, X^{n-1}). In our case $X^{n-1} = (X+1)(1 + x + \ldots + X^{n-1})$, so the greatest common divisor between X^{n-1} and r is $X + 1$. It follows immediately that $d = 1$ and the rank of $\mathrm{circ}(h)$ is $n - 1$. As a consequence, $|\ker(\mathcal{L}_{r,0})| = 2^n/2^{n-1} = 2$, that is, it contains only y and the zero element. In conclusion, since there are only two possible solutions of that system, we must take care of the usage of t, changing its value every time we sign a document.

6 Experimental Results

In the following, we describe the choice of the parameters for security levels $1, 3$ and 5, as required by the NIST. The choice is justified by some related experimental results. According to the description of the setup phase, described in Section 2, we choose w_{pq} and w in order to have approximately half the number of bits of n. The weight of r is taken in such a way that $w_r \approx 3 \log n$, and I_t is the interval of length $\sqrt{n}$ and centered in $\sqrt{n}$. This choice allows us to compute I in such a way that it will be possible for a genuine signer to sign efficiently and it will be unfeasible for an attacker to break the protocol. The choice for the size of these parameters is summarized in Table 1.

Parameter	Size
ω_{pq}	$\approx \sqrt{n}$
I_t	$[\lceil \frac{\sqrt{n}}{2} \rceil, \ldots, \lceil \frac{2\sqrt{n}}{3} \rceil]$
ω_r	$\approx 3\log(n)$
ω	$\approx \sqrt{n}$

Table 1: Size of the parameters.

In Table 2 we define the different values for n for the standard security levels and then we describe the best attacks to our scheme.

As shown in the previous section the security of $h = pq^{-1}$ is strictly related to hardness of the *Codeword Finding Problem*. This problem in NP-hard in the general case and in the case of Quasi-Cyclic codes the attack is just slightly improved (DOOM attack [35]). In order to estimate the complexity of an attack, we used a Python script which can be found in [7]. We used the same test to analyze the security level of

Security level	n
128	14627
192	18143
256	21067

Table 2: Values for the security levels.

$s = x + hy$, which rely on the difficulty of the *Syndrome Decoding Problem*. Clearly, the CFP is a particular instance of the SDP and it is reasonable to think that there are algorithms which work faster in the case of the CFP, however this is not true, as the best algorithms that solves SDP are also the best algorithms for solving CFP. This is the reason for which we use the same set of algorithms. The explanation of these attacks and the relative adaptation to CFP can be found in [4]. The results are summarized in Table 3.

n	Security bits (attack to h)	Security bits (attack to s)
14627	250	250
18143	277	277
21067	297	297

Table 3: Security bits for attacks to h and s.

The size of n may seem overestimated with respect to the security parameters, but this is not the only point on which the security lies. We also have to take into account the distance that the weight distribution of α has with respect to a binomial distribution with parameters $(n, \frac{1}{2})$. We know that the weight distribution of α is a binomial random variable with parameter (n, p^*), where p^* is computed as described in Corollary 3. Since the weight of t is not fixed, but can vary within the range I_t, also p^* can vary accordingly. In Table 4 we summarize the range of p^* and the interval I for the different security levels. The interval I is computed in such a way that, if the signer is honest, the probability that the weight of α and β result outside the range I is $\approx 10^{-10}$.

Security level	Range of p^*	I
128	$[0.38918, 0.42088]$	$[5339, 6515]$
192	$[0.38549, 0.41790]$	$[6601, 7981]$
256	$[0.38174, 0.41564]$	$[7620, 9186]$

Table 4: Ranges for α and β.

So, an attacker may try to guess an α in the range I and check if $\beta = \alpha h + rs$ also belongs to the range I. Notice that if an attacker can find such α, then it can create a forgery. In this case, since we do not have any constrains about the weight of h, β appears as a totally random vector, i.e. its weight distribution is a binomial random variable with parameters $(n, \frac{1}{2})$. In Table 5 we summarize the security bits for this type of attack.

Security level	Security bits for random attack to α
128	133
192	196
256	256

Table 5: Attack by guessing α.

Notice that an attacker could try to solve the SDP related to α, i.e. it can try to find (qt, y) by knowing that $(\mathbb{1} \mid\mid \mathrm{circ}(r))(\bar{qt}, \bar{y})^\top = \bar{\alpha}^\top$. Observe that this instance of the SDP is easily solvable since the associated code is an LDPC code [18]. In Table 6 we can observe that the total number of solutions is too large to hope that the vector v given from the decoder is exactly the sensible information (qt, y).

Security level	p_{qt}	$\omega_{\max}$	Number of solutions
128	0.37153	5652	$\approx 10^{1905}$
192	0.37118	6977	$\approx 10^{2333}$
256	0.37169	8092	$\approx 10^{2670}$

Table 6: Attack to α. $\mathrm{w}(qt)$ is distributed as a binomial r.v. with parameters (n, p_{qt}), and $\omega_{\max}$ is such that the probability that the weight of (qt, y) is greater than $\omega_{\max}$ is negligible.

At the end, we can study the hardness of forging a signature by solving the SDP instance defined in Section 5. In order to do this we use the best algorithm to attack the SDP and the complexity of these attacks can be computed using [7]. Since, in this case, the parameters are very big, we could not compute the complexity of all the attacks. Table 7 provides an idea about the complexity of computing a forgery in this way.

Security level	Security bits for forgery exploiting SDP
128	≈ 7000
192	≈ 8000
256	≈ 9000

Table 7: Forgery complexity by attacking SDP.

7 Conclusions and Future Works

We have described an efficient general approach for constructing a post-quantum code-based digital signature scheme. The main advantages arise from the structure of Quasi-Cyclic codes, which allow to achieve small key sizes, together with the structure of LDPC codes, which provide good performance overall. We have provided hints on the security of the scheme, nevertheless this topic requires a more deep analysis and possibly a formal proof reducing the security of the scheme to the complexity of the related complexity problems. Notice that, according to Corollary 3, the distribution of the Hamming weight of α and β can be modeled as a binomial with parameter $p^* \neq \frac{n}{2}$, and thus distinguishable from the distribution of random vectors. This feature, even though fundamental to the success of the signing process, may be linked to a possible information leakage. This security issue has to be carefully analyzed to understand the behavior of the proposed signature algorithm with respect to an attacker knowing multiple valid signatures. The result of this study will be fundamental to determine the keys' lifecycle.

8 Acknowledgements

The publication was created with the co-financing of the European Union - FSE-REACT-EU, PON Research and Innovation 2014-2020 DM1062 / 2021. The first and the last authors are members of the INdAM Research Group GNSAGA. The core of this

work is contained in the second author's M.Sc. thesis, and was partially presented at PQCifris2022 School&Workshop on Post-Quantum Cryptography, October 10-14 2022, Trento, by the third author. The share of the author's contributions in this paper is equal.
The authors would like to thank Marco Baldi, Jean-Christophe Deneuville, Edoardo Persichetti and Paolo Santini for their helpful comments, as well as the other participants of PQCifris2022 for the interesting discussion.

References

[1] N. Aragon, P. S. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Guneysu, C. A. Melchor, et al. Bike: bit flipping key encapsulation. 2017.

[2] D. Augot, M. Finiasz, and N. Sendrier. A family of fast syndrome based cryptographic hash functions. In *International Conference on Cryptology in Malaysia*, pages 64–83. Springer, 2005.

[3] M. Baldi. *QC-LDPC code-based cryptography*. Springer Science & Business, 2014.

[4] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini. A finite regime analysis of information set decoding algorithms. *Algorithms*, 12(10):209, 2019.

[5] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini. Ledacrypt: low-density parity-check code-based cryptographic systems. *NIST round*, 2, 2019.

[6] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini. Ledacrypt: Qc-ldpc code-based cryptosystems with bounded decryption failure rate. In *Code-Based Cryptography Workshop*, pages 11–43. Springer, 2019.

[7] E. Bellini and A. Esser. Syndrome decoding estimator, 2021.

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

[9] D. J. Bernstein, T. Chou, T. Lange, I. von Maurich, R. Misoczki, R. Niederhagen, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, et al. Classic McEliece: conservative code-based cryptography. *Submission to the NIST's post-quantum cryptography standardization process*, 2017.

[10] J. Bruck and M. Naor. The hardness of decoding linear codes with preprocessing. *IEEE Transactions on Information Theory*, 36(2):381–385, 1990.

[11] Q. Cheng. Hard problems of algebraic geometry codes. *IEEE Transactions on Information Theory*, 54(1):402–406, 2008.

[12] N. T. Courtois, M. Finiasz, and N. Sendrier. How to achieve a mceliece-based digital signature scheme. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 157–174. Springer, 2001.

[13] G. D'Alconzo, A. Meneghetti, and P. Piasenti. Security issues of cfs-like digital signature algorithms. *arXiv preprint arXiv:2112.00429*, 2021.

[14] L. Dallot. Towards a concrete security proof of courtois, finiasz and sendrier signature scheme. In *Western European Workshop on Research in Cryptology*, pages 65–77. Springer, 2007.

[15] I. B. Damgård. A design principle for hash functions. In *Conference on the Theory and Application of Cryptology*, pages 416–427. Springer, 1989.

[16] K. Fukushima, P. S. Roy, R. Xu, et al. Random code-based signature scheme (racoss). In *First Round Submission to the NIST Post-quantum Cryptography Call*. 2017.

[17] P. Gaborit, O. Ruatta, J. Schrek, and G. Zémor. Ranksign: an efficient signature algorithm based on the rank metric. In *International Workshop on Post-Quantum Cryptography*, pages 88–107. Springer, 2014.

[18] R. Gallager. Low-density parity-check codes. *IRE Transactions on information theory*, 8(1):21–28, 1962.

[19] V. Gandikota, B. Ghazi, and E. Grigorescu. On the np-hardness of bounded distance decoding of reed-solomon codes. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 2904–2908. IEEE, 2015.

[20] V. Guruswami and A. Vardy. Maximum-likelihood decoding of reed-solomon codes is np-hard. *IEEE Transactions on Information Theory*, 51(7):2249–2256, 2005.

[21] G. Kabatianskii, E. Krouk, and B. Smeets. A digital signature scheme based on random error-correcting codes. In *IMA International Conference on Cryptography and Coding*, pages 161–167. Springer, 1997.

[22] Y. Lee, W. Lee, Y. S. Kim, and J.-S. No. Modified pqsigrm: Rm code-based signature scheme. *IEEE Access*, 8:177506–177518, 2020.

[23] R. Lidl and H. Niederreiter. *Finite fields*. Number 20. Cambridge university press, 1997.

[24] A. Lobstein. The hardness of solving subset sum with preprocessing. *IEEE Transactions on Information Theory*, 36(4):943–946, 1990.

[25] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Coding Thv*, 4244:114–116, 1978.

[26] C. A. Melchor, N. Aragon, M. Bardet, S. Bettaieb, L. Bidoux, O. Blazy, and J.-C. Deneuville. Rollo-rank-ouroboros, lake & locker. 2019.

[27] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, G. Zémor, and I. Bourges. Hamming quasi-cyclic (hqc). *NIST PQC Round*, 2:4–13, 2018.

[28] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, and G. Zémor. Rank quasi-cyclic (rqc). 2017.

[29] A. Meneghetti, A. Pellegrini, and M. Sala. On the equivalence of two post-quantum cryptographic families. *Annali di Matematica Pura ed Applicata (1923-)*, pages 1–25, 2022.

[30] R. C. Merkle. One way hash functions and des. In *Conference on the Theory and Application of Cryptology*, pages 428–446. Springer, 1989.

[31] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Contr. Inform. Theory*, 15(2):157–166, 1986.

[32] A. Otmani and J.-P. Tillich. An efficient attack on all concrete kks proposals. In *International Workshop on Post-Quantum Cryptography*, pages 98–116. Springer, 2011.

[33] W. Peterson. Encoding and error-correction procedures for the bose-chaudhuri codes. *IRE Transactions on information theory*, 6(4):459–470, 1960.

[34] F. Ren, D. Zheng, W. Wang, et al. An efficient code based digital signature algorithm. *Int. J. Netw. Secur.*, 19(6):1072–1079, 2017.

[35] N. Sendrier. Decoding one out of many. In *International Workshop on Post-Quantum Cryptography*, pages 51–67. Springer, 2011.

[36] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, 1997.