Code-based signatures without trapdoors through restricted vectors

Marco Baldi, Franco Chiaraluce, and Paolo Santini

Università Politecnica delle Marche, Ancona, Italy
{m.baldi, f.chiaraluce, p.santini}@univpm.it

Abstract. The Schnorr-Lyubashevsky approach has been shown able to produce secure and efficient signature schemes without trapdoors in the lattice-based setting, exploiting small vectors in the Euclidean metric and rejection sampling in the signature generation. Translating such an approach to the code-based setting has revealed to be challenging, especially for codes in the Hamming metric. In this paper, we propose a novel adaptation of the Schnorr-Lyubashevsky framework to the code-based setting, by relying on random non-binary linear codes and vectors with restricted entries to produce signatures. We provide some preliminary arguments to assess the security of the new scheme and to compute its parameters. We show that it achieves compact and competitive key and signature sizes, even without resorting to structured random codes.

Keywords: Code-based cryptography · digital signatures · post-quantum cryptography

1 Introduction

There are basically two approaches to code-based digital signatures. The first one, derived from the "hash-and-sign" paradigm used for instance in RSA signatures, encounters some obstacles when applied to the code-based setting. This is due to the difficulty of randomly generating a decodable syndrome, yielding code-based schemes that are inefficient or insecure (or both). Two historical proposals along this line are CFS [13] and KKS [21], which however have important limitations. In fact, it is very difficult to find secure though efficient instances of the KKS scheme [24]. The CFS scheme is more consolidated, but requires Goppa codes with extreme parameters (e.g., with very high rate) in order to be efficient, and this exposes the scheme to Goppa code distinguishers [18]. Some variants of CFS aimed at using non-algebraic codes for reducing the public key size have been proposed [7], but changing the underlying family of codes yielded to successful cryptanalysis [27].

Another important drawback of existing hash-and-sign code-based signature schemes is the large public key. A recent and relevant scheme in this line, Wave [14], is based on the hardness of decoding vectors with very large Hamming weight and has a public key size growing quadratically with the security level, which is an important improvement over CFS. Nevertheless, it requires public keys of more than 3 megabytes for 128-bit security, which are still significantly larger than those required by competitor schemes relying on different trapdoors, like lattice-based ones.

A different approach to code-based signatures, which has the advantage of not relying on any trapdoor for key derivation, is that of applying the Fiat-Shamir transform [19] to an identification scheme. In fact, consolidated zero-knowledge code-based identification schemes exist since a long time [31], which however exhibit significant soundness errors and thus require many repetitions to achieve reasonable security levels. This results in large signature sizes when they are used for digital signatures. Subsequent variants of these schemes aim at overcoming such limitations [33, 12, 1, 17, 9, 10, 5], but their characteristics are still far from being comparable with those of signature schemes relying on other mathematical objects, such as lattices.

One of the main advantages of lattice-based schemes is that they can exploit the approach introduced by Lyubashevsky in [23], achieving very compact keys and short signatures, besides high algorithmic efficiency. Such an approach is at the basis of Dilithium [16], one of the most promising digital signature schemes participating to the ongoing NIST competition. This has motivated many attempts to translate the Schnorr-Lyubashevsky approach into the domain of code-based schemes, as done in [25, 30, 22]. In most cases, however, these attempts have been shown to be unsecure [29, 15, 2, 6]. While there are adaptations of the Schnorr-Lyubashevsky approach in the rank metric code-based setting that are considered secure [3] and achieve competitive performance, no valid solution has been found in the Hamming metric code-based setting to date. In many of the aforementioned examples using codes in the Hamming metric, binary codes and sparse signatures were used, obtained from a sparse secret key via linear algebra: this feature is at the core of the corresponding attacks and definitely represents a weak choice. The crucial difference between codes and lattices lies in how a small vector can be defined. In the Hamming metric, smallness has often been associated with sparsity, thus requiring the presence of a large number of zero entries in noisy vectors: this unfortunately makes the noise ineffective in masking the secret key. Lattices instead are defined in the Euclidean metric, for which a small vector does not necessarily contain zero entries: this is the key to dispose of small though secure noisy vectors.

Our contribution In this paper we propose a novel code-based adaptation of the Lyubashevsky signature scheme. In particular, we rely on the hardness of the recently introduced Restricted Syndrome Decoding Problem (R-SDP), proven NP-complete in [5], which asks to decode vectors whose entries lie in a subset of the underlying finite field. We employ a bunch of such vectors, which we call *restricted*, as the secret key and use their syndromes through a public parity-check matrix as the public key. We rely on dense, restricted noise vectors to hide the secret key into signatures and, as in Schnorr-Lyubashevsky scheme, we exploit rejection sampling to tune the signatures distribution and prevent information leakage. We assess the security of our scheme by studying the hardness of solving the associated R-SDP problems, and propose some preliminary sets of parameters. Our results show that the proposed scheme achieves competitive public key and signature sizes, with acceptable rejection rates in the signature generation.

2 Notation and background

For two integers a and b, we denote as $[a;b]$ the set of integers x such that $a \le x \le b$; we use $[\![a;b]\!]$ to indicate the range defined by all reals x such that $a \le x \le b$. As usual, $\mathbb{F}_q$ denotes the finite field with q elements; throughout the paper, we always assume that q is prime. We denote matrices and vectors with bold capital and small letters, respectively. Given a matrix $\mathbf{A}$, we denote its entry in the i-th row and j-th column as $a_{i,j}$; for a vector $\mathbf{a}$, we refer to its i-th entry as a_i. The identity matrix of size r will be indicated as $\mathbf{I}_r$. By support of a vector $\mathbf{a}$, we mean the set containing the indexes of non-zero coordinates. For two vectors $\mathbf{a}$ and $\mathbf{b}$ with length n, we indicate their inner product as $\langle \mathbf{a} \,;\, \mathbf{b} \rangle = \sum_{i=0}^{n-1} a_i b_i$. If $\mathcal{D}$ is a probability distribution, with some abuse of notation, we write $\mathcal{D}(\mathbf{a})$ to indicate the probability that $\mathcal{D}$ outputs $\mathbf{a}$. For a set A, we write $a \xleftarrow{\$} A$ if a is uniformly picked at random among all the elements of A.

2.1 Coding theory preliminaries

Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a linear code over $\mathbb{F}_q$ with length n, dimension k, redundancy $r = n - k$ and rate $R = k/n$. We represent codes through their parity-check matrix, i.e., a full rank matrix $\mathbf{H} \in \mathbb{F}_q^{r \times n}$ such that $\left\{\mathbf{H}\mathbf{c}^\top = \mathbf{0} \mid \forall \mathbf{c} \in \mathscr{C}\right\}$. A *systematic* parity-check matrix is a parity-check matrix in the form $\mathbf{H} = [\mathbf{I}_r, \mathbf{P}]$, where $\mathbf{P} \in \mathbb{F}_q^{r \times k}$. For $\gamma \le \lfloor q/2 \rfloor$, we denote with $S_{\gamma,t}$ the set of length-n vectors with entries over $\{0, \pm 1, \ldots, \pm\gamma\} \subseteq \mathbb{F}_q$ and support size t. For the set of vectors with support size not greater than t, we instead write $B_{\gamma,t} = \bigcup_{i=0}^{t} S_{\gamma,i}$. The Restricted-Syndrome Decoding Problem (R-SDP) is defined as follows [5].

Problem 1. R-SDP$_{\gamma,\le t}$: R-SDP with bounded support size t
Let $\mathbf{H} \in \mathbb{F}_q^{r \times n}$, $\mathbf{s} \in \mathbb{F}_q^{r}$ and $t \in \mathbb{N}$. Find $\mathbf{e} \in B_{\gamma,t}$ such that $\mathbf{H}\mathbf{e}^\top = \mathbf{s}$.

As proven in [5] with a reduction from the decoding problem in the Hamming metric, the decisional version of the above problem is NP-complete, regardless of the value of γ. We will also consider a slightly modified version of Problem 1, where we require the support of the searched vector to be exactly t (and hence, require the solution vector to be in $S_{\gamma,t}$). We denote the associated problem as R-SDP$_{\gamma,=t}$. It is easily seen that the decisional version of R-SDP$_{\gamma,=t}$ is NP-complete, as well: a polynomial time solver can be used to solve any R-SDP$_{\gamma,\le t}$ instance (we invoke such a solver for no more than t times). Notice that the difference between these two problems is mostly formal, since they can be solved with the same techniques. More details are given in Section 5.

3 New signature scheme

The scheme we propose is parameterized by the positive integers r, n, q, b, w_E, w_c, t_E, γ, $\bar{\gamma}$, with $r, b, t_E < n$, $w_E, w_c \leq b$ and $\bar{\gamma} < \gamma < q/2$. It additionally employs two probability distributions $\mathcal{D}_y, \mathcal{D}_z$ defined over $\mathbb{F}_q^n$. Finally, we make use of an hash function Hash that outputs vectors of size b with entries over $\{0, \pm 1\} \subseteq \mathbb{F}_q$ and support of size w_c. The scheme we propose consists of the following triplet of algorithms.

Key generation

1. Select at random $\mathbf{P} \in \mathbb{F}_q^{r \times k}$ and set $\mathbf{H} = [\mathbf{I}_r | \mathbf{P}]$.
2. Select at random $\mathbf{E} \in \{0, \pm 1\}^{b \times n}$ such that each column has support size w_E, and each row has support size not lower than t_E;
3. Compute $\mathbf{S} = \mathbf{E}\mathbf{H}^\top \in \mathbb{F}_q^{b \times r}$;
4. Set $\mathsf{sk} = \mathbf{E}$, $\mathsf{pk} = \{\mathbf{H}, \mathbf{S}\}$.

Signature generation On input a message m:

1. Sample $\mathbf{y} \in \mathbb{F}_q^n$ from $\mathcal{D}_y$;
2. Compute $\mathbf{s}_y = \mathbf{y}\mathbf{H}^\top$;
3. Compute $\mathbf{c} = \mathsf{Hash}(m, \mathbf{s}_y)$;
4. Compute $\mathbf{z} = \mathbf{c}\mathbf{E} + \mathbf{y}$;
5. Perform rejection sampling to tune the distribution of $\mathbf{z}$ to $\mathcal{D}_z$;
6. Output $\sigma = \{\mathbf{z}, \mathbf{c}\}$.

Signature verification On input m and $\sigma = \{\mathbf{z}, \mathbf{c}\}$

1. Verify that $\mathcal{D}_z(\mathbf{z}) \neq 0$, reject otherwise;
2. compute $\mathbf{s}_y = \mathbf{z}\mathbf{H}^\top - \mathbf{c}\mathbf{S}$;
3. accept if $\mathbf{c} = \mathsf{Hash}(m, \mathbf{s}_y)$, reject otherwise.

We choose $\mathcal{D}_y$ as the uniform distribution over $\{0, \pm 1, \ldots, \pm\gamma\} \subseteq \mathbb{F}_q$, and set $\mathcal{D}_z$ as the uniform distribution over $\{0, \pm 1, \ldots, \pm\bar{\gamma}\} \subseteq \mathbb{F}_q$. Through the rejection sampling, we tune the distribution of each entry in $\mathbf{z}$ to be distributed according to $\mathcal{D}_z$, and thus make $\mathbf{z}$ indistinguishable from a uniform element of $S_{\bar{\gamma}, n}$. To achieve a low rejection rate, we set $\bar{\gamma}$ to be slightly lower than γ. We aim at rejecting each entry with a very low probability ϵ, so that the average number of signatures that one computes before the signing algorithm outputs something is given by $(1 - \epsilon)^{-n}$. More details about this phase are given in Section 4.

In the verification process, to test whether $\mathcal{D}_z(\mathbf{z})$ is null or not, it is enough to verify that each entry of $\mathbf{z}$ is not outside $\{0, \pm 1, \ldots, \pm\bar{\gamma}\}$. Finally, it is easily seen that an honest signature always gets accepted, since

$$\mathbf{z}\mathbf{H}^\top - \mathbf{c}\mathbf{S} = (\mathbf{c}\mathbf{E} + \mathbf{y})\mathbf{H}^\top - \mathbf{c}\mathbf{E}\mathbf{H}^\top = \mathbf{y}\mathbf{H}^\top = \mathbf{s}_y.$$

The public key size corresponds to the number of bits one needs to represent $\mathbf{H}$ and $\mathbf{S}$. Notice that, since $\mathbf{H}$ is random and systematic, it can be fully represented through the seed used to generate $\mathbf{P}$. To represent $\mathbf{S}$, exploiting an efficient representation for arrays in $\mathbb{F}_q$, we can use $\lceil br\log_2(q)\rceil$ bits. The signature size is given by the size of $\mathbf{z}$ plus that of $\mathbf{c}$. Since each entry of $\mathbf{z}$ takes values in $\{0, \pm 1, \ldots, \pm\bar{\gamma}\}$, for its representation we can use $\lceil n\log_2(1+2\bar{\gamma})\rceil$ bits. For $\mathbf{c}$, it is enough to send its support, together with a single bit for each non-null entry, representing whether it is 1 or -1: this requires $w_c + n\lceil\log_2(n)\rceil$ bits.

4 Statistics and rejection sampling

In this section we study the statistical distribution of the entries of each produced signature. Due to lack of space, the proofs of this section are reported in Appendix A. We start by deriving the probability distribution of the entries in the product $\mathbf{cE}$; to do this, we consider the inner product between a random $\mathbf{c} \in \{0, \pm 1\}^b$ and a vector $\mathbf{e} \in \{0, \pm 1\}^b$ modeling a column of the secret $\mathbf{E}$.

Lemma 1. *Let $w_E, w_c \in \mathbb{N}$ such that w_E is even and $\min\{w_c\ ,\ w_E\} < q$. Let $\mathbf{e} \in \{0, \pm 1\}^b$ with support size w_E. Let $\mathbf{c}$ be random over $\{0, \pm 1\}^b$, with support size w_c. Then, the probability that $\langle \mathbf{c}\ ;\ \mathbf{e}\rangle$ is equal to $\beta \in \mathbb{F}_q$ is given by*

$$g_{q,w_E,w_c,b}(\beta) = \sum_{\substack{v=\beta_q \\ v \text{ and } \beta_q \text{ have the same parity}}}^{\min\{w_E\ ,\ w_c\}} 2^{-v}\frac{\binom{v}{\frac{v+\beta_q}{2}}\binom{w_E}{v}\binom{b-w_E}{w_c-v}}{\binom{b}{w_c}},$$

where $\beta_q = \min\{\beta\ ,\ q-\beta\}$.

Lemma 2. *Let $w_E, w_c \in \mathbb{N}$ such that w_E is even and $\min\{w_c\ ,\ w_E\} < q$. Let $\mathbf{e} \in \{0, \pm 1\}^b$ with support size w_E. Let $\mathbf{c}$ be random over $\{0, \pm 1\}^b$, with support size w_c, and $y \xleftarrow{\$} \{0, \pm 1, \ldots, \pm\gamma\}$. Then, $\Pr\left[\langle \mathbf{c}\ ;\ \mathbf{e}\rangle + y = \beta\right]$ is equal to*

$$\tilde{g}_{q,w_E,w_c,b,\gamma}(\beta) = \frac{\sum_{x=-\gamma}^{\gamma} g_{q,w_E,w_c,b}(\beta - x)}{2\gamma+1}.$$

In the signing algorithm, we employ a rejection sampling criterion to tune the distribution of produced signatures to a desired target. In particular, we want each entry of $\mathbf{z}$ to follow the uniform distribution over $\{0, \pm 1, \ldots, \pm\gamma\} \subseteq \mathbb{F}_q$. To estimate the rejection rate, we rely on the following proposition, which in turn is based on the well-known rejection sampling lemma.

Proposition 1 (Rejection sampling). *Let $w_E, w_c, \gamma, \bar{\gamma} \in \mathbb{N}$ such that w_E is even, $\min\{w_c\ ,\ w_E\} < q$ and $\bar{\gamma} < \gamma < q$. Let $\mathbf{e} \in \{0, \pm 1\}^b$ with support size w_E. Let $\mathcal{F}$ be the uniform distribution over $\{0, \pm 1, \ldots, \pm\bar{\gamma}\}$, with probability distribution $f : \mathbb{F}_q \mapsto [\![0;1]\!]$. Let $M = \max_{\beta\in\{0,\pm 1,\ldots,\pm\bar{\gamma}\}}\left\{\frac{f(\beta)}{\tilde{g}_{w_E,w_c,q,b,\gamma}(\beta)}\right\}$, and $\mathcal{G}$ be the distribution resulting from the following experiment:*

1. *sample $\mathbf{c}$ at random over $\{0, \pm 1\}^b$, with support size w_c;*
2. *sample $y \xleftarrow{\$} \{0, \pm 1, \ldots, \pm\gamma\}$;*

3. compute $z = \langle \mathbf{c} \,;\, \mathbf{e} \rangle + y$;
4. output z with probability $\frac{f(z)}{M\tilde{g}_{w_E,w_c,q,b,\gamma}(z)}$.

Then, $\mathcal{G}$ outputs something with probability $1/M$, and the samples obtained from $\mathcal{G}$ are distributed according to $\mathcal{F}$.

As a trivial application, we extend the above proposition to the case of multiple samples obtained from $\mathcal{G}$. In other words, we consider a vector $\mathbf{z}$ obtained by repeating the experiment of $\mathcal{G}$ for n times, i.e.:

1. we choose $\mathbf{E} \in \mathbb{F}_q^{b\times n}$ such that each column has support size w_E;
2. we pick $\mathbf{c}$ at random over $\{0, \pm 1\}^b$, with support size w_c;
3. we pick $\mathbf{y} \xleftarrow{\$} \{0, \pm 1, \dots, \pm\gamma\}^n$ and compute $\mathbf{z} = \mathbf{cE} + \mathbf{y}$;
4. we output $\mathbf{z}$ with probability $\frac{1}{M^n}\prod_{i=0}^{n-1} \frac{f(z_i)}{\tilde{g}_{w_E,w_c,q,b,\gamma}(z_i)}$.

It is easily seen that the resulting distribution outputs something with probability M^{-n}, and that each entry of the sample is distributed according to $\mathcal{F}$.

5 Solving the R-SDP

We consider the setting in which $\mathbf{H}$ is the parity-check matrix of a random code $\mathscr{C}$ with length n and dimension k, and assume that at least one solution always exists. For the fixed support size problem (i.e., R-SDP$_{\gamma,=t}$), we assume that the target syndrome is picked as $\mathbf{s} \xleftarrow{\$} \left\{\mathbf{eH}^\top,\ \mathbf{e} \in S_{\gamma,t}\right\}$; in such a case, the number of solutions can be estimated as

$$N_{\gamma,=t} = 1 + \frac{|S_{\gamma,t}| - 1}{q^{n-k}} \approx 1 + \binom{n}{t} 2^{t\left(1+\log_2(\gamma)\right) - (n-k)\log_2(q)}. \tag{1}$$

Indeed for each $\mathbf{e} \in S_{\gamma,t}$ we have that $\mathbf{eH}^\top$ is random over $\mathbb{F}_q$ (since $\mathbf{H}$ is random), hence it is equal to $\mathbf{s}$ with probability $q^{-(n-k)}$. Considering that $S_{\gamma,t}$ contains $\binom{n}{t}(2\gamma)^t$ vectors, and that at least one solution always exists by hypothesis, we obtain the result in (1). For the maximum support size version of R-SDP (i.e., R-SDP$_{\gamma,\leq t}$), we instead consider $\mathbf{s} \xleftarrow{\$} \left\{\mathbf{eH}^\top,\ \mathbf{e} \in B_{\gamma,t}\right\}$, and consequently estimate the number of solutions as

$$N_{\gamma,\leq t} = 1 + \frac{|B_{\gamma,t}| - 1}{q^{n-k}} \approx 1 + \sum_{i=0}^{t} \binom{n}{i} 2^{i\left(1+\log_2(\gamma)\right) - (n-k)\log_2(q)}. \tag{2}$$

We will furthermore distinguish between two cases, depending on the relation between t and n: in the so-called *small support case* we have $t \ll n$, while in the *large support case* we have that t is close to n. For space reasons, the proofs relevant to this section are reported in Appendix B.

5.1 Solving the R-SDP with fixed and small support

When the support of the searched vector is small, one may solve the R-SDP with Information Set Decoding (ISD), which are the best general solvers for the SDP in the Hamming metric. Prange's algorithm [28], which historically dates as the first ever proposed ISD algorithm, can be used to decode both binary and non-binary codes with essentially the same complexity (if we neglect the cost of linear algebra), and requires an information set where the searched vector does not contain non-null coordinates. During the years, many improved ISD algorithms have been proposed, aimed at reducing the computational cost (see [4] for an overview of algorithms for the binary field). Roughly, the main idea is that of increasing the guessing probability by allowing the presence of some non-null entries in the information set, and then proceed to identify them through enumeration. Combining this idea with collision search techniques, the complexity of ISD can be significantly reduced (see and [8], [26, 20] for the state-of-the-art in the binary and non-binary cases, respectively).

Let us now consider the R-SDP with solution having support size $t \ll n$. In this case a solution for the R-SDP is also a solution for the corresponding SDP instance, so we can use any non-binary ISD algorithms without any modification. Yet, we can probably do better by taking into account some observations:

1. in the enumeration phase, one should take into account that the vector takes values in a subset of the underlying finite field. This leads to a polynomial reduction in the ISD complexity, with respect to the Hamming metric case;
2. when the set entries of a candidate vector are not equally distributed over $\{\pm 1, \dots, \pm\gamma\}$, then we can further speed-up the enumeration phase. Indeed, if we know that some values are very unlikely to appear, then we can remove them from the search, achieving another advantage in the cost.

Based on the above considerations, the complexity of solving R-SDP$_{\gamma,=t}$ with small support can be bounded between the cost of binary ISD algorithms (as lower bound) and that of non-binary ISD algorithms (as upper bound) for the same support size. Indeed, the easiest R-SDP instance is the one in which the set entries of the solution have all the same value (say, are all equal to 1). In such a case, the problem is identical to the binary SDP, with the only exception that the considered code lives in a non-binary finite field. Notice that the presence of a non-binary finite field is supposed to somehow increase the cost, with respect to the binary case: for instance, the cost of linear algebra becomes larger. Employing the well-known approximation for the cost of a binary ISD from [32], we can then conservatively assume that solving R-SDP with small support costs at least

$$C_{ISD}(n,k,t) = 2^{-t\log_2(1-k/n)-\log_2(N_{\gamma,=t})}, \tag{3}$$

where the term $\log_2(N_{\gamma,=t})$ takes into account the existence of multiple solutions. Notice that, if $t \ll n$, then $N_{\gamma,=t} \approx 1$, so that the reduction in the complexity does not take place.

5.2 Solving the R-SDP with large support

When the support size of the solution is rather large (say, close to n), ISD algorithms become ineffective, since the searched vector does not contain a large number of null entries. We here generalize the approach of [5], which solves the R-SDP with fixed and maximum support, for the sole case of $\gamma = 1$.

To this end, we propose to use Algorithm 1. Basically, the algorithm first

Input: $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $\ell \in [0; n-k]$
Output: $\mathbf{e} \in B_{\gamma,t}$ such that $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$

```
1 Pick a random permutation π.
2 Find A such that Aπ(H) = [ I_{n−k−ℓ}        H′ ∈ F_q^{(n−k−ℓ)×(k+ℓ)} ]
                          [ 0_{ℓ×(n−k−ℓ)}    H″ ∈ F_q^{ℓ×(k+ℓ)}       ] ; if it is not
     possible, restart from line 1.
3 Compute [s′, s″] = As, with s′ ∈ F_q^{n−k−ℓ} and s″ ∈ F_q^ℓ.
4 Produce a set E_{γ,ℓ} ∈ S_{γ,k+ℓ} of solutions to the R-SDP_{γ,k+ℓ} instance
     represented by {s″, H″}.
5 for e″ ∈ E_{γ,ℓ} do
6 |  Compute e′ = s′ − e″H″^⊤
7 |  if e′ ∈ S_{γ,t−k−ℓ} then
8 |  |  return π^{−1}([e′, e″])
9 Restart from line 1.
```

Algorithm 1: PGE+SS approach to solve R-SDP$_{\gamma,t}$

brings the given parity-check matrix into partially row-reduced echelon form, via a column permutation π and row operations described by the full-rank matrix $\mathbf{A}$. The same transformation is applied to the syndrome $\mathbf{s}$ (line 3 of the Algorithm), in order to obtain a length-ℓ sub-syndrome $\mathbf{s}''$ which, together with $\mathbf{H}''$, is given as input to a R-SDP$_{\gamma,k+\ell}$ solver (line 4 of the Algorithm). Finally, the found solutions, which are grouped in a set $E_{\gamma,\ell}$, are tested aiming to produce a solution to the initial R-SDP$_{\gamma,=t}$ instance (lines 5–8 in the Algorithm).

Proposition 2. *Let $\{\mathbf{H},\mathbf{s}\} \in \mathbb{F}_q^{(n-k)\times n} \times \mathbb{F}_q^{n-k}$ be an R-SDP$_{\gamma,\leq t}$ instance, with $\mathbf{H}$ being a parity-check matrix of a random code $\mathscr{C} \subseteq \mathbb{F}_q^n$ with dimension k, and $\mathbf{s} \xleftarrow{\$} \left\{\mathbf{e}\mathbf{H}^\top \mid \mathbf{e} \in B_{\gamma,t}\right\}$. Then, Algorithm 1 solves R-SDP$_{\gamma,=t}$ with an average cost of*

$$O\left(\frac{T(t,\gamma,\ell) + \frac{M(t,\gamma,\ell)}{1+\widetilde{M}(t,\gamma,\ell)}}{\eta(t,\gamma,\ell)\left(1 - \prod_{i=k+\ell}^{t}\left(1 - \binom{i}{k+\ell}/\binom{n}{k+\ell}\right)^{N_{\gamma,=i}}\right)}\right),$$

where $N_{\gamma,=i}$ and $N_{\gamma,\leq t}$ are as in (1), and (2), $T(t,\gamma,\ell)$ is the average cost of an algorithm that produces $M(t,\gamma,\ell)$ solutions to an instance of R-SDP$_{\gamma,=k+\ell}$, and $\widetilde{M}(t,\gamma,\ell)$ out of these solutions lead to a success of Algorithm 1. Finally, $\eta(t,\gamma,\ell)$ denotes the probability that $\widetilde{M}(t,\gamma,\ell)$ is not null.

To conclude the analysis, we have to consider the cost of solving the small R-SDP$_{\gamma,k+\ell}$ instance represented by $\{\mathbf{H}'', \mathbf{s}''\}$. As in [5], we consider the application of Wagner's algorithm [34], originally proposed as a solver for the subset sum problem. Among the solutions to the problem, we assume that

1. the number of *good solutions*, i.e., vectors leading to a success for Algorithm 1, is given by
$$U_{\gamma,t,\ell} = \sum_{i=k+\ell}^{t} \frac{\binom{i}{k+\ell}}{\binom{n}{k+\ell}} N_{\gamma,=i}; \tag{4}$$

2. the number of *bad solutions*, i.e. vectors that do not lead to a success of Algorithm 1, is given by
$$U'_{\gamma,y,\ell} = \max\left\{0\ ,\ q^{-\ell}\left((2\gamma)^{k+\ell} - U_{\gamma,t,\ell}\right)\right\}. \tag{5}$$

Wagner's algorithm on a levels to solve the R-SDP with maximum weight associated to $\{\mathbf{H}'', \mathbf{s}''\}$ is detailed in Algorithm 2. For the sake of simplicity, we assume that $k+\ell$ is a multiple of 2^a; $\mathbf{H}''_i$, for $i \in [0; 2^a - 1]$, denotes the matrix formed by the columns of $\mathbf{H}''$ in the positions $\{i\frac{k+\ell}{2^a}, \ldots, (i+1)\frac{k+\ell}{2^a} - 1\}$. The merging operation between two lists, which we denote as $\mathcal{L}_{2j}^{(i)} \sqcap_{u_i} \mathcal{L}_{2j+1}^{(i)}$, is defined as follows

$$\{(\mathbf{z}_{2j} + \mathbf{z}_{2j+1}, [\mathbf{p}_{2j}, \mathbf{p}_{2j+1}]) \mid (\mathbf{z}_b, \mathbf{p}_b) \in \mathcal{L}_b^{(i)}, \mathbf{z}_{2j} + \mathbf{z}_{2j+1} = \mathbf{0} \text{ in the last } u_i \text{ entries}\}.$$

Similarly to [5, Proposition 16], in the following proposition we assess the com-

Input: $\mathbf{H}''_0, \cdots, \mathbf{H}''_{2^a-1} \in \mathbb{F}_q^{\ell \times \frac{k+\ell}{2^a}}$, $\mathbf{s}'' \in \mathbb{F}_q^{\ell}$
Output: A list $\mathcal{L}_0^{(a)} = \left\{\left(\mathbf{p}\mathbf{H}''^\top\ ,\ \mathbf{p}\right)\right\}$ such that $\mathbf{p} \in \{\pm 1, \ldots, \pm\gamma\}^{k+\ell}$ and $\mathbf{p}\mathbf{H}''^\top = \mathbf{s}''$
Data: $v, a \in \mathbb{N}$, with $a \geq 1$ and $v \leq \frac{k+\ell}{2^a}$, and positive integers $0 < u_1 < \cdots < u_{a-1} < \ell$.

```
Set u_0 = −1, u_a = ℓ.
Choose random subsets R_0, ··· , R_{2^a−1} ⊆ {±1, ..., ±γ}^{(k+ℓ)/2^a}, each of size (2γ)^v.
Build the lists L_j^(0) = {(z = p H''_j^T, p) | p ∈ R_j} for j ∈ [0; 2^a − 2].
Build the list L_{2^a−1}^(0) = {(z = p H''_{2^a−1}^T − s'' , p) | p ∈ R_{2^a−1}}.
for i = 1 to a do
    for j = 0 to 2^{a−i} − 2 do
        L_j^(i+1) = L_{2j}^(i) ⊓_{u_i} L_{2j+1}^(i)
return L_0^(a)
```

Algorithm 2: Wagner's algorithm structured on a levels

plexity of using Wagner's algorithm to find one of the desired solutions.

Proposition 3. *Let* $\{\mathbf{H}'', \mathbf{s}''\} \in \mathbb{F}_q^{\ell\times(k+\ell)} \times \mathbb{F}_q^{\ell}$ *be an R-SDP*$_{\gamma,=k+\ell}$ *instance with* $U_{\gamma,t,\ell}$ *good solutions and* $U'_{\gamma,t,\ell}$ *bad solutions. Assume to run Algorithm 2 on a levels, with options* $0 < u_1 < \cdots < u_{a-1} < \ell$.
Let $\rho = 2^{\left(2^a v-k-\ell\right)\left(1+\log_2(\gamma)\right)-\log_2(q)\sum_{i=1}^{a-1} u_i 2^{a-1-i}}$. *Then, the computational complexity used by Wagner's algorithm is given by*

$$T(k,\ell,\gamma) = \max_{i\in[0;a-1]} \left\{2^{a-i} L_i\right\},$$

where $L_i = \begin{cases} 2^{v\left(1+\log_2(\gamma)\right)} & \text{if } i=0, \\ 2^{2^i v\left(1+\log_2(\gamma)\right)-\log_2(q)\left(u_i+\sum_{j=1}^{i-1} u_j 2^{i-1-j}\right)} & \text{otherwise.}\end{cases}$
The algorithm finds good solutions with probability $\eta(t,\gamma,\ell) = 1-(1-\rho)^{U_{\gamma,t,\ell}}$, *and on average outputs* $M(t,\gamma,\ell) = \rho\left(U_{\gamma,t,\ell} + U'_{\gamma,t,\ell}\right)$ *solutions, with* $\widetilde{M}(t,\gamma,\ell) = \rho U_{\gamma,t,\ell}$ *of them being good.*

Remark 1. In principles, one may also rely on *representations* to solve R-SDP. Namely, we still express a solution $\mathbf{e}$ to R-SDP as $\mathbf{e} = \sum_i \mathbf{e}_i$ and seek for the terms $\mathbf{e}_i$, but we allow for some overlapping among the supports of each $\mathbf{e}_i$. In the most simple application of this criterion, we can express $\mathbf{e}$ as $\mathbf{e}_0+\mathbf{e}_1$, where $\mathbf{e}_0, \mathbf{e}_1$ are generic vectors over $\mathbb{F}_q^{k+\ell}$. This way, we have $q^{k+\ell}$ representations for $\mathbf{e}$. When there are $N_{\gamma,=k+\ell}$ solutions to R-SDP, we are satisfied as soon as we find one of the $N_{\gamma,=k+\ell}q^{k+\ell}$ representations. To search for one of them, we can produce two lists $\mathcal{L}_0 = \left\{\left(\mathbf{e}_0, \mathbf{H}^\top - \mathbf{s}\right),\ \mathbf{e}_0 \in \mathcal{R}_0\right\}$, $\mathcal{L}_1 = \left\{\left(\mathbf{e}_1, \mathbf{H}^\top\right),\ \mathbf{e}_1 \in \mathcal{R}_1\right\}$, where $\mathcal{R}_0, \mathcal{R}_1 \subseteq \mathbb{F}_q^{k+\ell}$ and have size $q^{v'}$. The computational cost to perform this search can be estimated as $q^{v'}$. Notice that we hit a representation with probability that can be roughly estimated as $1-\left(1-\frac{N_{\gamma,=k+\ell}q^{v}}{q^{2k+\ell}}\right)^{q^{2v'}}$. We would like to keep v' as low as possible, otherwise the complexity of this approach becomes intractable: hence, the previous probability can be approximated by $\eta' = N_{\gamma,=k+\ell}q^{3v'-2(k+\ell)}$. To have a fair comparison with Algorithm 2, we consider just one level and lists of the same size, thus choose $v = v'\log_2(q)/\left(1+\log_2(\gamma)\right)$. The corresponding success probability of Wagner's algorithm is given by $\eta = N_{\gamma,=k+\ell}2^{(2v-k-\ell)\left(1+\log_2(\gamma)\right)}$. With simple computations, we find that $\eta' > \eta$ when

$$v' > (k+\ell)\left(2 - \frac{1+\log_2(\gamma)}{\log_q}\right) > k+\ell,$$

where the last inequality comes from the consideration that $2\gamma < q$. Clearly, $v' > k+\ell$ is an absurd: this proves that Wagner's algorithm with one level always outperforms the simple representations technique, on one level. It is clear that there may be improved ways to use representations and, as in [11], one can use hybrid approaches where some layers of Wagner's algorithm actually use some kind of representations. Yet, as we have briefly explained here, at a first glance it appears that this choice does not lead to significant improvements in the complexity to solve R-SDP.

6 Security and practical instances

Due to lack of space, we do not provide a formal proof of security, but only some hints at how such a proof should work, by highlighting possible attack strategies. This also allows us to show how secure system parameters can be designed.

One-wayness of key generation Each row of the public $\mathbf{S}$ is the syndrome of the corresponding row of the secret $\mathbf{E}$. Notice that, in the key generation algorithm, we guarantee that the rows have minimum support size t_E, with $t_E \ll n$. Since the entries of $\mathbf{E}$ are either null or equal to ± 1, we have that finding each row of $\mathbf{E}$ can be seen as facing an R-SDP$_{1,\leq t^*}$ instance, for some $t^* \geq t_E$. As we have highlighted in Section 5, the computational complexity to solve R-SDP in case of a small support size t^*, for a code $\mathscr{C}$, is not lower than that of solving the Hamming SDP for a code with same length and dimension of $\mathscr{C}$, but defined over the binary finite field, searching for a vector with Hamming weight t^*. As (3) shows, the complexity grows exponentially with the weight of the searched vector. Hence, conservatively, we assess the complexity of attacks aimed at recovering the secret key as $2^{-t_E \log_2(r/n) - \log_2(N_{1,t_E})}$.

Unforgeability To forge a signature, an attacker may proceed as follows. First, he picks a random $\mathbf{y}$ with the desired distribution and computes $\mathbf{s}_y = \mathbf{y}\mathbf{H}^\top$. Then, he sets $\mathbf{c} = \mathsf{Hash}(m, \mathbf{s}_y)$ and computes $\mathbf{s}_z = \mathbf{s}_y + \mathbf{c}\mathbf{S}$. If he is able to produce a vector $\mathbf{z} \in B_{\gamma,n}$ such that $\mathbf{z}\mathbf{H}^\top = \mathbf{s}_z$, then the pair $\{\mathbf{c}, \mathbf{z}\}$ can be used as a valid signature. Notice that, to do this, he must solve an R-SDP$_{\bar{\gamma},\leq n}$ instance. To assess the hardness of this attack, we rely on Proposition 2.

Unfeasibility of noise recovery Assume that the adversary is able to retrieve $\mathbf{y}$ from $\mathbf{s}_y$. If he succeeds, he can then compute $\mathbf{z} - \mathbf{y} = \mathbf{c}\mathbf{E}$ and, since $\mathbf{c}$ is known, retrieve information on $\mathbf{E}$. Exploiting the sparsity of both $\mathbf{c}$ and $\mathbf{E}$, the rows of $\mathbf{E}$ can be trivially recovered. Then, we have to guarantee that recovering $\mathbf{y}$ is unfeasible. Again, this reduces to the problem of solving an R-SDP$_{\gamma,\leq n}$ instance (hence, we use Proposition 2 to estimate the complexity of attacks of this kind).

Statistical indistinguishability Finally, we consider the possibility for an attacker to retrieve some information about the secret key by performing a statistical analysis on a bunch of collected honest signatures. This is motivated by the fact that the same private key is used to construct many signatures. To describe how such a dependence can be exploited, consider a collection of signatures for which the digests $\mathbf{c}$ have a common set entry. To analyze this situation, we assume for simplicity that such an entry is the first one, and that $\bar{\gamma} = \gamma$. In such a case, the first row of $\mathbf{E}$ contributes to all the collected signatures, and its entries can be recovered through a statistical analysis. Indeed, in the positions $i \in [0; n-1]$ such that $e_{0,i} = \pm 1$, we have that the i-th entry of the signatures takes a value equal to $\pm\bar{\gamma}$ with a probability that is slightly smaller than $(2\bar{\gamma}+1)^{-1}$. If $e_{0,i} = 0$, on the contrary, this statistical bias is not present. Hence, collecting a sufficient number of signatures with such digests would be enough to recover the first row of $\mathbf{E}$. Thanks to the rejection sampling in the signing algorithm, it is enough to choose $\bar{\gamma} \leq \gamma - 1$ to prevent this type of weakness. Yet, the attack can be generalized by considering the occurrence, in the digests, of specific tuples of size larger than $\gamma - \bar{\gamma}$. Indeed, a pattern of this size may be such that its product with a column of $\mathbf{E}$ yields a value larger than $\gamma - \bar{\gamma}$ (or lower than $-(\gamma - \bar{\gamma})$): in such a case, a statistical bias in the signatures appears again, and can somehow be exploited to recover information about the secret $\mathbf{E}$. To completely prevent from this kind of attacks, it is enough to choose the values of γ and $\bar{\gamma}$ so that all the values of $\mathbf{cE}$ are outside $\{0, \pm 1, \ldots, \pm(\gamma - \bar{\gamma})\}$ with probability less than $2^{-\lambda}$, where λ is the desired security level in bits. Notice that such a probability equals $1 - \left(\sum_{i=-(\gamma-\bar{\gamma})}^{\gamma-\bar{\gamma}} \tilde{g}_{w_E, w_c, q, b, \gamma}(i)\right)^n$.

6.1 System parameters

In order to design secure parameters for the new scheme, we must first guarantee that the number of possible digests is sufficiently large, that is, $\binom{b}{w_c} 2^{w_c} > 2^{2\lambda}$. To set t_E, we consider (3) and derive the minimum value of t_E guaranteeing that the complexity of ISD is larger than 2^λ. The average number of times the key generation algorithm has to be repeated, before obtaining a matrix $\mathbf{E}$ where each row has support size at least t_E, is given by $\left(1 - \sum_{i=0}^{t_E - 1} \binom{n}{i} \left(\frac{w_E}{b}\right)^i \left(1 - \frac{w_E}{b}\right)^{n-i}\right)^{-1}$. We choose w_E so that the previous quantity is sufficiently small. Taking all of this into account, in Table 1 we provide some parameters sets achieving $\lambda = 128$ bits of classical security. As we see from the table, the scheme achieves very compact signatures and public keys, and also enables flexible trade-offs between different parameter choices.

(n, k, q)	b	w_E	w_c	t_E	γ	$\bar{\gamma}$	Avg rejections	$\lvert\sigma\rvert$ in kB	$\lvert\mathsf{pk}\rvert$ in kB
$(400, 300, 16381)$	218	46	67	64	3420	3375	199.80	0.72	38.15
$(500, 375, 16381)$	250	42	61	64	3890	3849	199.78	0.88	54.69
$(400, 320, 16381)$	240	45	63	56	3460	3417	148.64	0.72	33.60
$(500, 375, 32749)$	260	44	60	64	4600	4559	87.88	0.90	65.99
$(600, 400, 32749)$	240	42	63	81	5370	5329	99.30	1.09	89.95

Table 1. Instances achieving 128 bits of classical security. For all the considered parameter sets, the rejection rate in the key generation algorithm is lower than 0.1.

7 Conclusion

We have proposed a novel adaptation of the Schnorr-Lyubashevsky approach to the design of digital signature schemes based on codes. By relying on vectors with restricted entries, the proposed scheme is able to withstand known cryptanalysis approaches, while achieving very compact signatures and keys. In this first proposal, we have considered random, non-structured codes, which allow relying on the general formulation of the corresponding decoding problems for the security of the scheme. A formal security proof, along with the study of variants adopting structured codes, is left for future works.

References

[1] C. Aguilar, P. Gaborit, and J. Schrek. "A new zero-knowledge code based identification scheme with reduced communication". In: *2011 IEEE Information Theory Workshop (ITW)*. Paraty, Brazil, Oct. 2011, pp. 648–652.

[2] N. Aragon et al. *Cryptanalysis of a code-based full-time signature*. submitted to DCC. 2020. arXiv: `2011.08326 [cs.CR]`.

[3] N. Aragon et al. "Durandal: A Rank Metric Based Signature Scheme". In: *Advances in Cryptology – EUROCRYPT 2019*. Ed. by Y. Ishai and V. Rijmen. Cham: Springer International Publishing, 2019, pp. 728–758.

[4] M. Baldi et al. "A Finite Regime Analysis of Information Set Decoding Algorithms". In: *Algorithms* 12.10 (2019).

[5] M. Baldi et al. *A New Path to Code-based Signatures via Identification Schemes with Restricted Errors*. 2021. arXiv: `2008.06403 [cs.CR]`.

[6] M. Baldi et al. *Cryptanalysis of a code-based signature scheme without trapdoors*. Cryptology ePrint Archive, Report 2021/134. `https://eprint.iacr.org/2021/134`. 2021.

[7] M. Baldi et al. "Using LDGM Codes and Sparse Syndromes to Achieve Digital Signatures". In: *Post-Quantum Cryptography*. Ed. by P. Gaborit. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–15.

[8] A. Becker et al. "Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding". In: *Advances in Cryptology - EUROCRYPT 2012*. Ed. by D. Pointcheval and T. Johansson. Vol. 7237. Lecture Notes in Computer Science. Springer Verlag, 2012, pp. 520–536.

[9] E. Bellini et al. "Improved Veron Identification and Signature Schemes in the Rank Metric". In: *2019 IEEE International Symposium on Information Theory (ISIT)*. Paris, France, 2019, pp. 1872–1876.

[10] J.-F. Biasse et al. "LESS is More: Code-Based Signatures Without Syndromes". In: *Progress in Cryptology - AFRICACRYPT 2020*. Ed. by A. Nitaj and A. Youssef. Cham: Springer International Publishing, 2020, pp. 45–65.

[11] R. Bricout et al. "Ternary Syndrome Decoding with Large Weight". In: *Selected Areas in Cryptography – SAC 2019*. Ed. by K. G. Paterson and D. Stebila. Cham: Springer International Publishing, 2020, pp. 437–466.

[12] P.-L. Cayrel, P. Véron, and S. M. El Yousfi Alaoui. "A zero-knowledge identification scheme based on the q-ary syndrome decoding problem". In: *Selected Areas in Cryptography*. Springer Berlin Heidelberg, 2011, pp. 171–186.

[13] N. T. Courtois, M. Finiasz, and N. Sendrier. "How to Achieve a McEliece-Based Digital Signature Scheme". In: *Advances in Cryptology - ASIACRYPT 2001, Lecture Notes in Computer Science* 2248 (2001), pp. 157–174.

[14] T. Debris-Alazard, N. Sendrier, and J.-P. Tillich. "Wave: A new family of trapdoor one-way preimage sampleable functions based on codes". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2019, pp. 21–51.

[15] J.-C. Deneuville and P. Gaborit. "Cryptanalysis of a code-based one-time signature". In: *Designs, Codes and Cryptography* 88.9 (2020), pp. 1857–1866.

[16] L. Ducas et al. "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018.1 (Feb. 2018), pp. 238–268.

[17] S. M. El Yousfi Alaoui et al. "Code-Based Identification and Signature Schemes in Software". In: *Security Engineering and Intelligence Informatics*. Ed. by A. Cuzzocrea et al. Springer Berlin Heidelberg, 2013, pp. 122–136.

[18] J. Faugère et al. "A distinguisher for high rate McEliece cryptosystems". In: *2011 IEEE Information Theory Workshop*. 2011, pp. 282–286.

[19] A. Fiat and A. Shamir. "How to prove yourself: Practical solutions to identification and signature problems". In: *CRYPTO*. Springer. 1986, pp. 186–194.

[20] C. Interlando et al. "Generalization of the Ball-Collision Algorithm". In: *CoRR* (2018). arXiv: `1812.10955`. URL: `http://arxiv.org/abs/1812.10955`.

[21] G. Kabatianskii, E. Krouk, and B. J. M. Smeets. "A Digital Signature Scheme Based on Random Error-Correcting Codes". In: *IMA Int. Conf.* 1997, pp. 161–167.

[22] Z. Li, C. Xing, and S. L. Yeo. *A New Code Based Signature Scheme without Trapdoors*. Cryptology ePrint Archive, Report 2020/1250. 2020.

[23] V. Lyubashevsky. "Lattice signatures without trapdoors". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2012, pp. 738–755.

[24] A. Otmani and J.-P. Tillich. "An Efficient Attack on All Concrete KKS Proposals". In: *Post-Quantum Cryptography*. Ed. by B.-Y. Yang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 98–116.

[25] E. Persichetti. "Efficient One-Time Signatures from Quasi-Cyclic Codes: A Full Treatment". In: *Cryptography* 2.4 (2018).

[26] C. Peters. "Information-Set Decoding for Linear Codes over Fq". In: *Post-Quantum Cryptography*. Ed. by N. Sendrier. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 81–94.

[27] A. Phesso and J.-P. Tillich. "An Efficient Attack on a Code-Based Signature Scheme". In: *Post-Quantum Cryptography*. Ed. by T. Takagi. Cham: Springer International Publishing, 2016, pp. 86–103. ISBN: 978-3-319-29360-8.

[28] E. Prange. "The use of information sets in decoding cyclic codes". In: *IRE Trans. Inf. Theory* 8.5 (Sept. 1962), pp. 5–9.

[29] P. Santini, M. Baldi, and F. Chiaraluce. "Cryptanalysis of a One-Time Code-Based Digital Signature Scheme". In: *2019 IEEE International Symposium on Information Theory (ISIT)*. Paris, France, 2019, pp. 2594–2598.

[30] Y. Song et al. "A code-based signature scheme from the Lyubashevsky framework". In: *Theoretical Computer Science* 835 (2020), pp. 15–30.

[31] J. Stern. "A new identification scheme based on syndrome decoding". In: *Advances in Cryptology — CRYPTO' 93*. Ed. by D. R. Stinson. Springer Berlin Heidelberg, 1994, pp. 13–21.

[32] R. Torres and N. Sendrier. "Analysis of Information Set Decoding for a Sublinear Error Weight". In: *PQCrypto 2016*. Springer International Publishing, 2016, pp. 144–161.

[33] P. Véron. "Improved identification schemes based on error-correcting codes". In: *Applicable Algebra in Engineering, Communication and Computing* 8.1 (1997), pp. 57–69.

[34] D. Wagner. "A Generalized Birthday Problem". In: *Advances in Cryptology — CRYPTO 2002*. Ed. by M. Yung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 288–304.

Appendix A - Proofs of Section 4

Proof of Lemma 1

Proof. The probability is the same for β and $q-\beta$, so we only consider the case of $\beta \in [0; \lfloor q/2 \rfloor]$, for which $\beta_q = \beta$. Let u_1 denote the number of indexes i for which $c_i e_i = 1$, and u_{-1} denote that of indexes for which $c_i e_i = -1$. To have $\langle \mathbf{c} \; ; \; \mathbf{e} \rangle = \beta$, it must $u_1 - u_{-1} = \beta$. Let $v = u_1 + u_{-1}$, that is, the number of intersections between the support of $\mathbf{c}$ and that of $\mathbf{e}$. To have $\langle \mathbf{c} \; ; \; \mathbf{e} \rangle = \beta$, the following two conditions must be verified:

- we have $\max\{0 \; , \; w_c + w_E - \ell\} \leq v \leq \min\{w_E \; , \; w_c\}$;
- since $u_{-1} = v - u_1$, it must be $u_1 = \frac{v+\beta}{2}$.

Hence, we obtain the following probability:

$$\Pr\left[\langle \mathbf{a} \; ; \; \mathbf{b} \rangle = \beta\right] = \sum_{\substack{v=\beta \\ v \text{ and } \beta \text{ have the same parity}}}^{\min\{w_E \; , \; w_c\}} 2^{-v} \frac{\binom{v}{\frac{v+\beta}{2}}\binom{w_E}{v}\binom{\ell - w_E}{w_c - v}}{\binom{\ell}{w_c}}.$$

□

Proof of Lemma 2

Proof. The proof is straightforward. Indeed, assume that $y = x$, which happens with probability $1/(2\alpha+1)$. To have $\langle \mathbf{c} \; ; \; \mathbf{e} \rangle + x = \beta$, it must be $\langle \mathbf{c} \; ; \; \mathbf{e} \rangle = \beta - x$. Summing over all possible values of x, we obtain the formula in the thesis. □

Appendix B - Proofs of Section 5

Proof of Proposition 2

For a vector $\mathbf{e} \in S_{\gamma,i}$ for which $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$, the probability that π is such that the vector $\mathbf{e}''$ formed by the last $k+\ell$ entries of $\pi(\mathbf{e})$ are all non null is given by $\epsilon_i = \binom{i}{k+\ell}/\binom{n}{k+\ell}$. Notice that, if $\mathbf{e}''$ has a different support size, then it will never be in the set $E_{\gamma,\ell}$ produced in Line 4 and, hence, we have that $\mathbf{e}$ is never returned as output from Algorithm 1. The number of solutions with support equal to i can be estimated as $N_{\gamma,=i}$, so that the probability that π is not valid for all of them is obtained as $(1-\epsilon_i)^{N_{\gamma,=i}}$. Multiplying over the values of i from $k+\ell$ to t, and taking the complementary, we derive the probability that π is valid for at least one out of the $N_{\gamma,\leq t}$ solutions. We multiply this probability by $\eta(t,\gamma,\ell)$ and obtain the success probability of one iteration of Algorithm 1.
Let $T(t,\gamma,\ell)$ denote the cost of an R-SDP$_{\gamma,=k+\ell}$ solver (i.e., for the instance represented by $\{\mathbf{H}'', \mathbf{s}''\}$), producing a set of solutions $E_{\gamma,\ell}$ that on average contains $M(t,\gamma,\ell)$ elements. Since there are $\widetilde{M}(t,\gamma,\ell)$ actually valid solutions (i.e., leading to the success of Algorithm 1), on average we test $M(t,\gamma,\ell)/\left(1+\widetilde{M}(t,\gamma,\ell)\right)$ vectors from $E_{\gamma,\ell}$, before Algorithm 1 successfully halts. Thus, we estimate the cost of executing lines 4–8 as $T(t,\gamma,\ell) + M(t,\gamma,\ell)/\left(1+\widetilde{M}(t,\gamma,\ell)\right)$.

Proof of Proposition 3

Let $\mathbf{x} \in \{\pm 1, \ldots, \pm\gamma\}^{k+\ell}$ be a solution, i.e., such that $\mathbf{x}\mathbf{H}''^{\top} = \mathbf{s}''$. Wagner's algorithm will output $\mathbf{x}$ among the vectors in the final list iff

a) $\mathbf{x} \in \mathcal{R}_0 \times \mathcal{R}_1 \times \cdots \times \mathcal{R}_{2^a-1}$;
b) at each merge, the vector $\mathbf{x}$ is not filtered.

Condition a) is verified with probability $\left(\frac{(2\gamma)^v}{(2\gamma)^{\frac{k+\ell}{2^a}}}\right)^{2^a} = 2^{(2^a v-k-\ell)\left(1+\log_2(\gamma)\right)}$. We now proceed by computing the probability that also condition b) happens, assuming condition a) holds. If $a=1$, then we have no filtering, while in the other cases it may happen in the levels from the first to the $(a-1)$-th one. To this end, we consider the i-th level (for $i \in [1;a]$), and divide $\mathbf{x}$ into 2^{a-i+1} chunks, each formed by $\frac{k+\ell}{2^{a-i+1}}$ consecutive entries, which we denote as $\mathbf{x}_j$, for $j \in [0; 2^{a-i}-1]$. In the i-th level, $\mathbf{x}$ will not be filtered if and only if

1. for $j \in [0; 2^{a-i}-2]$, $\mathbf{x}_{2j}\mathbf{H}''^{\top}_{2j} + \mathbf{x}_{2j+1}\mathbf{H}''^{\top}_{2j+1}$ is null in the last $u_i - u_{i-1}$ positions;
2. $\mathbf{x}_{2^{a-i+1}-2}\mathbf{H}''^{\top}_{2^{a-i+1}-2} + \mathbf{x}_{2^{a-i+1}-1}\mathbf{H}''^{\top}_{2^{a-i+1}-1} - \mathbf{s}''$ is null in the last $u_i - u_{i-1}$ positions.

Note that if condition 1 is met, then condition 2 is met as well, so we just have to consider the probability with which condition 1 happens. Given that both $\mathbf{H}''$ and $\mathbf{x}$ are random, in each merge, chunks $\mathbf{x}_{2j}$ and $\mathbf{x}_{2j+1}$ will not be filtered out with probability $q^{-(u_i-u_{i-1})}$. Given that, for $j \in [0; 2^{a-i}-2]$, we perform $2^{a-i}-1$ merges, condition 1 is verified with probability

$$\left(q^{-(u_i-u_{i-1})}\right)^{2^{a-i}-1} = 2^{-(2^{a-i}-1)(u_i-u_{i-1})\log_2(q)}.$$

Thus, the probability of $\mathbf{x}$ surviving till the last level is obtained as

$$\prod_{i=1}^{a-1}\left(q^{-(u_i-u_{i-1})}\right)^{2^{a-i}-1} = 2^{-\log_2(q)\sum_{i=1}^{a-1} u_i 2^{a-1-i}}.$$

Putting everything together, we get that each solution $\mathbf{x}$ is found with probability

$$\rho = 2^{(2^a v-k-\ell)\left(1+\log_2(\gamma)\right)-\log 2(q)\sum_{i=1}^{a-1} u_i 2^{a-1-i}}.$$

The total number of produced solutions is then estimated as

$$M(k,\ell,\gamma) = \rho(U_{\gamma,t,\ell} + U'_{\gamma,t,\ell}),$$

while the success probability is simply obtained by considering the probability that at least one good solution survives till the end, that is

$$\eta(\gamma,t,\ell) = 1-(1-\rho)^{U_{\gamma,t,\ell}}.$$

We finally derive the computation complexity to execute Wagner's algorithm. To this end, we assume that the cost of each merge is equal to size of the lists that are merged. We consider the i-th level, for $i \in [1; a-1]$, and denote with L_{i-1} the average size of the input lists, and as L_i that of the produced ones. For the initial level (i.e., $i = 1$), we use lists of size $L_0 = (2\gamma)^v = 2^{v\left(1+\log_2(\gamma)\right)}$, and obtain lists with average size given by $L_1 = L_0^2/q^{u_1} = 2^{2v\left(1+\log_2(\gamma)\right)-u_1\log_2(q)}$. In the subsequent level (i.e., $i = 2$) the average size of the produced lists is given by $L_2 = L_1^2/(q^{u_2-u_1}) = 2^{4v\left(1+\log_2(\gamma)\right)-(u_1+u_2)\log_2(q)}$. If we iterate this reasoning, we get that for the i-th level, with $i \geq 2$, we have an average size of the input lists given by

$$L_i = 2^{v2^i\left(1+\log_2(\gamma)\right)-\log_2(q)\left(u_i+\sum_{j=1}^{i-1}u_j2^{i-1-j}\right)}.$$

Finally, we consider that the in the i-th level we perform 2^{a-i} merges, and hence assess the complexity of the algorithm as $\max_{i\in[0;a-1]}\left\{2^{a-1-i}L_i\right\}$.

Appendix C - Numerical validation of the PGE+SS framework analysis

To have a confirmation of the analysis we provide in Section 5, we have run numerical experiments on some codes with small parameters. In Figure 1 we compare the empirical values of $N_{\gamma,=t}$ with the theoretical ones, estimated through (1), for the case of restricted vectors with maximum support size. As we see from the figure, the theoretical values of $N_{\gamma,=t}$ closely match the empirical ones.

We have also verified the performances of Wagner's algorithm, as described in Algorithm 2 and theoretically analyzed in Proposition 3. In our experiments, we have generated random parity-check matrices $\mathbf{H} \in \mathbb{F}_q^{\ell\times(k+\ell)}$ in systematic form, and have picked syndromes as $\mathbf{s} \xleftarrow{\$} \{\mathbf{e}\mathbf{H}^\top,\ \mathbf{e} \in \{\pm1,\ldots,\pm\gamma\}^{k+\ell}\}$. We have given $\{\mathbf{H},\mathbf{s}\}$ as input to Wagner's algorithm, and have measured algorithm features such as the success probability, the number of produced outputs and the lists sizes through the levels. For each considered setting (that is, the values ℓ, k, q and γ and the options for Wagner's algorithm), we have repeated the experiments for a sufficiently large number of times; the obtained results have been averaged over all considered experiments, and have been compared with the theoretical analysis of Proposition 3. The obtained results are reported in Table 2.

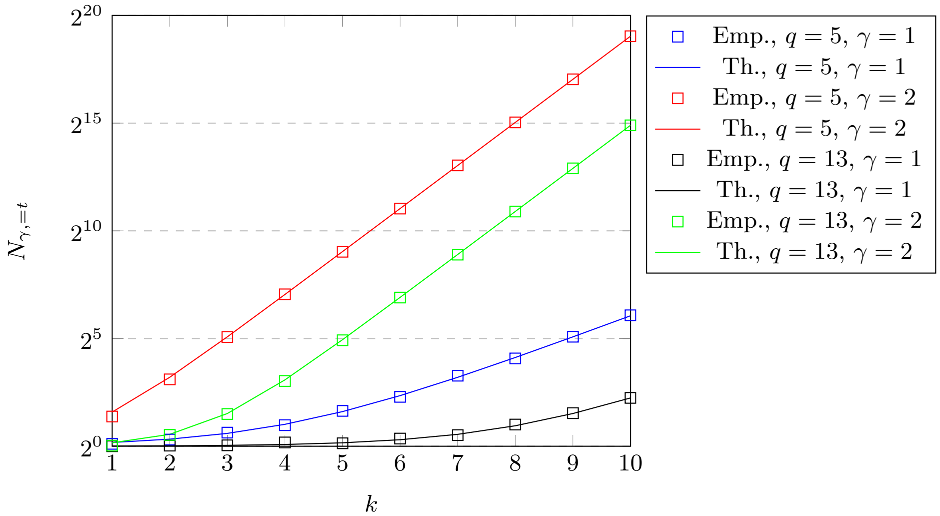

Fig. 1. Comparison between the numerical values of $N_{\gamma,=t}$ and the theoretical ones, estimated through (1). We have considered codes with redundancy $\ell = 3$, dimension k and length $k + \ell$, and have analyzed error vectors with support size $t = k + \ell$. For each setting and each value of k, we have considered 100 random codes.

(ℓ, k, q, γ)	v	a	$[u_1, \ldots, u_{a-1}]$	$\eta(k, \ell, \gamma)$	$M(k, \ell, \gamma)$	$[L_0, \ldots, L_{a-1}]$	# tests
$(16, 48, 13, 4)$	3	4	$[4\ ,\ 8\ ,\ 12]$	0	0	$[512\ ,\ 35.04\ ,\ 9.2\ ,\ 2.9 \cdot 10^{-3}\ ,\ 0]$	80,000
				$3.2 \cdot 10^{-24}$	$3.2 \cdot 10^{-24}$	$[512\ ,\ 35.06\ ,\ 9.2\ ,\ 2.9 \cdot 10^{-3}\ ,\ 3.0 \cdot 10^{-10}]$	-
$(10, 14, 11, 3)$	3	3	$[3\ ,\ 6]$	$6.8 \cdot 10^{-1}$	$6.8 \cdot 10^{-1}$	$[216\ ,\ 35.04\ ,\ 9.2 \cdot 10^{-1}]$	300,000
				$5.8 \cdot 10^{-1}$	$5.8 \cdot 10^{-1}$	$[216\ ,\ 35.05\ ,\ 9.2 \cdot 10^{-1}]$	-
$(6, 10, 11, 2)$	4	2	3	0.82253	1.8313	$[256\ ,\ 49.24053]$	100,000
				0.83845	1.8313	$[256\ ,\ 49.23817]$	-
$(6, 10, 11, 1)$	4	2	3	$7.7 \cdot 10^{-4}$	$7.7 \cdot 10^{-4}$	$[16\ ,\ 1.9 \cdot 10^{-1}]$	1,500,000
				$7.8 \cdot 10^{-4}$	$7.8 \cdot 10^{-4}$	$[16\ ,\ 1.9 \cdot 10^{-1}]$	-
$(7, 9, 11, 2)$	5	1	-	$5.4 \cdot 10^{-2}$	$5.6 \cdot 10^{-2}$	$[1024]$	9,000
				$5.3 \cdot 10^{-2}$	$5.5 \cdot 10^{-2}$	$[1024]$	-
$(13, 13, 11, 1)$	10	1	-	$1.5 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$[1024]$	9,000
				$1.5 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	$[1024]$	-

Table 2. Performances of Wagner's algorithm. For each considered setting (i.e., values ℓ, k, q, γ, v, a and $[u_1, \ldots, u_{a-1}]$) we have measured the performances through numerical experiments, and we have compared the obtained results with the theoretical analysis of Proposition 3. The results obtained via numerical simulations are marked by reporting (in the last column) the number of performed tests; for the theoretical estimates, instead, in the last column we have used the symbol **-**.