SPANSE: Combining sparsity with density for efficient one-time code-based digital signatures

note finali coverpage

(Article begins on next page)

30 July 2024

SPANSE: combining sparsity with density for efficient one-time code-based digital signatures

Marco Baldi, Franco Chiaraluce, and Paolo Santini

Università Politecnica delle Marche, Ancona, Italy

{m.baldi, f.chiaraluce, p.santini}@univpm.it

Abstract. The use of codes defined by sparse characteristic matrices, like QC-LDPC and QC-MDPC codes, has become an established solution to design secure and efficient code-based public-key encryption schemes, as also witnessed by the ongoing NIST post-quantum cryptography standardization process. However, similar approaches have been less fortunate in the context of code-based digital signatures, since no secure and efficient signature scheme based on these codes is available to date. The main limitation of previous attempts in this line of research has been the use of sparse signatures, which produces some leakage of information about the private key. In this paper, we propose a new code-based digital signature scheme that overcomes such a problem by publishing signatures that are abnormally dense, rather than sparse. This eliminates the possibility of deducing information from the sparsity of signatures, and follows a recent trend in code-based cryptography exploiting the hardness of the decoding problem for large-weight vectors, instead of its classical version based on small-weight vectors. In this study we focus on one-time use and provide some preliminary instances of the new scheme, showing that it achieves very fast signature generation and verification with reasonably small public keys.

Keywords: Code-based cryptography · Digital signatures · Large-weight decoding · QC-LDGM codes · QC-LDPC codes

1   Introduction

The problem of decoding a random-looking linear code is considered as one of the most understood and well established problems in post-quantum cryptography. Remarkably known to be NP-complete for both binary and non-binary codes [11,9], the decoding problem is at the basis of several secure and efficient encryption schemes, stemming from the original proposals of McEliece [22] and Niederreiter [25], or following more recent approaches [23,2]. As an evidence, three encryption schemes (ClassicMcEliece, BIKE and HQC) out of the seven currently admitted to the third round of the NIST post-quantum standardization process [24] are based on codes.

Differently from such a rather consolidated scenario for encryption schemes, the situation is different concerning digital signature schemes. 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 picking a decodable syndrome, yielding code-based schemes that are inefficient or insecure (or both). Two historical proposals along this line are CFS [14] and KKS [20], which however have important limitations. In fact, it is very difficult to find secure though efficient instances of the KKS scheme [26]. The CFS scheme is more consolidated, but requires Goppa codes with unpractical parameters yielding very large public keys. Moreover, Goppa codes with very high rate are required, which may expose the scheme to Goppa code distinguishers [17]. A recent important advance in this line of research is represented by the WAVE signature scheme [15], which however still requires a public key size in the order of 3 megabytes for 128 bits of classical security. Interestingly, WAVE avoids previous attacks against signature schemes derived from CFS by replacing Goppa codes with some new codes with a special $(U, U + V)$ structure and relies on the hardness of decoding large-weight vectors, instead of the classical decoding problem looking for small-weight vectors.

Other variants of CFS relying on codes different from Goppa codes have been proposed, but with less fortune. In [7], a digital signature scheme based on quasi-cyclic low-density generator matrix (QC-LDGM) codes was proposed, able to achieve very efficient signature generation and verification procedures, besides compact public keys. Unfortunately, the use of sparse signatures in such a scheme turned out to produce some leakage of information concerning the private key, which lead to successful cryptanalysis [27].

Opposed to the hash-and-sign paradigm used in the aforementioned signature schemes, a different approach to the design of code-based signatures is that of applying the Fiat-Shamir transform [18] to an identification scheme, which has the advantage of not relying on any trapdoor for key derivation. In fact, consolidated zero-knowledge code-based identification schemes exist since a long time [28], 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 [30,13,1,16,10,8,6,19,12], although their features are still far from being comparable with those of signature schemes relying on other mathematical objects, such as lattices [21].

1.1   Our contribution

We propose SPANSE, an evolution of the code-based signature scheme proposed in [7], following the hash-and-sign paradigm, with the aim of overcoming its main limitations and providing a new secure and efficient signature scheme based on a code-based trapdoor. The main innovations over [7] are as follows:

1. Codes and vectors are defined over $\mathbb{F}_q$, with $q > 2$ being a prime, while in [7] they were binary.
2. The transformation matrix S was sparse in [7], while it is chosen to be dense in the new scheme, with entries from a subset of the underlying finite field.

3. A transformation matrix Q was needed in [7] to hide the sparse nature of the other two secret matrices (H and S), while it is no longer required in the new scheme, owing to the dense nature of S. For this reason, the matrix Q is replaced with a simple permutation matrix P.
4. The generated signatures were sparse in [7], while they are dense and free of zero entries in the new scheme.

The above modifications allow avoiding attacks exploiting sparsity of signatures, which are no longer sparse, and make the scheme security rely on the hardness of decoding large-weight vectors instead of low-weight ones. Still, some tuning of the overall statistical distribution of produced signatures will be needed in order to make it indistinguishable from a random distribution when long-term keys are used. This, however, is left for future investigations, while in this paper we only focus on one-time keys. The public matrices in SPANSE still have a product structure like those used in the LEDA cryptosystems [4,5]. However, the dense nature of the multiplied matrices avoids the existence of weak keys like those identified in [3]. Codes in quasi-cyclic (QC) form like in [7] are used in the scheme we propose, and allow achieving public keys of reasonable size (in the order of 2 megabytes for 128 bits of classical security). Such a public key size may be further reduced by increasing the circulant block size, which we keep very limited in this preliminary study.

One important advantage of the new scheme stemming from [7] is that it exploits a straightforward decoding procedure, which makes signature generation, besides signature verification, very efficient from the computational standpoint. This is a favourable aspect with respect to alternative code-based schemes resorting to classical decoding algorithms like WAVE, which uses a modified Prange decoder exploiting the $(U, U+V)$ code structure resulting in a signature generation complexity of order $O(\lambda^3)$, where λ is the security level. Notice that, in the one-time case we consider, the running time of our algorithm is dominated by the key generation procedure, which is still of order $O(\lambda^3)$, while signature generation has complexity reduced to order $O(\lambda^2)$. However, even in the one-time case, key generation can be performed offline, so that its complexity is somewhat less important than the signature generation complexity. Moreover, in the multiple-time case using long-term keys, the signature generation complexity is definitely more important than the key generation complexity.

As anticipated, in this work we focus on one-time use of the new scheme and show that it is secure against known attacks. The study of multiple-time use, which will likely require the introduction of a final rejection sampling stage to avoid statistical attacks against long-term keys, is left as a future work.

2   Preliminaries

In this section we introduce the notation we will use throughout the paper and recall some basic notions concerning code-based cryptography.

2.1   Linear codes

Let $\mathbb{F}_q$ denote the Galois field of order q and let $\mathbb{F}_q^n$ denote the n-dimensional vector space defined on $\mathbb{F}_q$. A linear block code with length n and dimension k, denoted as $\mathcal{C}(n,k)$, is a linear subspace of dimension k of the vector space $\mathbb{F}_q^n$ formed by n-tuples defined over $\mathbb{F}_q$. An encoding for the code $\mathcal{C}$ is a map $\mathbb{F}_q^k \mapsto \mathbb{F}_q^n$ that creates a unique association between any k-tuple (or information word) u and an n-tuple (or codeword) c belonging to the code $\mathcal{C}$. Decoding instead starts from a version of c corrupted by an error vector e, $\hat{c} = c + e$, and aims at recovering c and e.

For any linear block code $\mathcal{C}(n,k)$, any set of k linearly independent codewords $\{g_0, g_1, \ldots g_{k-1}\}$ forms a basis of the codeword space, such that any codeword $c = [c_0, c_1, \ldots, c_{n-1}]$ can be expressed as a linear combination of the basis vectors:

$$c = u_0 g_0 + u_1 g_1 + \ldots + u_{k-1} g_{k-1}. \tag{1}$$

The coefficients u_i are taken from the information vector $u = [u_0, u_1, \ldots, u_{k-1}]$, such that encoding of u into c simply corresponds to the matrix operation $c = uG$, with

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} \tag{2}$$

being a $k \times n$ matrix known as *generator matrix* of the code $\mathcal{C}(n,k)$.

Any k linearly independent codewords can be selected to form G, therefore any code can be provided with as many encodings as the overall number of possible generator matrices. Encoding is said to be systematic if every codeword contains the information vector it is associated with.

A conventional form of systematic encoding is that in which each codeword is obtained by appending $r = n - k$ redundancy symbols to its corresponding k-symbol information word, that is

$$c = [u_0, u_1, \ldots, u_{k-1} | t_0, t_1, \ldots, t_{r-1}] \tag{3}$$

It follows that the associated generator matrix G can be written as

$$G = [I_k | P] \tag{4}$$

where I_k denotes the $k \times k$ identity matrix and P is a $k \times r$ general matrix.

Let us consider the orthogonal complement $\Gamma^\perp$ of the set of codewords Γ. Its dimension is

$$r = \dim\left(\Gamma^\perp\right) = n - \dim\left(\Gamma\right) = n - k. \tag{5}$$

Given r linearly independent n-symbol vectors $\{h_0, \ldots, h_{r-1}\}$ belonging to $\Gamma^\perp$, a basis of $\Gamma^\perp$ is simply obtained as

$$H = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{r-1} \end{bmatrix} \tag{6}$$

and $\Gamma = \text{Null} \{H\}$. The matrix H is known as a *parity-check matrix* of the code $\mathcal{C}(n,k)$ and every codeword $c \in \Gamma$ must verify

$$Hc^T = 0_{r \times 1} \tag{7}$$

where $0_{r \times 1}$ represents the $r \times 1$ all-zero vector. For any n-symbol vector $x \in \mathbb{F}_q^n$, the $r \times 1$ vector $s = Hx^T$ is denoted as the *syndrome* of x through H. It follows that a codeword belonging to $\mathcal{C}$ has an all-zero syndrome through H.

2.2   QC-LDGM codes

A linear block code is said to be low-density generator matrix (LDGM) if at least one of its generator matrices is sparse, i.e., has a fraction of non-zero entries $\ll 1/2$. SPANSE uses a secret LDGM code with length n and dimension k, characterized by a generator matrix that is sparse and has non-zero entries taking small values (only 1, as a special case). The rows of G have Hamming weight $w_g \ll n$. Due to their sparse nature, it is very likely that, by summing two or more rows of the generator matrix of an LDGM code, a vector with Hamming weight $> w_g$ is obtained. If the linear combination of any group of rows of G yields a codeword with weight greater than or equal to w_g, then the LDGM code has minimum distance w_g.

   The special class of LDGM codes used in SPANSE is that of QC-LDGM codes, having generator and parity-check matrices formed by circulant bocks with size $p \times p$, $p \in [2; r]$. As already known, using matrices in this form allows reducing the memory needed to store them, and hence the public key size. In fact, a QC code is defined as a linear block code with dimension $k = p \cdot k_0$ and length $n = p \cdot n_0$, in which each cyclic shift of a codeword by n_0 symbols results in another valid codeword [29]. From the definition of QC codes it follows that the characteristic matrices of these codes can be written as formed by circulant blocks of the type:

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{p-1} \\ a_{p-1} & a_0 & a_1 & \cdots & a_{p-2} \\ a_{p-2} & a_{p-1} & a_0 & \cdots & a_{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & a_3 & \cdots & a_0 \end{bmatrix}. \tag{8}$$

   It is then evident that any circulant matrix can be fully described by just one of its rows or columns. The set of $p \times p$ circulant matrices with entries from $\mathbb{F}_q$ forms a ring under the standard operations of matrix addition and multiplication over $\mathbb{F}_q$. The zero element is the all-zero matrix, and the identity element is the $p \times p$ identity matrix. If we consider the algebra of polynomials mod $(x^p - 1)$ over $\mathbb{F}_q$, $\mathbb{F}_q[x]/\langle x^p + 1 \rangle$, the following map is an isomorphism between this algebra and that of $p \times p$ circulant matrices over $\mathbb{F}_q$:

$$A \leftrightarrow a(x) = \sum_{i=0}^{p-1} a_i \cdot x^i. \tag{9}$$

According to (9), any circulant matrix is associated to a polynomial in the variable x having coefficients over $\mathbb{F}_q$ which coincide with the entries in the first row of the matrix:

$$a\left(x\right) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{p-1} x^{p-1}. \tag{10}$$

Similarly, the all-zero circulant matrix corresponds to the null polynomial and the identity matrix to the unitary polynomial.

As it will be described next, the main part of the secret key of SPANSE is formed by a QC-LDGM code described through its $k \times n$ generator matrix G and $r \times n$ parity-check matrix H. The values of n, k and r are all multiples of the circulant block size p.

For a QC-LDGM code, the generator matrix G takes the following general form

$$G = \begin{bmatrix} G_{0,0} & G_{0,1} & G_{0,2} & \dots & G_{0,n_0-1} \\ G_{1,0} & G_{1,1} & G_{1,2} & \dots & G_{1,n_0-1} \\ G_{2,0} & G_{2,1} & G_{2,2} & \dots & G_{2,n_0-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{k_0-1,0} & G_{k_0-1,1} & G_{k_0-1,2} & \dots & G_{k_0-1,n_0-1} \end{bmatrix}, \tag{11}$$

where each $G_{i,j}$ is a sparse circulant matrix or a null matrix with size $p \times p$. Hence, in this case the code length, dimension and redundancy are $n = n_0 p$, $k = k_0 p$ and $r = (n_0 - k_0)p = r_0 p$, respectively. Since a circulant matrix is defined by one of its rows (conventionally the first), storing a matrix in the form (11) requires $k_0 n_0 p$ symbols, yielding a reduction by a factor p with respect to a matrix with a general form.

An important feature of the LDGM codes used in SPANSE is that it is easy to obtain a random codeword c belonging to the code and having weight approximately equal to a fixed, small value w_c. Let us suppose, for simplicity, that w_c is an integer multiple of w_g. Since the rows of G are sparse, it is very likely that, by summing a small number of rows, the Hamming weight of the resulting vector is about the sum of their Hamming weights. Hence, by summing $\frac{w_c}{w_g}$ rows of G, chosen at random, we get a random codeword with Hamming weight about w_c. It follows that the number of random codewords with weight close to w_c can be roughly estimated as

$$A_{w_c} \approx \binom{k}{\frac{w_c}{w_g}}. \tag{12}$$

3   Dense code-based signatures

SPANSE inherits the setting for the generation of sparse code-based signatures introduced in [7], with the main difference that the structure of the matrix S is changed to obtain dense signatures intead of sparse ones.

As in [7], two public functions are fixed beforehand: a hash function $\mathcal{H}$ and a function $\mathcal{F}_\Theta$ that converts the output vector of $\mathcal{H}$ into a sparse binary vector s with length r and weight w ($\leq r$). Note that, by sparse, we here refer to a vector taking values in $\{0; 1\} \subseteq \mathbb{F}_q$. The vector s is a public binary syndrome vector resulting from the signature generation procedure. The output of $\mathcal{F}_\Theta$ is uniformly distributed and depends on a parameter Θ, which is chosen for each message to be signed and is made public by the signer. The choice of $\mathcal{F}_\Theta$ is discussed in Section 3.2.

3.1 Key generation

A first component of the secret key of SPANSE is a secret QC-LDGM code having a generator matrix of the type (11). The overall row weight of the generator matrix G of the secret code used in SPANSE is equal to w_g. A SPANSE key pair is generated according to the following steps.

1. The signer randomly chooses a secret QC-LDGM code with length $n = n_0 p$ and dimension $k = k_0 p$, having a generator matrix G in the form (11), with row weight w_g, and from G computes a systematic parity-check matrix H for the same code. Note that G is binary as well, i.e., takes values in $\{0; 1\} \subseteq \mathbb{F}_q$.
2. Differently from [7], where the two secret matrices Q and S were used, in SPANSE the signer chooses two different secret matrices, P and S, still in QC form: an $r \times r$ secret permutation matrix P and an $n \times n$ matrix S with entries $\in \mathbb{F}_q$. Having S defined over $\mathbb{F}_q$ is another important difference of SPANSE with respect to [7]. In fact, in [7] the matrix S was sparse (besides QC). In SPANSE, instead, S is still in QC form but it is dense, with a different constraint of having small-valued entries taken from $\mathbb{F}_q$. Let us denote as d_i the fraction of entries equal to $i, i = 0, 1, 2, \ldots, q-1$, in each row of S. Then, we define

$$d(x) = \sum_{i=0}^{q-1} d_i x^i, \tag{13}$$

with $\sum_{i=0}^{q-1} d_i = 1$ as the polynomial describing the density of the symbols of $\mathbb{F}_q$ in each row of S. As we will see next, the polynomials used in SPANSE as $d(x)$ are characterized by large values of d_i for small values of i, and small or zero values of d_i for large values of i. For brevity, we say that such polynomials are *zero-concentrated*. Note that this does not mean that S is sparse. In fact, as we will see next, the overall density of non-zero entries in S can easily be in the order of 50%. The matrix Q used in [7], instead, is no longer needed in SPANSE because of the dense nature of S, and is replaced with a simple permutation matrix P.
3. The private key is $\{P, G, S\}$ and the public key is obtained as $H' = P^{-1} \cdot H \cdot S^{-1}$.

3.2 Signature generation

In SPANSE, the signature of a message (m) is computed as follows.

1. The signer computes $h = \mathcal{H}(m)$ and $s = \mathcal{F}_\Theta(h)$. The parameter Θ is chosen at random if statistical signatures are desired, otherwise it is fixed or computed as a one-way function of the message m if deterministic signatures are desired. The chosen value Θ^* is published along with the signature.
2. The signer computes the permuted syndrome $s' = Ps$. Then, being provided with the systematic parity-check matrix H corresponding to the secret generator matrix G, the signer computes a sparse error vector e having syndrome s' through H. Due to the systematic form of H, e is easily obtained as $e = \left[0_{1 \times k} | s'^T\right]$, where $0_{1 \times k}$ is an all-zero row vector of length k.
3. The signer selects a random weight-w_c codeword c belonging to the secret QC-LDGM code and computes the signature of m as $\sigma = (e + c) \cdot S^T$. If σ has one or more zero entries, the signer chooses another random codeword c and tries again, until σ is free of zero entries.
4. The message m, the associated parameter Θ^* and the signature σ are then published.

An important parameter for any digital signature scheme is the total number of different signatures. In SPANSE, a different signature corresponds to a different r-bit vector s, having weight w and values in $\{0; 1\}$, so the total number of different signatures is

$$N_s = \binom{r}{w}. \tag{14}$$

3.3   Signature verification

Signature verification in SPANSE is performed as follows.

1. The verifier checks that σ is free of zero entries. If such a check fails, the signature is discarded.
2. Otherwise, the verifier computes $s^* = \mathcal{F}_{\Theta^*}(\mathcal{H}(m))$ and checks that s^* has weight w. If such a check fails, the signature is discarded.
3. Otherwise, the verifier computes $H' \cdot \sigma^T = P^{-1} \cdot H \cdot S^{-1} \cdot S \cdot (e^T + c^T) = P^{-1} \cdot H \cdot (e^T + c^T) = P^{-1} \cdot H \cdot e^T = P^{-1} \cdot s' = s$ and compares it with s^*.
4. If $s = s^*$ the signature is accepted; otherwise, it is discarded.

4   Security analysis

As anticipated, in this work we study the security of SPANSE under the simplifying assumption of one-time use, which is a necessary condition for a possible multiple-time use. Since secure instances of SPANSE in the one-time case can easily be identified, as it will be shown next, the possible extension to multiple-time use will be the object of future works.

As a preliminary observation, we note that SPANSE lays its foundations on the scheme introduced in [7], with the crucial difference that attacks exploiting the sparsity of signatures are intrinsically avoided. Among these, support decomposition attacks described in [7] are the most dangerous ones, and are

actually avoided in SPANSE by resorting to dense signatures. The other attacks described in [7] are still feasible, at least in principle, but their complexity remains exponential in the key size and can be easily rendered very large, as we will show next.

4.1   Attacks based on decoding large-weight vectors

Opposed to the system proposed in [7], which is based on the hardness of the classical decoding problem for low-weight vectors, the security of SPANSE relies on the difficulty of finding large-weight vectors through decoding of general codes. In fact, general decoding algorithms could be used to:

- mount key recovery attacks based on retrieving the rows of the dense generator matrix of the public code $G' = GS^T$, from which an attacker could try to recover the two secret matrices G and S, or
- mount forgery attacks by searching for a dense vector σ_f corresponding to a given syndrome s through the public parity-check matrix H'.

In both these cases, an attacker should be able to find a vector free of zero entries[1] that corresponds to a given syndrome (the all-zero vector for a key recovery attack and s for a forgery attack) through the public parity-check matrix H'. By using brute force, the probability of success for the attacker is easily computed as the fraction of valid vectors for each syndrome, that is,

$$p_{BF} = \frac{(q-1)^n}{q^n q^r}. \tag{15}$$

For example, by considering $n = 15000$ and $q = 127$, we have that $\left(\frac{q-1}{q}\right)^n < 2^{-170}$, which clearly highlights the unfeasibility of signature forgery attacks through brute force, even if the term q^r is neglected.

So, as in WAVE, the security of our scheme relies on the hardness of decoding a given syndrome into a large-weight vector. Differently from WAVE, however, we require signatures to have maximum weight. A similar condition can also be imposed on the rows of the public code generator matrix $G' = GS^T$, as can be easily shown. Obtaining vectors of full density is made easy in SPANSE by combining together sparsity and density in vector-matrix products. Let us focus on signatures, but a similar reasoning also applies to the computation of $G' = GS^T$. Generating signatures free of zero entries is made easy in SPANSE by the sparse nature of the vector $e + c$ combined with the dense and zero-concentrated nature of S. To show this, let us consider $\mathbb{F}_q$ with q being a prime and let us drop the modulo operation, considering for the moment that the operations are performed on the full set of integers. The dense nature of S makes the probability that any entry of the signature computed as $(e + c)S^T$ is exactly equal to zero negligible. Furthermore, the sparse nature of $e+c$ together with the

---

[1] Note that, more generally, one may require vectors to have weight larger than some threshold value (which should be $\geq \frac{q-1}{q}n$).

zero-concentrated nature of S allow making the probability that any signature entry overcomes $q-1$ arbitrary small, or even null. Based on these considerations, when we restore the modulo operation we obtain that the probability of having a zero entry over $\mathbb{F}_q$ is negligible or even null. This motivates the fact that generating signatures free of zero entries is easy for the signer. Nevertheless, generating valid signatures may require some rejection sampling, depending on the choice of $d(x)$. However, we show that when proper parameters are chosen, the number of needed attempts is very small.

The PGE+SS framework In [15] the authors study the so-called PGE+SS framework to solve the large-weight vector decoding problem over the ternary finite field (i.e., for $q = 3$). Their algorithm works by first applying Partial Gaussian Elimination (PGE) to reduce to a smaller instance, which is then solved through techniques inherited from the Subset Sum (SS) framework. We extend the approach in [15] to the setting of interest for SPANSE, that is, considering vectors with entries over $\mathbb{F}_q$ with $q > 3$ and full (or very large) density. Starting from s and H, a PGE can first be used to obtain

$$H' = \begin{bmatrix} A \in \mathbb{F}_q^{(r-u)\times(n-u)} & 0_{(r-u)\times u} \in \mathbb{F}_q^{(r-u)\times u} \\ B \in \mathbb{F}_q^{u\times(n-u)} & I_u \in \mathbb{F}_q^{u\times u} \end{bmatrix}, \tag{16}$$
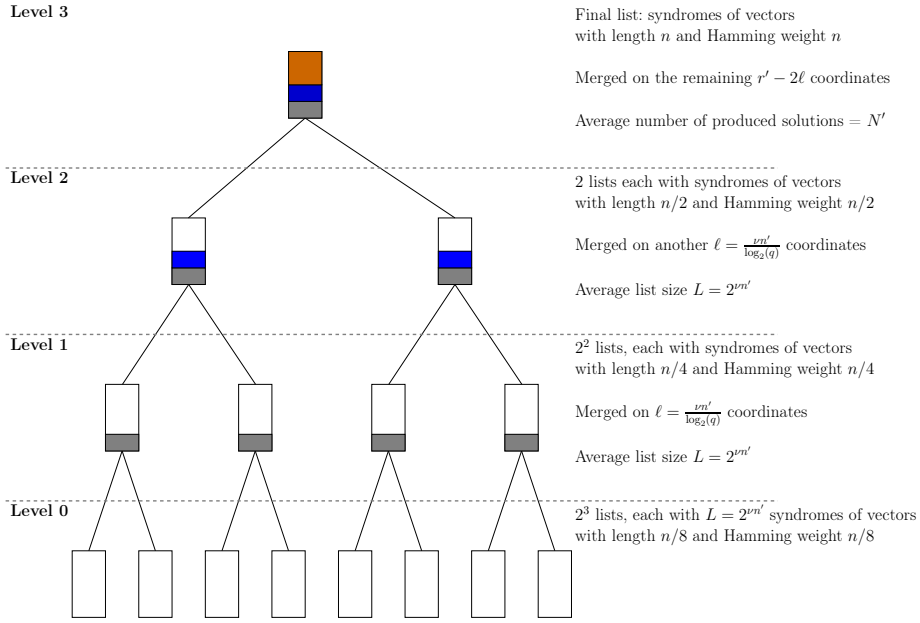
where $u = \varphi n$ with $0 < \varphi < r = (1-R)n$. We apply the same transformation to the syndrome s, obtaining $s' = [s_A, s_B]$, where s_A has length $r - u$ and s_B has length u. Let $e = [e_A, e_B]$ such that $He^T = s$. Then, we obtain

$$\begin{cases} Ae_A^T = s_A, \\ e_B^T = s_B - Be_A^T. \end{cases} \tag{17}$$

To solve the above system, one first finds e_A and then uses it to compute the corresponding e_B. Since both e_A and e_B must have full weight, we have that A and s_A represent the input parity-check matrix and syndrome for another large weight decoding instance, with length $n' = n - u = (1 - \varphi)n$ and redundancy $r' = r - u = (1 - R - \varphi)n$. We refer to it as the *small instance* and use a SS solver to find solutions, i.e., candidates for e_A.

As in [15], we consider here the application of a tree merging structure, in which we start from 2^b lists and then pairwise merge them until we are left with a final list of solutions. Each list if populated with L vectors of length $n'2^{-b}$ and full Hamming weight $n'2^{-b}$. Due to lack of space, we do not report here the full details of the algorithm, but it can be easily seen that it is a simple generalization of the technique described in [15]. Its operating principle is exemplified in Figure 1 for the case of $b = 3$.

We consider $L = 2^{\nu n'}$ and, as in [15], focus on the use of the so-called *amortized lists*: in each merge we desire output lists having the same (average) size of the input ones. In other words, given that we start from lists of size L, we want to always have lists of the same average size. This is achieved if, in the first $b - 1$ levels, one always merges on $\ell = \frac{\nu n'}{\log_2(q)}$ positions.

Level 3

Final list: syndromes of vectors
with length n and Hamming weight n

Merged on the remaining $r' - 2\ell$ coordinates

Average number of produced solutions $= N'$

Level 2

2 lists each with syndromes of vectors
with length $n/2$ and Hamming weight $n/2$

Merged on another $\ell = \frac{\nu n'}{\log_2(q)}$ coordinates

Average list size $L = 2^{\nu n'}$

Level 1

2^2 lists, each with syndromes of vectors
with length $n/4$ and Hamming weight $n/4$

Merged on $\ell = \frac{\nu n'}{\log_2(q)}$ coordinates

Average list size $L = 2^{\nu n'}$

Level 0

2^3 lists, each with $L = 2^{\nu n'}$ syndromes of vectors
with length $n/8$ and Hamming weight $n/8$

Fig. 1. Example of the algorithm to solve the small instance, for the case of $b = 3$.

The list which is produced as the output of the last level contains the solutions to the problem; note that its average size can be estimated as

$$
\begin{aligned}
N' &= 2^{2\nu n' - \left(r' - (b-1)\ell\right)\log_2(q)} \\
&= 2^{\nu(b+1)n' - r'\log_2(q)} \\
&= 2^{\left((b+1)\nu - (1-R')\log_2(q)\right)n'} \\
&= 2^{\left((b+1)\nu - (1-R')\log_2(q)\right)(1-\varphi)n} \\
&= 2^{\rho(b,\nu,\varphi)n}, \tag{18}
\end{aligned}
$$

where $R' = 1 - \frac{r'}{n'} = \frac{R}{1-\varphi}$. When constructing the initial lists, we also have the constraint that it must be $2^{\nu n'} \leq (q-1)^{n' 2^{-b}}$, from which we obtain

$$
\nu \leq 2^{-b}\log_2(q-1). \tag{19}
$$

Also, consider that each of the initial lists must contain at least one element, so that we also obtain the following constraint

$$
\begin{aligned}
b &\leq \lfloor \log_2(n') \rfloor \\
&= \lfloor \log_2(n) + \log_2(1-\varphi) \rfloor. \tag{20}
\end{aligned}
$$

Once the small instance has been solved, we use each found candidate for e_A to construct e_B. The probability that a candidate e_B is valid, i.e., that it has full weight u, can be estimated as $\left(1 - \frac{1}{q}\right)^u = 2^{\log_2(1-1/q)\varphi n}$, so that the probability that one iteration of the algorithm is successful is

$$
\begin{aligned}
\Pr[\text{success}] &= 1 - \left(1 - 2^{\log_2(1-1/q)\varphi n}\right)^{N'} \\
&\approx \min\left\{1 \; ; \; N'2^{\log_2(1-1/q)\varphi n}\right\} \\
&= \min\left\{1 \; ; \; 2^{\left(\left((b+1)\nu - (1-R')\log_2(q)\right)(1-\varphi) + \varphi \log_2(1-1/q)\right)n}\right\} \\
&= \min\left\{1 \; ; \; 2^{\chi(b,\varphi)n}\right\} \\
&= 2^{n\cdot\min\{0 \; ; \; \chi(b,\varphi)\}}.
\end{aligned}
\tag{21}
$$

We neglect the cost of performing PGE (since it is polynomial in n), and consider that lists initialization and merging comes with a cost of $L = O\left(2^{\nu n'}\right) = O\left(2^{(1-\varphi)n}\right)$. Testing each one of the produced N' solutions for the small instance requires $O(N')$ operations. So, each iteration of the algorithm requires $O\left(2^{\nu(1-\varphi)n} + 2^{\rho(b,\nu,\varphi)n}\right) = O\left(2^{n\cdot\max\{\nu(1-\varphi) \; ; \; \rho(b,\nu,\varphi)\}}\right)$ operations. Multiplying this quantity by the average number of performed iterations, that is, the reciprocal of the success probability, we obtain the following cost estimate

$$
T_{SDP} = \min_{\substack{\varphi\in\mathbb{R}, \; 0<\varphi<1-R \\ b\in\mathbb{N}, \; b\leq\lfloor\log_2((1-\varphi)n)\rfloor \\ \nu\in\mathbb{R}, \; 0<\nu<2^{-b}\log_2(q-1)}} \left\{\frac{2^{\nu(1-\varphi)n} + 2^{\rho(b,\nu,\varphi)n}}{\Pr[\text{success}]}\right\}.
\tag{22}
$$

Taking into account all the above considerations, it is easily seen that the time complexity of the resulting algorithm can be compactly expressed as $2^{\alpha(\varphi,b,\nu)n}$, where the complexity exponent is computed as

$$
\min_{\substack{\varphi\in\mathbb{R}, \; 0<\varphi<1-R \\ b\in\mathbb{N}, \; b\leq\lfloor\log_2((1-\varphi)n)\rfloor \\ \nu\in\mathbb{R}, \; 0<\nu<2^{-b}\log_2(q-1)}} \left\{\max\left\{\nu(1-\varphi) \; ; \; \rho(b,\nu,\varphi)\right\} - \min\left\{0 \; ; \; \chi(b,\nu,\varphi)\right\}\right\}.
$$

$$
\tag{23}
$$

4.2   Decoding One Out of Many

In the Decoding One Out of Many (DOOM) setting, one has a set of syndromes $\mathcal{S} = \left\{s^{(0)}, s^{(1)}, \cdots, s^{(N-1)}\right\}$ and wants to find a vector e with some weight requirement and so that $H'e^T \in \mathcal{S}$. For the low weight SDP, it is well known that general decoding algorithms receive a speed-up of $\sqrt{N}$. At the best of our knowledge, the DOOM setting has not been studied for the problem of decoding

vectors with large weight. However, it is likely that DOOM is easier that the standard case in which there is only one target syndrome. In the scheme in [7], an attacker can construct a set $\mathcal{S}$ with the following strategy:

- prepare a set of N' syndromes, computed as $\widetilde{s}^{(\Theta)} = \mathcal{F}(m||\Theta)$, where Θ is a counter and takes values in $\{0, 1, \cdots, N'-1\}$;
- populate $\mathcal{S}$ with each $\widetilde{s}^{(\Theta)}$ and all of its $p-1$ QC shifts.

By doing this, the attacker end up with a set $\mathcal{S}$ of size $N = pN'$: clearly, solving DOOM in this case yields a valid syndrome for the message m.

In SPANSE, differently from [7], the matrix Q is replaced by a permutation matrix and Θ is no longer used for rejection sampling, thus it can be fixed as a one-way function of the message to obtain deterministic signatures. Consequently, at best, an attacker can build $\mathcal{S}$ by using only the QC shifts of $s = \mathcal{F}(m)$. This implies that $N = p$. To consider the possible speed-ups coming from DOOM, we assume that the gain is the same that one has in the low-weight regime. Hence, we assess the cost of a signature forgery attack as

$$T_{DOOM} = \frac{T_{SDP}}{\sqrt{p}} = O\left(2^{\alpha(\varphi,b,\nu)n - 0.5\log_2(p)}\right). \tag{24}$$

5   System design

In this section we study the system design and the choice of the relevant parameters. We also provide a preliminary set of parameters for an instance of SPANSE able to achieve 128 bits of classical security. We remark that such a set of parameters does not result from an optimization procedure, and that many degrees of freedom exist for tuning the system parameters, which can be exploited to design other instances of SPANSE possibly achieving even better tradeoffs and higher levels of security. This, however, is left for future works.

There are two main factors that affect the design of parameters for SPANSE:

- The security level, which is based on the difficulty of decoding large-weight vectors, as shown in the previous section.
- The average number of attempts needed for generating a valid signature, which depends on the desired density of the signatures (related to the security level) and the distribution $d(x)$ of the entries of S.

For the latter, in the next section we consider the simplified case in which the entries of S take values in $\{0; 1\}$ with equal probability, that is $d(x) = 0.5 + 0.5x$. This choice allows an easy modeling of the rejection sampling rate, as shown in the next section. However, more general choices of $d(x)$ are possible, which may provide stronger security guarantees paid in terms of some slower rejection sampling. Some examples will be provided in Section 5.3, along with the corresponding rejection rates estimated through Monte Carlo simulations. A more general theoretical analysis of the rejection rate is left for future works.

5.1   Rejection sampling

As described in Section 3.2, the signature generation of SPANSE may require some rejection sampling in order to guarantee that each produced signature has maximum weight n. To this end, we need to estimate the probability that a random signature is valid: the reciprocal of this probability corresponds to the average number of attempts needed to generate a valid signature.

To do this, we consider that

$$\sigma = (c + e)S^T = cS^T + eS^T = \widetilde{c} + \widetilde{e},$$

where $c = uG$ with u having weight m_g, G is formed by rows with weight w_g, e has weight w and S has columns and rows with weight m_S. In particular, e is partitioned as

$$e = [\ \underbrace{e'}_{\text{Length } k}\ ||\ \underbrace{e''}_{\text{Length } n - k}\ ],$$

where e' is an all-zero vector and e'' has weight w. Note that u and e take values in $\{0; 1\}$, while σ generically takes values in $\mathbb{F}_q$. From now on, we consider the simplified case in which also the entries of G and S take values in $\{0; 1\}$. Note that this is not a constraint in SPANSE, and is considered in this preliminary work only for easiness of analysis.

We start by deriving the weight distribution of $\widetilde{c}$. To this end, we consider that also c has a varying weight: indeed, each of its entries is obtained as the inner product between u and the corresponding column of G. We approximate each column of G as a vector formed by k samples of a Bernoulli variable with parameter $\frac{w_g}{n}$. Since $m_g < q$, any entry of c will be equal to x with probability

$$\Pr[c_i = x] = \binom{m_g}{x} \left(\frac{w_g}{n}\right)^x \left(1 - \frac{w_g}{n}\right)^{m_g - x}$$

$$= f_{\frac{w_g}{n}}(m_g, x). \tag{25}$$

When $m_g \ll k$ and $w_g \ll n$, one can approximate the above probability by simply considering that either $c_i = 0$ or $c_i = 1$. Consequently, we approximate c as a vector of n samples of a Bernoulli variable with parameter $\rho_c = 1 - \left(1 - \frac{w_g}{n}\right)^{m_g} \approx \frac{m_g w_g}{n}$.

We now consider the product $\widetilde{c} = cS^T$: each column of S^T has constant weight m_S, so that each entry of $\widetilde{c}$ is in the form $\sum_{i=0}^{n-1} c_i s_i$. Again, we approximate s_i as a Bernoulli variable with parameter $\rho_S = \frac{m_S}{n}$. Consequently, we have

$$\Pr[\widetilde{c}_i = x] = \sum_{z=x}^{m_g w_g} f_{\rho_c}(n, z) \sum_{\substack{x' = x, x+q, \cdots \\ x' \leq z}} f_{\rho_S}(z, x'). \tag{26}$$

We repeat the same reasoning for $\widetilde{e} = eS^T$, considering that e has constant weight w only in the last $n - k$ entries. Since $w < q$, we obtain

$$\Pr\left[\widetilde{e}_i = x\right] = \binom{w}{x'} \left(\frac{m_S}{n}\right)^{x'} \left(\frac{1 - m_S}{n}\right)^{z - x'}$$
$$= f_{\rho_S}(w, x'). \tag{27}$$

We are finally ready to obtain the probability that an entry in σ is null, that is

$$\Pr[\sigma_i = 0] = \sum_{x=0}^{q-1} \Pr[\widetilde{c} = x]\Pr[\widetilde{e} = -x]. \tag{28}$$

So, the probability to obtain a valid signature is

$$\Pr[\sigma \text{ is valid}] = \left(1 - \Pr[\sigma_i = 0]\right)^n. \tag{29}$$

The reciprocal of the above probability is the average number of attempts, before a valid signature is generated.

5.2   Computational complexity

As we have described in the previous section, the bottleneck in the attacks consists in forgery attacks, whose running time is an exponential function of the code length n. Actually, because of the DOOM setting, the time complexity receives a small speed-up owing to the quasi-cyclicity of the code. Yet, the speed-up grows with the square root of p. Also, note that the code length n is linear in p. In practice, we choose p as a very small constant (say, a few hundreds), so that the cost of forgery attacks is essentially linear in n. Then, in practice, to achieve a security level of λ bits we need $n = O(\lambda)$.

Key generation requires to compute inversions over $\mathbb{F}_q$ (since we need to compute the systematic parity-check matrix and S^{-1}), which can be performed with cost $O(n^3) = O(\lambda^3)$. The generation of a signature, instead, requires only[2] vector sums and vector-matrix multiplications, so takes time $O(n^2) = O(\lambda^2)$. Notice that, in principle, one needs to consider also the average number of rejected signatures before a valid one is obtained. Yet, when q is large enough, the rejection rate is practically constant (and extremely close to 0). This implies that the final rejection sampling has no practical impact on the cost of the signature generation algorithm, which we consequently estimate as $O(\lambda^2)$. Finally, signature verification clearly comes with the same cost $O(\lambda^2)$ since, again, only hashes and multiplications are involved.

5.3   System parameters

Based on the above considerations, let us consider a possible instance of SPANSE with the following parameters.

---

[2] Note that computing hash functions takes time which does not depend on n

- Code length: $n = 24000$
- Code dimension: $k = 12000$
- Circulant block size: $p = 101$
- Field order: $q = 127$
- Syndrome weight: $w = 26$
- Weight of the rows of G: $w_g = 11$
- Number of rows of G selected per generated codeword: $m_g = 12$

For the choice of the distribution of the entries of $\mathbb{F}_q$ in each row of S, we can consider several solutions. A first simple case is that of considering a uniform binary matrix, that is, $d(x) = 0.5 + 0.5x$. Through the analysis reported in Section 5.1 we can verify that this guarantees a negligible rejection rate for the system parameters we are considering. In fact, for these parameters, the probability that a produced signature is not valid (i.e., does not have full weight) is estimated as $1.44 \cdot 10^{-6}$, confirming that the rejection rate can, in practice, be neglected. For instance, the probability that signature generation has to be repeated for more than two times is lower than 2^{-38}.

However, we may also consider other distributions for the entries of S including more elements of $\mathbb{F}_q$. For example, by considering $d(x) = 0.5783 + 0.4167x + 0.0042x^2 + 0.00083x^{13}$ we have verified through Monte Carlo simulations that the signature rejection rate is in the order of 1%, which is still largely acceptable. Including larger values from $\mathbb{F}_q$ in the entries of S inevitably increases the signature rejection rate. As another example, we have considered $d(x) = 0.5775 + 0.4167x + 0.0042x^2 + 0.00083x^{13} + 0.00083x^{25}$ and verified through Monte Carlo simulations that the signature rejection rate in this case is in the order of 98.5%. Considering that the generation of each signature in SPANSE is very fast, however, such a rejection rate may still be acceptable and result in an overall fast signature generation.

From the above choice of the system parameters, it follows that:

- The public key size is $K_s = 2436.6$ kB.
- The number of different signatures is $N_s = 2^{263.9}$.
- The number of different codewords is $N_c = 2^{133.8}$.
- The work factor of attacks based on information set decoding (ISD) is 2^{132}.

Concerning the PGE+SS solver, we found that the attack is optimized with the following parameters:

- $b = 9$;
- $\nu = 0.010725$;
- $\varphi = 0.493000$.

The resulting cost, taking also DOOM into account, is $2^{131.6}$.

6   Conclusion

We have introduced SPANSE, a code-based digital signature scheme exploiting the hash-and-sign paradigm and the difficulty of performing decoding of large-weight vectors for random-like codes. The new scheme stems from the previous

proposal in [7], but prevents the applicability of attacks exploiting sparse matrices by replacing them with dense ones. This is paid in terms of public key size, which is increased by the use of non-binary codes defined over $\mathbb{F}_q$ instead of the binary ones that were used in [7]. However, the resulting public keys appear to be in the order of 2.5 MB for 128 bits of classical security, which seems a reasonable size. Moreover, the new system exploits a straightforward signature generation procedure, only based on vector-matrix operations, which makes it intrinsically faster than alternative systems requiring decoding for signature generation.

In this first work we only studied the case of one-time keys, showing that known attacks have exponential complexity and allow devising secure instances under such a setting. The multiple-time case needs the introduction of an additional rejection sampling stage to tune the distribution of produced signatures, and will be studied in future works.

References

1. Aguilar, C., Gaborit, P., Schrek, J.: A new zero-knowledge code based identification scheme with reduced communication. In: 2011 IEEE Information Theory Workshop (ITW). pp. 648–652. Paraty, Brazil (Oct 2011)
2. Aguilar-Melchor, C., Blazy, O., Deneuville, J.C., Gaborit, P., Zémor, G.: Efficient encryption from random quasi-cyclic codes. IEEE Transactions on Information Theory **64**(5), 3927–3943 (2018). https://doi.org/10.1109/TIT.2018.2804444
3. Apon, D., Perlner, R., Robinson, A., Santini, P.: Cryptanalysis of LEDAcrypt. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 389–418. Springer International Publishing, Cham (2020)
4. Baldi, M., Barenghi, A., Chiaraluce, F., Pelosi, G., Santini, P.: LEDAkem: A post-quantum key encapsulation mechanism based on qc-ldpc codes. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography. pp. 3–24. Springer International Publishing, Cham (2018)
5. Baldi, M., Barenghi, A., Chiaraluce, F., Pelosi, G., Santini, P.: LEDAcrypt: QC-LDPC code-based cryptosystems with bounded decryption failure rate. In: Baldi, M., Persichetti, E., Santini, P. (eds.) Code-Based Cryptography. pp. 11–43. Springer International Publishing, Cham (2019)
6. Baldi, M., Battaglioni, M., Chiaraluce, F., Horlemann-Trautmann, A.L., Persichetti, E., Santini, P., Weger, V.: A new path to code-based signatures via identification schemes with restricted errors (2021), https://arxiv.org/abs/2008.06403
7. Baldi, M., Bianchi, M., Chiaraluce, F., Rosenthal, J., Schipani, D.: Using LDGM codes and sparse syndromes to achieve digital signatures. In: Gaborit, P. (ed.) Post-Quantum Cryptography, Lecture Notes in Computer Science, vol. 7932, pp. 1–15. Springer Berlin Heidelberg (2013)
8. Barenghi, A., Biasse, J.F., Persichetti, E., Santini, P.: LESS-FM: fine-tuning signatures from the code equivalence problem. In: International Conference on Post-Quantum Cryptography. pp. 23–43. Springer (2021)
9. Barg, S.: Some new NP-complete coding problems. Problemy Peredachi Informatsii **30**(3), 23–28 (1994)
10. Bellini, E., Caullery, F., Gaborit, P., Manzano, M., Mateu, V.: Improved Veron identification and signature schemes in the rank metric. In: 2019 IEEE International Symposium on Information Theory (ISIT). pp. 1872–1876. Paris, France (2019)

11. Berlekamp, E., McEliece, R., van Tilborg, H.: On the inherent intractability of certain coding problems. IEEE Trans. Inform. Theory **24**(3), 384–386 (May 1978)
12. Bettaieb, S., Bidoux, L., Blazy, O., Gaborit, P.: Zero-knowledge reparation of the véron and ags code-based identification schemes. In: 2021 IEEE International Symposium on Information Theory (ISIT). pp. 55–60. IEEE (2021)
13. Cayrel, P.L., Véron, P., El Yousfi Alaoui, S.M.: A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) Selected Areas in Cryptography. pp. 171–186. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
14. Courtois, N., Finiasz, M., Sendrier, N.: How to achieve a McEliece-based digital signature scheme. Cryptology ePrint Archive, Report 2001/010 (2001), `http://eprint.iacr.org/`
15. Debris-Alazard, T., Sendrier, N., Tillich, J.P.: Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 21–51. Springer International Publishing, Cham (2019)
16. El Yousfi Alaoui, S.M., Cayrel, P.L., El Bansarkhani, R., Hoffmann, G.: Code-based identification and signature schemes in software. In: Cuzzocrea, A., Kittl, C., Simos, D.E., Weippl, E., Xu, L. (eds.) Security Engineering and Intelligence Informatics. pp. 122–136. Springer Berlin Heidelberg (2013)
17. Faugère, J.C., Otmani, A., Perret, L., Tillich, J.P.: A distinguisher for high rate McEliece cryptosystems. In: Proc. IEEE Information Theory Workshop (ITW). pp. 282–286. Paraty, Brazil (Oct 2011)
18. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology — CRYPTO' 86. pp. 186–194. Springer Berlin Heidelberg, Berlin, Heidelberg (1987)
19. Gueron, S., Persichetti, E., Santini, P.: Designing a practical code-based signature scheme from zero-knowledge proofs with trusted setup. Cryptography **6**(1), 5 (2022)
20. Kabatianskii, G., Krouk, E., Smeets, B.: A digital signature scheme based on random error correcting codes. In: Proc. 6th IMA Int. Conf. on Cryptography and Coding. pp. 161–167. London, UK (1997)
21. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012. pp. 738–755. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
22. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. DSN Progress Report pp. 114–116 (1978)
23. Misoczki, R., Tillich, J.P., Sendrier, N., Barreto, P.S.L.M.: MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In: 2013 IEEE International Symposium on Information Theory. pp. 2069–2073 (2013). `https://doi.org/10.1109/ISIT.2013.6620590`
24. National Institute of Standards and Technology: Post-quantum crypto project (Dec 2016), `http://csrc.nist.gov/groups/ST/post-quantum-crypto/`
25. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Probl. Contr. and Inform. Theory **15**, 159–166 (1986)
26. Otmani, A., Tillich, J.P.: An efficient attack on all concrete KKS proposals. In: Yang, B.Y. (ed.) Post-Quantum Cryptography. pp. 98–116. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
27. Phesso, A., Tillich, J.P.: An Efficient Attack on a Code-Based Signature Scheme, pp. 86–103. Springer International Publishing, Cham (2016)

28. Stern, J.: A new identification scheme based on syndrome decoding. In: Stinson, D.R. (ed.) Advances in Cryptology — CRYPTO' 93. pp. 13–21. Springer Berlin Heidelberg, Berlin, Heidelberg (1994)
29. Townsend, R., Weldon, E.J.: Self-orthogonal quasi-cyclic codes. IEEE Trans. Inform. Theory **13**(2), 183–195 (Apr 1967)
30. Véron, P.: Improved identification schemes based on error-correcting codes. Applicable Algebra in Engineering, Communication and Computing **8**(1), 57–69 (1997)