UNIVERSITÀ POLITECNICA DELLE MARCHE
Repository ISTITUZIONALE

Cryptanalysis of a code-based full-time signature

This is the peer reviewd version of the followng article:

Original
Cryptanalysis of a code-based full-time signature / Aragon, N., Baldi, M., Deneuville, J., Khathuria, K., Persichetti, E., Santini, P.. - In: DESIGNS, CODES AND CRYPTOGRAPHY. - ISSN 0925-1022. - ELETTRONICO. - 89:9(2021), pp. 2097-2112. [10.1007/s10623-021-00902-7]

Availability:
This version is available at: 11566/291093 since: 2024-07-02T08:26:32Z

Publisher:

Published
DOI:10.1007/s10623-021-00902-7

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. The use of copyrighted works requires the consent of the rights' holder (author or publisher). Works made available under a Creative Commons license or a Publisher's custom-made license can be used according to the terms and conditions contained therein. See editor's website for further information and terms and conditions.
This item was downloaded from IRIS Università Politecnica delle Marche (https://iris.univpm.it). When citing, please refer to the published version.

(Article begins on next page)

05 October 2026

Noname manuscript No.
(will be inserted by the editor)

Cryptanalysis of a code-based full-time signature

Nicolas Aragon · Marco Baldi · Jean-Christophe Deneuville · Karan Khathuria · Edoardo Persichetti · Paolo Santini

Received: date / Accepted: date

Abstract We present an attack against a code-based signature scheme based on the Lyubashevsky protocol that was recently proposed by Song, Huang, Mu, Wu and Wang (SHMWW). The private key in the SHMWW scheme contains columns coming in part from an identity matrix and in part from a random matrix. The existence of two types of columns leads to a strong bias in the distribution of set bits in produced signatures. Our attack exploits such a bias to recover the private key from a bunch of collected signatures. We provide a theoretical analysis of the attack along with experimental evaluations, and we show that as few as 10 signatures are enough to be collected for successfully recovering the private key. As for previous attempts of adapting Lyubashevsky's protocol to the case of code-based cryptography, the SHMWW scheme is thus proved unable to provide acceptable security. This confirms that devising secure code-based signature schemes with efficiency comparable to that of other post-quantum solutions (e.g., based on lattices) is still a challenging task.

This work was partially funded by the French DGA. Karan Khaturia was supported by University of Zurich Forschungskredit grant no. FK-19-080. Edoardo Persichetti was supported by the U.S. National Science Foundation grant CNS-1906360.

N. Aragon
XLIM-MATHIS, University of Limoges, Limoges, France
E-mail: nicolas.aragon@unilim.fr

M. Baldi and P. Santini
Department of Information Engineering, Marche Polytechnic University, Ancona, Italy
E-mail: {m.baldi,p.santini}@staff.univpm.it

J.-C. Deneuville
Ecole Nationale de l'Aviation Civile, University of Toulouse, Toulouse, France
E-mail: jean-christophe.deneuville@enac.fr

K. Khathuria
Institute of Mathematics, University of Zurich, Zurich, Switzerland
E-mail: karan.khathuria@math.uzh.ch

E. Persichetti
Department of Mathematical Sciences, Florida Atlantic University, Boca Raton FL, USA
E-mail: epersichetti@fau.edu

Keywords Post-Quantum Cryptography · Coding Theory · Digital Signature · Cryptanalysis

Mathematics Subject Classification (2010) 94A60 · 11T71 · 14G50

1 Introduction

Digital signature schemes are a class of cryptographic primitives designed to provide a digital equivalent to their paper counterpart, namely to authenticate the original issuer of a document. Efficient constructions of signature schemes have been proposed alongside the advent of public key cryptography [23]. Since then, a long line of research has aimed at making these constructions more efficient, by reducing the public key size and/or shortening the signature. While many well-established and widespread signature schemes rely on integer factorization, the most efficient constructions rely on the intractability of extracting discrete logarithms over the additive group of points on an elliptic curve. In 1994, assuming the existence of a sufficiently large quantum computer, Shor [25] presented an algorithm to solve both problems in polynomial time (as opposed to the best known classical algorithms, that require sub-exponential time). Finding quantum-safe alternatives to cryptosystems relying on the hardness of number theory problems is therefore of prime importance.

Among the quantum-safe alternatives, schemes based on Euclidean lattices and error-correcting codes stand as the most promising candidates. The latter defines the area known as code-based cryptography, which was initiated by McEliece [18] in 1978, and essentially relies on the intractability of decoding random linear codes, a problem that has been proved to be NP-complete [9]. While it is relatively easy to build secure code-based public-key encryption schemes (for which the original McEliece approach is still robust), obtaining efficient and secure digital signature schemes using the standard code-based approach (Hamming metric and syndrome decoding) is considerably more challenging.

Two methods are commonly used to design such schemes. The first one, the "hash-and-sign" paradigm that works very well for some traditional primitives (e.g. RSA), appears to be rather inadequate for code-based schemes. In fact, when relying on the hardness of decoding in the Hamming metric [9, 7], the difficulty of efficiently sampling decodable syndromes leads to protocols that are either inefficient or insecure (or both). CFS [12], which historically dates as the first one in this category, is still technically unbroken (despite the introduction of a distinguisher [15]) but fails to be practical due to its long signing times and large key sizes. The latest hash-and-sign scheme, Wave [13], follows a new approach based on decoding of vectors of very large weight. In Wave, the public-key size grows quadratically in the security parameter, which is an important improvement over CFS. However, Wave still requires a public key of over 3 megabytes for 128 bits of classical security, and signing times of about 0.3 seconds. The second method, which consists of converting an identification scheme via Fiat-Shamir, typically results in very long signatures, due to the necessity of repeating the underlying Sigma protocol many times. The first code-based scheme of this type was proposed

by Stern [27] in '93, and the approach was successively refined through several subsequent works [28, 11, 1, 8, 10]. Yet, the signature sizes that one can obtain with this approach are still not optimal.

A very promising solution, for lattice-based schemes, was given by Lyubashevsky in [16], leading to one of the top contenders for NIST's Post-Quantum standardization effort [19], Dilithium [17]. The paradigm consists of a "one-round" application of an identification scheme à la Schnorr. This allows to obtain very compact signature sizes, as well as a simple and efficient signing procedure. As a consequence, there is a long history of works trying to adapt Lyubashevsky's protocol to the case of code-based cryptography. A first attempt was given by Persichetti [20], concluding that a simple conversion using both the traditional Hamming metric and the rank metric was unlikely to succeed. A subsequent work [21], using quasi-cyclic codes and restricting to one-time usage, was susceptible to a similar attack [24, 14]. Finally, the authors in [2] present a solution based on the rank metric, including a slight modification of the Lyubashevsky protocol (with an additional masking error component), which appears to be secure and offers reasonable performance. However, there are still some doubts about information leakage in the scheme, and the security reduction leads to a rather convoluted, ad-hoc problem (named PSSI$^+$). Moreover, schemes based on the rank metric have shown vulnerabilities in recent times [5, 6], which have undermined the community's confidence in this setting. In the end, the problem of adapting the Lyubashevsky protocol through a decoding problem in the Hamming metric (which has been studied for decades and is now well-understood) is still open.

Contributions. In this paper we cryptanalyze the SHMWW scheme proposed in [26], which is another attempt at adapting the Lyubashevsky framework to coding theory. The peculiarity of the SHMWW scheme consists in the structure of the private key, which is constructed according to an ad-hoc procedure that ensures the low weight of the signatures (this feature is at the core of the security proof). However, the authors of [26] have not considered that the distribution of set bits in the produced signatures is highly biased, according to the secret structure. This information leakage can be exploited to mount a full key-recovery attack, which can determine the private key after collecting a certain number of valid signatures. In light of our results, the SHMWW scheme can only be considered secure for one-time usage (at best); more generally, this work represents another evidence of the fact that the Lyubashevsky framework appears to be not well-suited for coding theory.

Techniques. Our proposed cryptanalysis of the SHMWW scheme can be divided into two steps. After having collected few signatures, one can perform a statistical test to distinguish between columns of weight one and the other columns in the private key. This knowledge is then used to drive the information set choice in ISD algorithms: this way, the success probability for each ISD iteration becomes extremely high, and very few iterations are needed to recover each row of the private key. We first provide a theoretical analysis of a basic version of our attack, and show that it runs in time which is polynomial in the scheme parameters (this result, which comes with a closed formula for the running time of the attack, is

summarized in Proposition 4). Yet, this theoretical analysis is strongly conservative: as we show in Section 5 with supporting experiments, the scheme can actually be broken with as little as 10 signatures (even 6 signatures are enough for attacking PARA-1 with a few days of running time). With as few as 32 signatures, the cryptanalysis successfully returns the secret key within 2 minutes for PARA-1 and 1 hour for PARA-2.

Related work. The two independent works [4] and [3] described a similar strategy for efficiently attacking the SHMWW signature scheme. Starting from those works, we present a unified cryptanalysis approach and an extended set of results.

2 Background and Notation

We start by introducing the notation used in this paper, which is kept as close as possible to that used in [26].
We denote with $\mathbb{F}_q$ the finite field of q elements. We use bold upper case (resp. lower case) letters to denote matrices (resp. vectors). The identity matrix of size $n \times n$ is denoted by $\mathbf{I}_n$. Vectors are measured using the Hamming metric, and the Hamming weight of a vector $\mathbf{x}$ is denoted by $\mathrm{wt}(\mathbf{x})$. The notation $\mathcal{V}_{n,w,q}$ indicates the set of all vectors of length n and Hamming weight w, with components in $\mathbb{F}_q$. When the underlying field is clear from the context, this notation is simplified to $\mathcal{V}_{n,w}$. We use $\mathfrak{B}(\rho)$ to denote the Bernoulli distribution with parameter ρ, and will write $x \sim \mathfrak{B}(\rho)$ to denote that x is a random variable distributed according to $\mathfrak{B}(\rho)$.

3 The SHMWW Signature Scheme

In this section we briefly recall the scheme in [26] and describe its main features. Public parameters are the integers $n, k, n', k', \ell, w_1, w_2, d_{GV}$, whose meaning will be clarified next. The scheme operates over the binary field, hence, for the remainder of this work, we will restrict our attention to the case $q = 2$. The scheme also uses a "weight restricted" hash function $\mathsf{WRH} : \{0,1\}^* \to \mathcal{V}_{k',w_1}$, *i.e.* a hash function that returns digests of length k' and fixed weight w_1, which is not a novelty in code-based cryptography.

Essentially, the authors propose a matricial version of the basic scheme described in [20, Table 7.17], where the private key, instead of consisting of a single low-weight vector, is formed as a "low-weight" matrix, where by this we mean a matrix with a large number of zero entries. This is obtained by juxtaposing the systematic generator matrices $\mathbf{E}_1, \ldots, \mathbf{E}_\ell$ of ℓ distinct $[n', k']$ codes; the presence of the zeros is guaranteed by the identity matrix that appears as the leftmost block of a generator in systematic form. The matrix is then scrambled via both row and column permutations (the matrices $\mathbf{P}_1$ and $\mathbf{P}_2$, respectively) so that the final secret $\mathbf{E} = \mathbf{P}_1 \left[\mathbf{E}_1 | \cdots | \mathbf{E}_\ell\right] \mathbf{P}_2$ is essentially a large code (of length $n = n'\ell$) which should be, in the authors' intention, uncorrelated to the smaller codes forming it. The public key consists of a parity-check matrix $\mathbf{H}$ of a random $[n, k]$ code, and the matrix $\mathbf{S} = \mathbf{H}\mathbf{E}^\top$.

Algorithm 1 KeyGen

Input: Public parameters $\mathsf{params} = (n, k, n', k', \ell, w_1, w_2, d_{GV})$.

Output: $(\mathsf{pk}, \mathsf{sk})$ with $\mathsf{pk} = (\mathbf{H}, \mathbf{S}) \in \mathbb{F}_2^{(n-k)\times n} \times \mathbb{F}_2^{(n-k)\times k'}$ and $\mathsf{sk} = \mathbf{E} \in \mathbb{F}_2^{k'\times n}$

1: Sample $\mathbf{H} \xleftarrow{\$} \mathbb{F}_2^{(n-k)\times n}$

2: For $i = 1, \ldots, \ell$, sample $\mathbf{R}_i \xleftarrow{\$} \mathbb{F}_2^{k'\times(n'-k')}$ and set $\mathbf{E}_i \leftarrow (\mathbf{I}_{k'}|\mathbf{R}_i)$

3: Sample uniform random permutation matrices $\mathbf{P}_1, \mathbf{P}_2$ of respective sizes $k' \times k'$ and $n \times n$

4: Set $\mathbf{E} \leftarrow \mathbf{P}_1 [\mathbf{E}_1| \cdots |\mathbf{E}_\ell] \mathbf{P}_2$

5: **return** $\mathsf{pk} = \left(\mathbf{H}, \mathbf{S} = \mathbf{H}\mathbf{E}^\top\right), \mathsf{sk} = \mathbf{E}$

Algorithm 2 Sign

Input: Public key pk, private key sk, and message $\mathbf{m} \in \{0,1\}^*$

Output: Signature $\sigma = (\mathbf{z}, \mathbf{c}) \in \mathbb{F}_2^n \times \mathbb{F}_2^{k'}$ of message $\mathbf{m}$

1: Sample $\mathbf{e} \xleftarrow{\$} \mathcal{V}_{n,w_2}$

2: Compute $\mathbf{s} \leftarrow \mathbf{H}\mathbf{e}^\top$ and $\mathbf{c} \leftarrow \mathsf{WRH}(\mathbf{m} \parallel \mathbf{s})$

3: Set $\mathbf{z} \leftarrow \mathbf{c}\mathbf{E} + \mathbf{e}$

4: **return** $\sigma = (\mathbf{z}, \mathbf{c})$

Algorithm 3 Verify

Input: Public key pk, message $\mathbf{m}$, and signature $\sigma = (\mathbf{z}, \mathbf{c})$

Output: Accept if σ is a valid signature of $\mathbf{m}$, Reject otherwise

1: **if** $\mathrm{wt}(\mathbf{z}) \leq \ell(w_1 + n' - k') + w_2$ **then**

2: Compute $\hat{\mathbf{s}} \leftarrow \mathbf{H}\mathbf{z}^\top - \mathbf{S}\mathbf{c}^\top$

3: **if** $\mathsf{WRH}(\mathbf{m} \parallel \hat{\mathbf{s}}) = \mathbf{c}$ **then**

4: **return** Accept

5: **else**

6: **return** Reject

7: **else**

8: **return** Reject

Fig. 1: Song *et al.* code based proposal [26].

To sign a message $\mathbf{m}$, a mask $\mathbf{e}$ of small weight w_2 is sampled uniformly at random, then committed by its syndrome, together with the message, to get the challenge $\mathbf{c} = \mathsf{WRH}\left(\mathbf{m} \parallel \mathbf{H}\mathbf{e}^\top\right)$. The response $\mathbf{z}$ to this challenge is the product of the private key and the challenge, hidden by the committed mask: $\mathbf{z} = \mathbf{c}\mathbf{E} + \mathbf{e}$. The signature σ consists of the challenge and the response: $\sigma = (\mathbf{z}, \mathbf{c})$. Note that no rejection sampling is performed during the signing process, unlike the original version of Lyubashevsky. Verification then proceeds accordingly with the dimensions of the objects in question, with the low "weight" of the secret matrix $\mathbf{E}$ guaranteeing the low Hamming weight of the first component of the signature (the response vector $\mathbf{z}$). The second component (the challenge vector $\mathbf{c}$) is formed via the weight restricted hash function to ensure the final Hamming weight is below the desired threshold (parameters are chosen such that this is slightly above the GV bound). The algorithms comprising the SHMWW signature scheme are presented in detail in Fig. 1.

Instance	n	k	$n-k$	ℓ	n'	k'	$n'-k'$	w_1	w_2	$d=d_{GV}$	λ
Para-1	4096	539	3557	4	1024	890	134	31	531	1191	80
Para-2	8192	1065	7127	8	1024	880	144	53	807	2383	128

Table 1: Original SHMWW parameters [26] for λ bits of security.

instance	keygen	sign	verif
PARA-I	415.98	3.81	4.48
PARA-II	2,197.27	17.00	19.45

Table 2: Running times (ms) for the SHMWW signature scheme primitives. The timings were obtained by generating 10^3 key generations, for each of which we generated 10^3 signatures and verified them. Notice that the message signed was directly sampled as a vector of small weight w_1, instead of resorting to a weight restricted hash function as described in [26].

Parameter selection. In [26], the authors study the impact of applying Prange's Information Set Decoding (ISD) algorithm for both "direct and indirect" key-recovery attacks. This essentially provides parameters n, k, d_{GV} and w_2; the other parameters follow by the Gilbert-Varshamov bound and by choosing a value for ℓ:

$$\ell\left(w_1+n'-k'\right)+w_2 \leq d_{GV}. \tag{1}$$

The proposed parameters are recalled in Table 1.

4 Description of the attack

The columns of the private key $\mathbf{E}$ in the SHMWW scheme can be divided into two groups, those due to identities, and those due to random submatrices: we will name the first ones as "identity columns", and the latter ones as "random columns". Finally, we will denote with $\mathcal{I}_R \subset \{1,\ldots,n\}$ the set of integers pointing at random columns. Let us represent the permutation defined by $\mathbf{P}_2$ as $\{i_1, i_2, \cdots, i_n\}$, such that the j-th column is placed in position i_j; then, we have

$$\mathcal{I}_R = \{i_{k'+1},\ldots,i_{n'}, i_{n'+k'+1},\ldots,i_{2(n')},\ldots,i_{(\ell-1)n'+k'+1},\ldots,i_{\ell n'}\}.$$

Note that the row permutation has no impact on the classification of the columns. For the sake of clarity, in Fig. 2 we provide an example of this division for a toy private key where, for simplicity, we have chosen $\mathbf{P}_1 = \mathbf{I}_{k'}$).

At a high level, our attack begins by recovering $\mathcal{I}_R$, *i.e.* the location of random columns; then, exploiting this knowledge, we are able to recover each row of the secret $\mathbf{E}$ using simple linear algebra. In the next sections we formalize this procedure and provide a detailed analysis of its computational complexity.

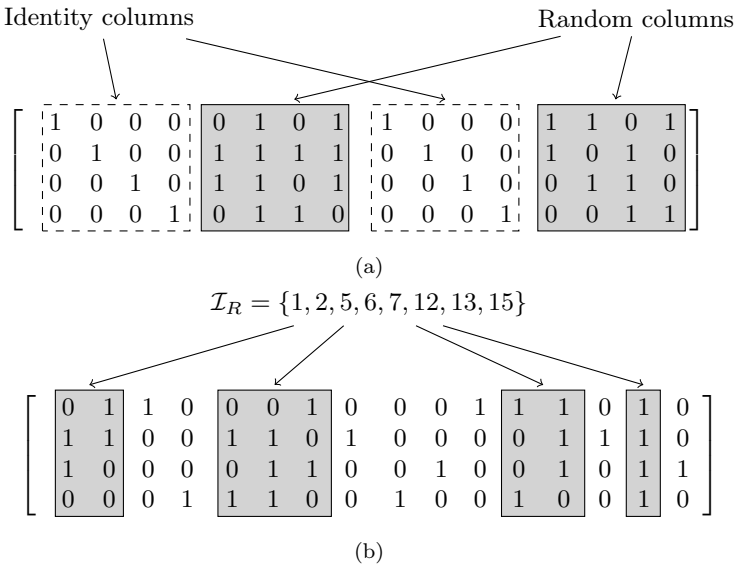

Fig. 2: Example of separation of identity and random columns, for a private key with $n' = 8$, $k' = 4$ and $\ell = 2$. Figure (a) shows the matrix $[\mathbf{E}_1|\mathbf{E}_2]$, while Figure (b) displays the private key after application of the permutations $\mathbf{P}_1$ and $\mathbf{P}_2$. In this example, we have chosen $\mathbf{P}_1$ equal to the identity and $\mathbf{P}_2$ being the matrix corresponding to the permutation $\{3, 8, 10, 4, 1, 15, 5, 13, 11, 14, 16, 9, 2, 7, 6, 12\}$
.

4.1 Leakage from the signatures

The existence of two types of columns in the private key leads to a strong bias in the distribution of set bits in produced signatures, as we highlight in the following proposition.

Proposition 1 *Let* $\mathbf{E}$ *be the private key and* $\mathbf{z} = (z_1, \ldots, z_n) = \mathbf{cE} + \mathbf{e}$ *be a signature. Further, let* $\mathcal{I}_R$ *be the set of random columns of* $\mathbf{E}$*. Then we have:*

- $\rho_R = \Pr[z_i = 1] = \frac{1}{2}$ *if* $i \in \mathcal{I}_R$*;*
- $\rho_I = \Pr[z_i = 1] = \frac{w_1}{k'} + \frac{w_2}{n}(1 - 2\frac{w_1}{k'})$ *otherwise.*

Proof We know that $\mathbf{z} = \mathbf{cE} + \mathbf{e}$, where $\mathbf{c}$ is a vector of length k' and weight w_1 and $\mathbf{e}$ is a vector of length n and weight w_2. Since $w_1 \ll \frac{k'}{2}$, $\mathbf{c}$ has a much lower weight than a random vector of the same length.

We first study the weight of each coordinate of the vector $\mathbf{z}' = \mathbf{cE}$. Let z'_i be the i-th coordinate of $\mathbf{z}'$; there are two possibilities:

- if $i \in \mathcal{I}_R$, *i.e.* if the i-th column of $\mathbf{E}$ is a random one, then $z'_i = 1$ with probability $\frac{1}{2}$;
- if $i \notin \mathcal{I}_R$, *i.e.* if the i-th column of $\mathbf{E}$ is an identity one, then $z'_i = 1$ with probability $\frac{w_1}{k'}$.

	Para-1	Para-2
ρ_R	0.5	0.5
ρ_I	0.155	0.147

Table 3: Values of $\Pr[z_i = 1]$ for the SHMWW parameter sets

Now we want to compute the probability $\Pr[z_i = 1]$ that the i-th coordinate of $\mathbf{z}$ is of weight 1. Since $\mathbf{z}'$ and $\mathbf{e}$ are independent we have

$$\begin{aligned}\Pr[z_i = 1] &= \Pr[z'_i = 1] + \Pr[e_i = 1] - 2 \cdot \Pr[z_i = 1 \wedge e_i = 1] \\ &= \Pr[z'_i = 1] + \Pr[e_i = 1]\big(1 - 2 \cdot \Pr[z'_i = 1]\big)\end{aligned}$$

Which gives the result by replacing $\Pr[z'_i = 1]$ by either $\frac{1}{2}$ or $\frac{w_1}{k'}$ depending on whether i belongs to $\mathcal{I}_R$ or not, and $\Pr[e_i = 1]$ by $\frac{w_2}{n}$.

□

Table 3 shows the values of $\Pr[z_i = 1]$ for the two SHMWW parameter sets that have been proposed in [26]. As a consequence of Proposition 1, we can distinguish between random and identity columns: when acquiring multiple signatures, the coordinates z_i for which, on average, their weight is lower than $\frac{1}{2}$ are more likely to be the coordinates corresponding to columns of weight 1. To provide an evidence of this fact, we have run numerical simulation on a random Para-1 instance; we have generated 1,000 signatures and, for each $i \in \{1, \ldots, n\}$, we have computed the relative frequency with which the i-th entry is set. The obtained results are displayed in Fig. 3.

In practice, one can guess $\mathcal{I}_R$ with a simple threshold criterion, which is applied after the observation of a bunch of honest signatures produced with the same key pair. Let N be the number of collected signatures, and denote with $(z_i)_j$ the i-th bit of the j-th collected one. For each $i \in \{1, \ldots, n\}$, the adversary can compute $\mu_i = \sum_{j=1}^{N} (z_i)_j$ and then apply the following rule

$$\begin{aligned}\mu_i \geq \delta N &\implies \text{guess } i \in \mathcal{I}_R, \\ \mu_i < \delta N &\implies \text{guess } i \notin \mathcal{I}_R,\end{aligned}$$

where $\delta \in \left(0; \frac{1}{2}\right)$.

A correct guess on $\mathcal{I}_R$ will be made if the values of μ_i are all $\geq \delta N$ for $i \in \mathcal{I}_R$, and are all lower than δN for the remaining indexes. We now derive the confidence level of this guessing phase, that is, the probability of making a correct guess for all indexes, as a function of the number of collected signatures N. To do this, we model each μ_i as the sum of N independent random variables, following a Bernoulli distribution whose parameter depends on whether $i \in \mathcal{I}_R$ or not. We recall Proposition 1 and, for a generic $i \in \mathcal{I}_R$, we estimate the probability of making a wrong guess as

$$\begin{aligned}\epsilon_R &= \Pr\left[\sum_{u=1}^{N} x_u < \delta N \mid x_u \sim \mathfrak{B}\left(\rho_R = 1/2\right)\right] \\ &= 2^{-N} \cdot \sum_{u=0}^{\lfloor \delta N \rfloor} \binom{N}{u}. \end{aligned} \tag{2}$$

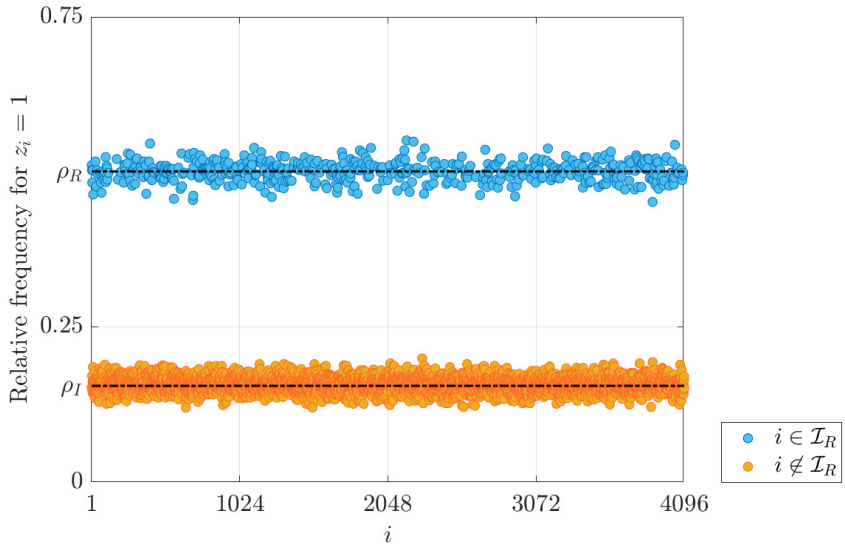

Fig. 3: Relative frequency of $z_i = 1$ occurrences, for a random SHMWW Para-1 instance; for the experiment, we have considered a randomly generated private key and 1,000 signatures.

In an analogous way, in the case of $i \notin \mathcal{I}_R$, we have that each μ_i is the sum of N Bernoulli variables with parameter $\rho_I = \frac{w_1}{k'} + \frac{w_2}{n}\left(1 - 2\frac{w_1}{k'}\right)$; thus, we estimate the probability of wrongly guessing as

$$\begin{aligned} \epsilon_I &= \Pr\left[\sum_{u=1}^{N} x_u \geq \delta N \mid x_u \sim \mathfrak{B}\left(\rho_I\right)\right] \\ &= \sum_{u=\lceil \delta N \rceil}^{N} \binom{N}{u} \rho_I^u (1-\rho_u)^{N-u}. \end{aligned} \tag{3}$$

Assuming that all values μ_i are independent, we have that the confidence level of the statistical test in the guessing phase, *i.e.* the probability of correctly guessing all indexes, is

$$\alpha = (1-\epsilon_R)^{|\mathcal{I}_R|}(1-\epsilon_I)^{n-|\mathcal{I}_R|} = (1-\epsilon_R)^{\ell(n'-k')}(1-\epsilon_I)^{\ell k'}. \tag{4}$$

It is intuitively seen that, for an appropriate choice of δ, the confidence level α rapidly grows with N; to further provide an understanding of this fact, we consider the following proposition.

Proposition 2 *Let us assume that the values* $\mu_i = \sum_{j=1}^{N}(z_i)_j$ *are independent and uncorrelated random variables. Let* $\alpha^* \in (0;1)$, $\delta \in \left(0; \frac{1}{2}\right)$ *and*

$$N^* = \max\left\{ \frac{4}{(1-2\delta)^2} \ln\left(\frac{2\ell(n'-k')}{1-\alpha^*}\right), \frac{(\delta+\rho_I)}{(\delta-\rho_I)^2} \ln\left(\frac{2\ell k'}{1-\alpha^*}\right) \right\}.$$

Then, the confidence level of the test, i.e. *the probability of correctly guessing whether $i \in \mathcal{I}_R$ or not for all $i \in \{1, \cdots, n\}$, using δ as threshold and $N \geq N^*$ as the number of collected signatures, is not lower than α^*.*

The proof of the proposition, which makes use of the well known Chernoff bound, is provided in Appendix A.

4.2 ISD complexity with the knowledge of positions of random columns

Once $\mathcal{I}_R$ is known, we can recover $\mathbf{E}$ line by line by applying any Information Set Decoding (ISD) algorithm, such as Prange's algorithm [22]. We briefly recall the definition of an information set and Prange's algorithm. An information set of an $[n, k]$ code is a subset $\mathcal{I}$ of $\{1, \ldots, n\}$ such that the columns of a parity-check matrix $\mathbf{H}$ indexed outside $\mathcal{I}$ form a non-singular matrix. Given as an input a parity-check matrix $\mathbf{H}$ and a syndrome $\mathbf{s}$, Prange's algorithm finds an error vector $\mathbf{e}$ of given weight w such that $\mathbf{H}\mathbf{e}^\top = \mathbf{s}^\top$. The algorithm is based on the fact that if the support of the error vector $\mathbf{e}$ lies outside an information set, then the error vector can be recovered in polynomial time by solving a linear system of $n - k$ equations in $n - k$ variables.

In order to recover the j-th line of $\mathbf{E}$, we apply Prange's algorithm on the parity-check matrix $\mathbf{H}$ with the j-th column of $\mathbf{S}$ as the syndrome. In addition, we choose an information set $\mathcal{I}$ such that $\mathcal{I}_R \subset \mathcal{I}$. This way we maximize the probability that every non-zero coordinates of the line we are trying to recover lies outside the information set.

Proposition 3 *The probability p that the ℓ non-zero coordinates of $\mathbf{E}$ (the ones from the non-random columns) are included in $\mathcal{I}$ is:*

$$p = \frac{\binom{n-k-(n'-k')\cdot\ell}{\ell}}{\binom{n-(n'-k')\cdot\ell}{\ell}}. \tag{5}$$

Proof By choosing an information set $\mathcal{I}$ such that $\mathcal{I}_R \subset \mathcal{I}$, we have to choose $|\mathcal{I}| - |\mathcal{I}_R| = n - k - (n' - k') \cdot \ell$ columns at random and hope that the ℓ remaining non-null coordinates (from the identity matrices) are included in this set.

From this we deduce that the probability of success is the probability that the ℓ non-null coordinates that are distributed in $n-(n'-k')\cdot\ell$ positions are included in an information set of size $n - k - (n' - k') \cdot \ell$, hence the result.

□

We are now going to estimate the complexity of recovering the private key $\mathbf{E}$ given the knowledge of the set $\mathcal{I}_R$.

Proposition 4 *Given the knowledge of $\mathcal{I}_R$, recovering the private key $\mathbf{E}$ costs $\frac{k'(n-k)^3}{0.2887 \cdot p}$ operations on average.*

Proof The complexity of solving a linear system to recover a line of $\mathbf{E}$ is $(n-k)^3$. Since the SHMWW scheme only uses binary matrices, the probability that the matrix defining said linear system is invertible can be estimated as $\prod_{i=1}^{n-k} 1-2^{-i} \approx 0.2887$, and the probability p that the system gives the correct solution is given by Proposition 3.
This has to be repeated for each of the k' lines of $\mathbf{E}$, which gives the complexity in the thesis. □

Input: $\mathbf{H}, \mathbf{S}$, a threshold value δ, a set of signatures $(\sigma_1, \ldots, \sigma_N) = ((\mathbf{z}_1, \mathbf{c}_1), \ldots, (\mathbf{z}_N, \mathbf{c}_N))$
Output: the secret matrix $\mathbf{E}$

1. $\mathcal{I}_R = \emptyset$
2. For each i from 1 to n:
   - compute $\mu_i = \sum_{j=1}^{N} (\mathbf{z}_j)_i$
   - if $\mu_i > N \cdot \delta$ then $\mathcal{I}_R = \mathcal{I}_R \cup \{i\}$
3. For each i from 1 to k':
   - recover the i-th line of $\mathbf{E}$ by using an ISD algorithm and the knowledge of $\mathcal{I}_R$
4. Return $\mathbf{E}$

Fig. 4: Private key recovery of the SHMWW scheme

4.3 Results

Taking into account the results we have discussed in the previous section, we are now ready to present a complete attack on the scheme. First, for the sake of completeness, in Fig. 4 we report the full procedure we use to attack the SHMWW scheme. The work factor of an adversary attacking the scheme with this algorithm is estimated in the next proposition.

Proposition 5 *For each fixed $\alpha \in (0; 1)$ and $\delta \in (\rho_I; 1/2)$, the algorithm described in Fig. 4 with a number of signatures equal to N^* (as defined in Proposition 2) returns the correct private key with probability at least α, and has an average running time not greater than*

$$n(N^* + 1) + \frac{k'(n-k)^3}{0.2887 \cdot p},$$

where p is computed as in Proposition 3.

Proof In the first step (*i.e.* instructions 1-2), the set $\mathcal{I}_R$ is guessed. To do this, for each $j \in \{1, \ldots, n\}$, one first computes μ_i (which costs N^* operations), and then applies a threshold criterion, whose cost can be assumed to be equal to one elementary operation. This justifies the first part of the complexity, while the second part simply corresponds to that of recovering the rows of $\mathbf{E}$ through Prange's ISD (see Proposition 4). For the success probability of the algorithm, we recall the analysis of Section 4.1: to obtain a confidence level of α, less than N^* signatures are needed. Then, using N^* as the number of collected signatures allows us to derive a conservative estimate on the algorithm complexity.

□

We are now able to assess the complexity of our attack on the proposed instances of the SHMWW scheme, targeting a confidence level of $\alpha = 0.9$:

- for the Para-1 instance, designed for 80 bits of security, we choose $\delta = 0.3005$, yielding to $N^* = 250$; with these choices, our attack requires no more than 2^{48} operations;

- for the Para-2 instance, designed for 128 bits of security, we choose $\delta = 0.3015$, yielding to $N^* = 264$; with these choices, our attack requires no more than 2^{52} operations.

4.4 Practical results and further considerations

The results in the previous section, as captured by Proposition 3, already show that the SHMWW scheme can be broken in polynomial time, using a really limited number of signatures. As we have already remarked, the analysis is rather conservative and, in a practical scenario, it is very likely that the attack can be performed with less significant effort; in this section, we motivate this claim with the aim of numerical results.

First, the number of signatures the adversary needs to collect, to reach a desired confidence level, is significantly lower than that estimated as in Proposition 2. Indeed, the expression of N^* is derived with the use of some conservative bounds, so this result is not surprising. To support this claim, we have simulated the guessing phase, for the two originally proposed SHMWW parameters sets [26]. We have considered several values for the number N of collected signatures and, for each value, we have simulated the guessing phase on 1,000 randomly generated keypairs. For each value of N, the value of δ has been chosen as the one maximizing the theoretical estimate of the confidence level expressed by (4). The comparison between the theoretical estimates, and the actual confidence levels obtained through numerical simulations, is shown in Table 4. As we see, there is a very close correspondence between the theoretical values and the numerical ones: this fact constitutes a confirmation for the validity of our theoretical analysis. Furthermore, it is easily seen that the number of signatures to reach a desired confidence levels are actually quite lower than those estimated through Proposition 2. Indeed, to reach $\alpha = 0.9$, we estimated $N^* = 250$ for Para-1 instances, and $N^* = 264$ for the Para-2 instances. As we see from the Table 4, such a confidence level can always be obtained after the collection of a much lower number of signatures.

We finally comment about the fact that, even when some additional indices are guessed inside $\mathcal{I}_R$, there is still some non null probability that an ISD algorithm can correctly return the rows of the private key. In other words, if we choose threshold lower than the optimal one mentioned in Table 4, the statistical test fails (for some positions outside $\mathcal{I}_R$). In this way, we guess some additional indices in $\mathcal{I}_R$, but there is still some non null, and rather high, probability that an ISD algorithm can return the rows of the private key. Thus, the scheme can still be attacked with a significantly lower number of collected signatures as described in the next section.

5 Experimental results for the cryptanalysis of both parameter sets

To provide an evidence that the number of signatures required to successfully break the scheme is significantly lower than the theoretical value obtained in the previous section, we have run our cryptanalysis with different numbers of signatures available to the adversary for both parameter sets. For PARA-1, all cryptanalyses

	Para-1			Para-2		
N	δ	Th. α	Emp. α	δ	Th. α	Emp. α
10	0.300439	$6.01 \cdot 10^{-200}$	0	0.300872	$1.22 \cdot 10^{-383}$	0
30	0.333439	$2.24 \cdot 10^{-31}$	0	0.300872	$1.20 \cdot 10^{-51}$	0
50	0.320439	$3.99 \cdot 10^{-6}$	0	0.300872	$2.39 \cdot 10^{-9}$	0
70	0.314439	$9.18 \cdot 10^{-2}$	0.187	0.300872	$2.73 \cdot 10^{-2}$	0.076
90	0.311439	0.616	0.648	0.300872	0.508	0.565
110	0.309439	0.903	0.923	0.300872	0.878	0.9
130	0.308439	0.978	0.984	0.307872	0.976	0.98
150	0.313439	0.996	1	0.306872	0.995	0.998
170	0.312439	0.999	1	0.306872	0.999	1
190	0.311439	0.999	1	0.305872	0.999	1

Table 4: Confidence level of the guessing phase on the original SHMWW parameters [26], for several values of the number N of collected signatures. For each value of N, the value of δ has been chosen as the one maximizing the confidence level α expressed by (4). To numerically estimate the success rate, for each value of N, we have run the guessing phase on 1,000 randomly generated key-pairs.

ran with 6 signatures or more were successful. This number had to be slightly increased for PARA-2 in order for the crytanalysis to complete within a week. As the number of available signatures increases, the execution timings quickly become very reasonable (minutes for PARA-1, hours for PARA-2). All the experiment results are reported on Fig. 5 for PARA-1 (targeting 80 bits of security) and Fig. 6 for PARA-2 (128 bits of security).

For these experiments, we use a threshold value obtained as a balanced combination of ρ_R and ρ_I in Prop. 1, where the weights correspond to the number of occurrences of each column type in $\mathbf{E}$:

$$\delta = \left\lfloor \frac{\ell k'}{n} \cdot \left[\frac{w_1}{k'} + \frac{w_2}{n} \left(1 - 2\frac{w_1}{k'}\right) \right] + \frac{\ell \left(n' - k'\right)}{n} \frac{1}{2} \right\rfloor . \tag{6}$$

We provide the resulting threshold for some numbers of collected signatures in Tab. 5.

	Number N of available signatures									
	10	16	24	32	64	128	160	192	224	256
PARA-1	2	3	6	9	12	25	32	38	44	51
PARA-2	1	3	6	9	12	25	31	37	44	50

Table 5: Experimental threshold values $N \cdot \delta$ to determine whether a column is random or not, according to Eq. (6).

Experiments were run over an Intel® Xeon® Gold 6230 CPU 2.10GHz with Ubuntu 18.04, GCC 7.5.0 with compilation flags `-O3`, NTL 11.4.3, and gf2x 1.3.0. The reported execution timings have been averaged over 1000 executions. Both our

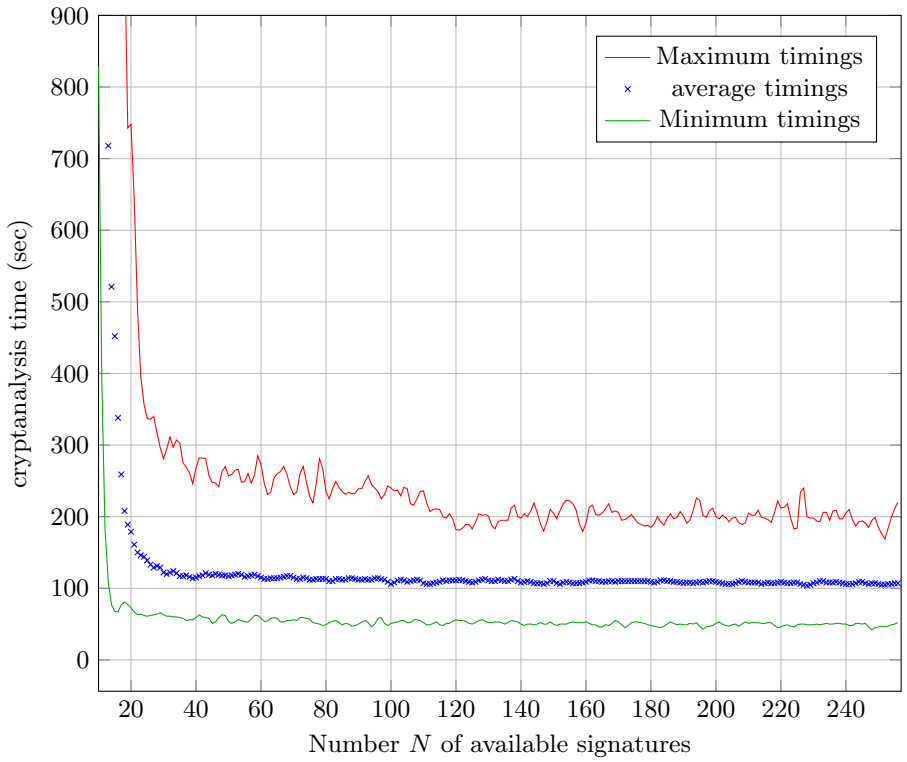

Fig. 5: Execution timings (sec) for breaking PARA-1 as a function of the number N of signatures available to the adversary. Timings were averaged over a thousand executions.

implementation of the SHMWW scheme (without WRF) and the cryptanalysis are available at: https://github.com/deneuville/cryptanalysisSHMWW_C

6 Conclusion

We have presented an efficient cryptanalysis of the signature scheme recently proposed by Song *et al.* in [26], adapting Lyubashevsky's framework to coding theory. Our attack affects both parameter sets, and given its asymptotic complexity, discourages further parameter tweaks to patch this signature scheme. Our results are supported by a theoretical analysis and proof-of-concept implementations of the SHMWW signature scheme and its cryptanalysis. For both parameter sets, our attack requires as little as 10 signatures to fully recover the private key. Our results prove that the SHMWW signature scheme does not reach its claimed security, and should not be considered secure for more than one-time use.

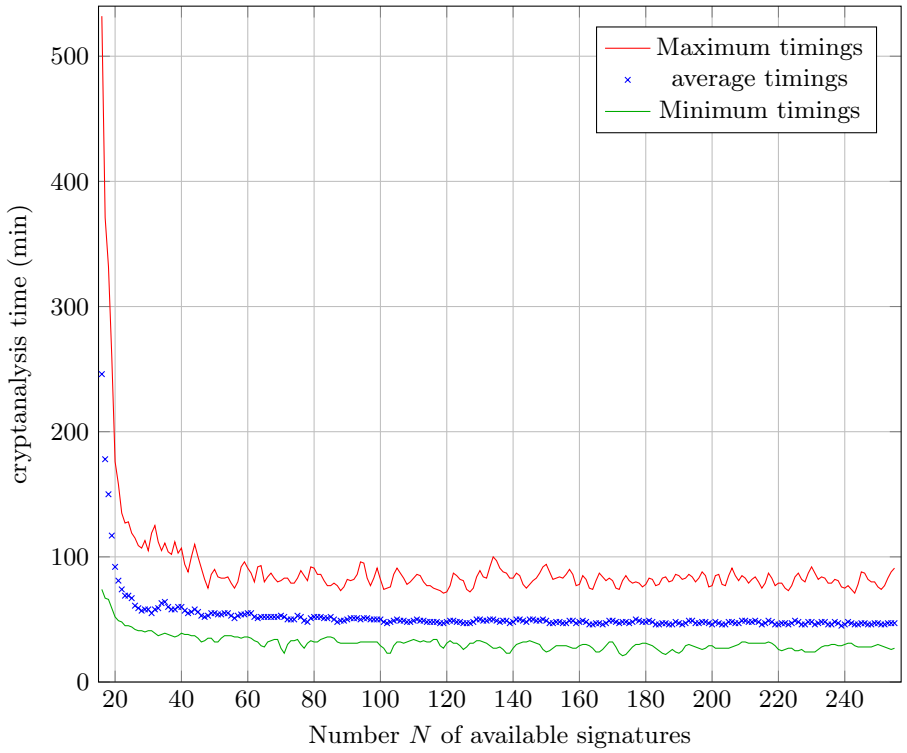

Fig. 6: Execution timings (min) for breaking PARA-2 as a function of the number N of signatures available to the adversary. Timings were averaged over a hundred executions.

Acknowledgement

The authors thank Philippe Gaborit for insightful discussions on preliminary versions of this work.

References

1. Aguilar C, Gaborit P, Schrek J (2011) A new zero-knowledge code based identification scheme with reduced communication. In: 2011 IEEE Information Theory Workshop, pp 648–652, DOI 10.1109/ITW.2011.6089577
2. Aragon N, Blazy O, Gaborit P, Hauteville A, Zémor G (2019) Durandal: A rank metric based signature scheme. In: Ishai Y, Rijmen V (eds) Advances in Cryptology – EUROCRYPT 2019, Springer International Publishing, Cham, pp 728–758
3. Aragon N, Deneuville JC, Gaborit P (2020) Another code-based adaptation of lyubashevsky's signature cryptanalysed. Cryptology ePrint Archive, Report 2020/923, https://eprint.iacr.org/2020/923

4. Baldi M, Khathuria K, Persichetti E, Santini P (2020) Cryptanalysis of a code-based signature scheme based on the Lyubashevsky framework. Cryptology ePrint Archive, Report 2020/905, https://eprint.iacr.org/2020/905
5. Bardet M, Briaud P, Bros M, Gaborit P, Neiger V, Ruatta O, Tillich J (2020) An algebraic attack on rank metric code-based cryptosystems. In: Advances in Cryptology - EUROCRYPT 2020 Proceedings, Part III, Springer, LNCS, vol 12107, pp 64–93
6. Bardet M, Bros M, Cabarcas D, Gaborit P, Perlner RA, Smith-Tone D, Tillich JP, Verbel JA (2020) Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In: Moriai S, Wang H (eds) ASIACRYPT 2020, Part I, Springer, Heidelberg, LNCS, vol 12491, pp 507–536, DOI 10.1007/978-3-030-64837-4_17
7. Barg S (1994) Some new NP-complete coding problems. Problemy Peredachi Informatsii 30(3):23–28
8. Bellini E, Caullery F, Gaborit P, Manzano M, Mateu V (2019) Improved Veron identification and signature schemes in the rank metric. In: 2019 IEEE International Symposium on Information Theory (ISIT), pp 1872–1876
9. Berlekamp ER, McEliece RJ, van Tilborg HCA (1978) On the inherent intractability of certain coding problems (corresp.). IEEE Trans Information Theory 24(3):384–386, DOI 10.1109/TIT.1978.1055873
10. Biasse JF, Micheli G, Persichetti E, Santini P (2020) LESS is more: Code-based signatures without syndromes. In: Nitaj A, Youssef A (eds) Progress in Cryptology - AFRICACRYPT 2020, Springer International Publishing, Cham, pp 45–65
11. Cayrel PL, Véron P, El Yousfi Alaoui SM (2011) A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In: Selected Areas in Cryptography, Springer Berlin Heidelberg, pp 171–186
12. Courtois N, Finiasz M, Sendrier N (2001) How to achieve a McEliece-based digital signature scheme. In: Boyd C (ed) ASIACRYPT 2001, Springer, Heidelberg, LNCS, vol 2248, pp 157–174, DOI 10.1007/3-540-45682-1_10
13. Debris-Alazard T, Sendrier N, Tillich JP (2019) Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. In: Galbraith SD, Moriai S (eds) ASIACRYPT 2019, Part I, Springer, Heidelberg, LNCS, vol 11921, pp 21–51, DOI 10.1007/978-3-030-34578-5_2
14. Deneuville JC, Gaborit P (2020) Cryptanalysis of a code-based one-time signature. Designs, Codes and Cryptography 88(9):1857–1866
15. Faugere JC, Gauthier-Umana V, Otmani A, Perret L, Tillich JP (2013) A distinguisher for high-rate mceliece cryptosystems. IEEE Transactions on Information Theory 59(10):6830–6844
16. Lyubashevsky V (2012) Lattice signatures without trapdoors. In: Pointcheval D, Johansson T (eds) EUROCRYPT 2012, Springer, Heidelberg, LNCS, vol 7237, pp 738–755, DOI 10.1007/978-3-642-29011-4_43
17. Lyubashevsky V, Ducas L, Kiltz E, Lepoint T, Schwabe P, Seiler G, Stehlé D (2019) CRYSTALS-DILITHIUM. Tech. rep., National Institute of Standards and Technology, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
18. McEliece RJ (1978) A Public-Key System Based on Algebraic Coding Theory, Jet Propulsion Lab, pp 114–116. DSN Progress Report 44

19. National Institute of Standards and Technology (2017) NIST Post-Quantum Standardization process. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography
20. Persichetti E (2012) Improving the efficiency of code-based cryptography. PhD thesis, Department of Mathematics, University of Auckland
21. Persichetti E (2018) Efficient one-time signatures from quasi-cyclic codes: A full treatment. Cryptography 2:30, DOI 10.3390/cryptography2040030
22. Prange E (1962) The use of information sets in decoding cyclic codes. IRE Trans Inf Theory 8(5):5–9
23. Rivest RL, Shamir A, Adleman LM (1978) A method for obtaining digital signatures and public-key cryptosystems. Communications of the Association for Computing Machinery 21(2):120–126
24. Santini P, Baldi M, Chiaraluce F (2019) Cryptanalysis of a one-time code-based digital signature scheme. In: 2019 IEEE International Symposium on Information Theory (ISIT), pp 2594–2598
25. Shor PW (1994) Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th FOCS, IEEE Computer Society Press, pp 124–134, DOI 10.1109/SFCS.1994.365700
26. Song Y, Huang X, Mu Y, Wu W, Wang H (2020) A code-based signature scheme from the Lyubashevsky framework. Theoretical Computer Science 835:15–30, DOI 10.1016/j.tcs.2020.05.011
27. Stern J (1994) A new identification scheme based on syndrome decoding. In: Stinson DR (ed) Advances in Cryptology — CRYPTO' 93, Springer Berlin Heidelberg, pp 13–21
28. Véron P (1997) Improved identification schemes based on error-correcting codes. Applicable Algebra in Engineering, Communication and Computing 8(1):57–69, DOI 10.1007/s002000050053

A Computing the number of signatures for a desired confidence level

We here prove Proposition 2. To bound the probabilities ϵ_R and ϵ_I which appear in (4) we will use the Chernoff bound, which we recall in the following.

Theorem 1 ***Chernoff bound***
Let $X = \sum_{u=1}^{M} x_u$, where the x_u are all independent and $x_u \sim \mathfrak{B}(\rho)$; then

i) $\Pr[X \leq (1-\gamma)\rho M] \leq e^{-\frac{\gamma^2}{2}\rho M}$, *for all* $0 < \gamma < 1$;

ii) $\Pr[X \geq (1+\gamma)\rho M] \leq e^{-\frac{\gamma^2}{2+\gamma}\rho M}$, *for all* $\gamma > 0$.

Applying condition i) of the Chernoff bound on (2), we have $\rho = \frac{1}{2}$ and $\gamma = 1 - 2\delta$, such that

$$\epsilon_R \leq e^{-\frac{(1-2\delta)^2}{4}N} = \epsilon_R^*. \tag{7}$$

In analogous way, applying condition ii) of the Chernoff bound on (3), we have $\rho = \rho_I$ and $\gamma = \frac{\delta}{\rho_I} - 1$, such that

$$\epsilon_I \leq e^{-\frac{(\delta-\rho_I)^2}{\delta+\rho_I}N} = \epsilon_I^*. \tag{8}$$

Using these bounds for ϵ_R and ϵ_I, we derive the following inequality on the success probability

$$\alpha \geq (1-\epsilon_R^*)^{\ell(n'-k')}(1-\epsilon_I^*)^{\ell k'}.$$

We first note that, regardless of the particular choice for δ, the probabilities ϵ_R^* and ϵ_I^* decay exponentially with N; thus, we can always choose N sufficiently high to make them extremely low. Using a well known approximation, we have

$$(1-\epsilon_R^*)^{\ell(n'-k')} \approx 1-\ell(n'-k')\epsilon_R^*,$$

$$(1-\epsilon_I^*)^{\ell k'} \approx 1-\ell k'\epsilon_I^*.$$

Now, let

$$N \geq N^* = \max\left\{\frac{4}{(1-2\delta)^2}\ln\left(\frac{2\ell(n'-k')}{1-\alpha^*}\right), \frac{(\delta+\rho_I)}{(\delta-\rho_I)^2}\ln\left(\frac{2\ell k'}{1-\alpha^*}\right)\right\}.$$

Then, $N \geq \frac{4}{(1-2\delta)^2}\ln\left(\frac{2\ell(n'-k')}{1-\alpha^*}\right)$ and (7) implies that

$$\epsilon_R^* \leq \frac{1-\alpha^*}{2\ell(n'-k')},$$

and, $N \geq \frac{(\delta+\rho_I)}{(\delta-\rho_I)^2}\ln\left(\frac{2\ell k'}{1-\alpha^*}\right)$ and (8) implies that

$$\epsilon_I^* \leq \frac{1-\alpha^*}{2\ell k'}.$$

Therefore, we obtain the following bound on the probability of success

$$\begin{aligned}
\alpha &\geq (1-\epsilon_R^*)^{\ell(n'-k')}(1-\epsilon_I^*)^{\ell k'} \\
&\approx 1-\ell(n'-k')\epsilon_R^* - \ell k'\epsilon_I^* + \ell^2 k'(n'-k')\epsilon_R^*\epsilon_I^* \\
&\geq 1-\ell(n'-k')\epsilon_R^* - \ell k'\epsilon_I^* \\
&\geq \alpha^*.
\end{aligned}$$