Wave Parameter Selection

Nicolas Sendrier

Inria[**]
nicolas.sendrier@inria.fr

Abstract. Wave is a provably EUF-CMA (existential unforgeability under adaptive chosen message attacks) digital signature scheme based on codes [15]. It is an hash-and-sign primitive and its security is built according to a GPV-like framework [19] under two assumptions related to coding theory: (i) the hardness of finding a word of prescribed Hamming weight and prescribed syndrome, and (ii) the pseudo-randomness of ternary generalized $(U|U+V)$ codes. Forgery attacks (i)—or message attacks—consist in solving the ternary decoding problem for large weight [7], while, to the best of our knowledge, key attacks (ii) will try to exhibit words that are characteristic of $(U|U+V)$ codes, which are called type-U or type-V codewords in the present paper. In the current state-of-the-art, the best known attacks both reduce to various flavours of Information Set Decoding (ISD) algorithms for different regime of parameters. In this paper we give estimates for the complexities of the best known ISD variants for those regimes. Maximizing the computational effort, thus the security, for both attacks lead to conflicting constraints on the parameters. We provide here a methodology to derive optimal trade-offs for selecting parameters for the Wave signature scheme achieving a given security. We apply this methodology to the current state-of-the-art and propose some effective parameters for Wave. For $\lambda = 128$ bits of classical security, the signature is 737 bytes long, scaling linearly with the security, and the public key size is 3.6 Mbytes, scaling quadratically with the security.

1 Introduction

The signature scheme Wave is built from a family of trapdoor one-way the preimage sampleable function [15]. It is a hash-and-sign digital signature scheme and, in contrast with [12], it easily scales with the security parameter. It is provably EUF-CMA under code-based hardness assumptions, namely the hardness of decoding and the indistinguishability of generalized ternary $(U|U+V)$ codes. The scheme enjoys some very attractive features: (1) very short signatures, less than 1 kbyte for 128 bits of security, and (2) fast verification, less than 1 millisecond [2]. The main drawback is a very large public key, of several megabytes.

In this paper we explore the best known attacks on the Wave signature scheme and provide some practical parameters based on the current state-of-the-art, as well as a methodology to adjust those parameters to any target security or if the state-of-the-art evolves. There are two ways to attack the Wave scheme:

[**] Inria de Paris, 2 rue Simone Iff, 75012 Paris

1. Message attacks, or forgery attacks, which consist in solving a generic decoding problem. Those problems are for ternary codes and for large weights, and the best solver [7] uses information set decoding (ISD) [27] combined with Wagner's generalized birthday algorithm (GBA) [31].
2. Structural attacks, or key attacks, which consist in defeating the indistinguishability assumption. To the best of our knowledge, the most efficient way to achieve that is to exhibit codewords in the public code (*i.e.* the code deriving from the public key) that have an expectedly low, or high, weight. Such codewords are likely to be among the, so-called, type-U or type-V codewords, and are susceptible to reveal information about the hidden $(U|U+V)$ structure. In practice, those key attacks boil down to finding codewords of specified weight in a generic code, for which the best solvers again derive from ISD.

We will observe that the two classes of attacks above provide conflicting constraints on the system parameters, and the choice of secure parameters derives from a trade-off between those constraints.

Related Works. Code-based digital signatures first appeared in [30]: the Stern authentication protocol using the Fiat-Shamir transform [17]. The first hash-and-sign signature based on codes was proposed in [12] but does not scale very well, making it unpractical. There has been improvements to Stern's scheme, [1] using cyclic codes and a 5 pass protocol, and more recently [16] using the MPC in the head paradigm [20]. The latter allows schemes with signatures shorter than the latter, around 10 kbytes or a bit less. Those techniques have relatively larger signature sizes compared to Wave, but much smaller public keys. Let us also mention the LESS scheme [4] with similar characteristic but a security based on the hardness of code equivalence rather than the hardness of decoding.

The paper is organized as follows. We first recall in §2 basic information about codes and hard problems, the definition of $(U|U+V)$ codes and their main properties, and a description of the Wave signature scheme. Then in §3 we give a framework to estimate the cost of ISD in the q-ary case, adapted from existing literature, presenting the variants that are relevant for the study of Wave's security. Finally, in §4, we provide estimates for the computational cost of the best known attacks and a methodology to derive practical parameters for the Wave digital signature scheme.

2 Preliminaries

Many statements in this section are often admitted as coding theory folklore. We point out [14] to the reader for a precise and rigorous presentation of some of those statements.

Notation

$\mathbf{F}_q$	The q-ary finite field
$\mathbf{x} \in \mathbf{F}_q^n$	$\mathbf{x} = (x_0, \dots, x_n)_{0 \le i < n} \in \mathbf{F}_q^n$, vectors generally use bold letters
$\lvert\mathbf{x}\rvert$	Hamming weight of $\mathbf{x} \in \mathbf{F}_q^n$, $\lvert\mathbf{x}\rvert = \lvert\{i, 0 \le i < n \mid x_i \neq 0\}\rvert$
$\mathcal{E}_{q;n,t}$	$\mathcal{E}_{q;n,t} = \{\mathbf{e} \in \mathbf{F}_q^n, \lvert\mathbf{e}\rvert = t\}$
$\mathbf{x} \star \mathbf{y}$	Component-wise product $\mathbf{x} \star \mathbf{y} = (x_i y_j)_{0 \le i < n}$, $\mathbf{x}, \mathbf{y} \in \mathbf{F}_q^n$
—	
$\mathbf{M} \in \mathbf{F}_q^{r \times n}$	$(M_{i,j})_{0 \le i < r, 0 \le j < n}$, $r \times n$ matrix over $\mathbf{F}_q$
$\langle \mathbf{M} \rangle$	The vector space spanned by the rows of matrix $\mathbf{M}$
	A matrix $\mathbf{M} \in \mathbf{F}_q^{r \times n}$ is in systematic form if its writes as $\mathbf{M} = (\mathbf{Id} \mid \mathbf{R})$ where $\mathbf{Id}$ is the $r \times r$ identity matrix
Reduced row echelon form	
	A matrix in $\mathbf{F}_q^{r \times n}$ is in reduced row echelon form if its r leftmost independent columns form an identity matrix
$\mathbf{x} \star \mathbf{M}$	Row-wise star product $\mathbf{x} \star \mathbf{M} = (x_j M_{i,j})_{0 \le i < r, 0 \le j < n}$
—	
$\mathcal{S}_n$	Group of permutations of $\{0, \dots, n-1\}$
$\mathbf{x}^\pi$	$\mathbf{x}^\pi = (x_{\pi(i)})_{i \le 0 < n}$, $\mathbf{x} \in \mathbf{F}_q^n$, $\pi \in \mathcal{S}_n$
$\mathbf{M}^\pi$	$\mathbf{M}^\pi = (M_{i,\pi(j)})_{0 \le i < r, 0 \le j < n}$, $\mathbf{M} \in \mathbf{F}_q^{r \times n}$, $\pi \in \mathcal{S}_n$
$\mathcal{X}^\pi$	$\mathcal{X}^\pi = \{\mathbf{x}^\pi \mid \mathbf{x} \in \mathcal{X}\}$, $\mathcal{X} \subset \mathbf{F}_q^n$, $\pi \in \mathcal{S}_n$

2.1 Error Correcting Code

A q-ary linear $[n,k]$ code C is a k-dimensional subspace of $\mathbf{F}_q^n$. A *generator matrix* $\mathbf{G} \in \mathbf{F}_q^{k \times n}$ of C is such that $\langle \mathbf{G} \rangle = C$ and a *parity check matrix* $\mathbf{H} \in \mathbf{F}_q^{(n-k) \times n}$ of C is such that $\langle \mathbf{H} \rangle^\perp = C$, *i.e.*

$$C = \{\mathbf{x}\mathbf{G} \mid \mathbf{x} \in \mathbf{F}_q^k\} \text{ and } C = \{\mathbf{y} \in \mathbf{F}_q^n \mid \mathbf{y}\mathbf{H}^\intercal = 0\}.$$

For any $\mathbf{y} \in \mathbf{F}_q^n$ the quantity $\mathbf{y}\mathbf{H}^\intercal$ is called the *syndrome* of $\mathbf{y}$ (relatively to $\mathbf{H}$). The *dual code* of C is $C^\perp = \langle \mathbf{H} \rangle$, the orthogonal of C.

Weight Distribution. When the code C is chosen uniformly at random, the expected number of its codewords of weight i is asymptotically [14]:

$$W_i(C) = \mathbb{E}\left[\left|\{\mathbf{c} \in C \mid |\mathbf{c}| = i\}\right|\right] = \frac{\binom{n}{i}(q-1)^i}{q^{n-k}}.$$

The actual number of codewords of a specific weight might differ for structured codes. If this difference is measurable, this could be used to distinguish a given code C from random.

2.2 Decoding Problem

The decoding problem over $\mathbf{F}_q$ is defined as follows:

Problem 1 (Decoding Problem – DP$(q; n, k, t)$*).* A finite field $\mathbf{F}_q$ and three integers n, k, t such that $n > k > 0$ and $0 \leq t \leq n$.
Instance: $(\mathbf{H}, \mathbf{s}) \in \mathbf{F}_q^{(n-k)\times n} \times \mathbf{F}_q^{n-k}$
Solution: $\mathbf{e} \in \mathbf{F}_q^n$ such that $|\mathbf{e}| = t$ and $\mathbf{e}\mathbf{H}^\intercal = \mathbf{s}$.

We denote $\mathrm{Dec}(q; \mathbf{H}, \mathbf{s}, t)$ an instance of the above problem and also, for convenience, the set of its solutions.

The problem DP$(q; n, k, t)$ is hard if solving $\mathrm{Dec}(q; \mathbf{H}, \mathbf{s}, t)$ is hard on average with $\mathbf{H}$ uniformly distributed in $\mathbf{F}_q^{(n-k)\times n}$ and $\mathbf{s} = \mathbf{x}\mathbf{H}^\intercal$ with $\mathbf{x}$ uniformly distributed in $\mathcal{E}_{q;n,t}$, the set of words of weight t. When the cardinality of $\mathcal{E}_{q;n,t}$ is (sufficiently) larger than q^{n-k}, this is the same as having $\mathbf{s}$ uniformly distributed $\mathbf{F}_q^{n-k}$, see for instance [14, §2.5].

In practice, when k/n and t/n are positive constants, the best known algorithms have an average complexity exponential in n when $t < \frac{q-1}{q}(n-k)$ or $t > k + \frac{q-1}{q}(n-k)$ and polynomial in n when $\frac{q-1}{q}(n-k) \leq t \leq k + \frac{q-1}{q}(n-k)$.

Codeword Finding. Finding codewords of given weight correspond to instances of DP with a zero syndrome $\mathbf{s}$. This problem is also hard and, in practice and in the general case, solvers are the same as for DP.

Decoding One Out of Many. The variant of the decoding problem with multiple instance is relevant for the security of code-based signature schemes. This problem was considered in the binary case in [21,28].

Problem 2 (DOOM Problem – DP$_N(q; n, k, t)$*).* A finite $\mathbf{F}_q$ field and three integers n, k, t such that $n > k > 0$ and $0 \leq t \leq n$. A number $N > 0$ of instances.
Instance: $(\mathbf{H}, \mathbf{s}_1, \ldots, \mathbf{s}_N) \in \mathbf{F}_q^{(n-k)\times n} \times \left(\mathbf{F}_q^{n-k}\right)^N$
Solution: $\mathbf{e} \in \mathbf{F}_q^n$ such that $|\mathbf{e}| = t$ and $\mathbf{e}\mathbf{H}^\intercal \subset \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$.

By extension, we denote DP$_\infty(q; n, k, t)$ the *unlimited* DOOM problem in which the solver is free to decide the value of N. If we denote $\mathrm{WF_N}$ the average computational effort for solving DP$_N(q; n, k, t)$, we have $\mathrm{WF_N} \geq \max(N, \mathrm{WF}_1/N)$ and it follows that $\mathrm{WF}_\infty \geq \sqrt{\mathrm{WF}_1}$. The unlimited DOOM problem cannot be easy if the corresponding Decoding Problem is hard.

2.3 Generalized Ternary $(U|U + V)$ Code

We consider integers n, k, k_U, k_V with n even such that $n > k > 0$, $k = k_U + k_V$, $0 < k_U < n/2$, and $0 < k_V < n/2$. Let $\mathbf{a} = (a_i)_{0\leq i<n/2}$, $\mathbf{b} = (b_i)_i$, $\mathbf{c} = (c_i)_i$, and $\mathbf{d} = (d_i)_i$ denote vectors in $\mathbf{F}_3^{n/2}$, such that

$$\forall i, 0 \leq i < n/2, \begin{cases} a_i c_i \neq 0 \\ a_i d_i - b_i c_i \neq 0 \end{cases} \tag{1}$$

Let $\mathbf{H}_U \in \mathbf{F}_3^{(\frac{n}{2}-k_U)\times\frac{n}{2}}$ and $\mathbf{H}_V \in \mathbf{F}_3^{(\frac{n}{2}-k_V)\times\frac{n}{2}}$ denote parity check matrices of the ternary linear codes U and V of length $n/2$ and dimension respectively k_U and

k_V. The generalized ternary $(U|U+V)$ code $\mathcal{C}$ associated to $(\mathbf{H}_U, \mathbf{H}_V, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$ admits the following parity check matrix

$$\mathbf{H} = \left(\begin{array}{c|c} \mathbf{d} \star \mathbf{H}_U & -\mathbf{b} \star \mathbf{H}_U \\ \hline -\mathbf{c} \star \mathbf{H}_V & \mathbf{a} \star \mathbf{H}_V \end{array}\right) \in \mathbf{F}_3^{(n-k)\times n} \tag{2}$$

Dual Code. If $\mathcal{C}$ is a generalized ternary $(U|U+V)$ code $\mathcal{C}$ associated to $(\mathbf{H}_U, \mathbf{H}_V, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$, then its dual is also a generalized $(U|U+V)$ code, associated to $(\mathbf{G}_V, \mathbf{G}_U, -\mathbf{c}, \mathbf{d}, \mathbf{a}, -\mathbf{b})$ where $\mathbf{G}_V$ and $\mathbf{G}_U$ are generator matrices of V and U, that is, equivalently, parity check matrices of the dual codes $V^{\perp}$ and $U^{\perp}$.

Decoder for Large Weights. A probabilistic decoding procedure for $\mathcal{C}$ is described in [15]:

$$\begin{aligned} \varPhi_{\mathcal{C},w} : \mathbf{F}_3^{n-k} &\longrightarrow \mathbf{F}_3^n \\ \mathbf{s} &\longmapsto \mathbf{e} \quad \text{such that } \mathbf{e}\mathbf{H}^{\intercal} = \mathbf{s}, |\mathbf{e}| = w \end{aligned} \tag{3}$$

It makes use of the $(U|U+V)$ structure and runs in polynomial time for some weight $w > k + \frac{2}{3}(n-k)$ such that the decoding problem $\mathrm{DP}(3; n, k, w)$ is hard.

2.4 The Wave Signature Scheme

1. Main parameters: code-length n, dimension k, error weight w,
2. Public Key: $\mathbf{H}_{\text{pub}} \in \mathbf{F}_3^{(n-k)\times n}$,
3. Signature: $\mathbf{e} \in \mathbf{F}_3^n$ such that $|\mathbf{e}| = w$ and $\mathbf{e}\mathbf{H}_{\text{pub}}^{\intercal} = \text{Hash}(\text{Message}) \in \mathbf{F}_3^{(n-k)}$.

The matrix $\mathbf{H}_{\text{pub}}$ is the reduced row echelon form of the parity check matrix of a generalized ternary $(U|U+V)$ code $\mathcal{C}$ of length n and dimension k, randomly permuted by $\pi \in \mathcal{S}_n$. The secret key is a decoding procedure, $\varPsi : \mathbf{F}_3^{n-k} \to \mathbf{F}_3^n$, to solve $\mathrm{Dec}(\mathbf{H}_{\text{pub}}, \mathbf{s}, w)$ which easily derives from $\varPhi_{\mathcal{C},w}$, see (3), and the permutation π. It involves the parameters k_U, k_V for the $(U|U+V)$ structure, see §2.3, and g, called the *gap*, which is used by the decoder. We will refer to the integers (k_U, k_V, g) as the secondary parameters, they are public but are only used for signing.

4. The secondary parameters (k_U, k_V, g) verify

$$k_U = g + \frac{3}{2}w - n, k_V = k - k_U \text{ and } 0 \le g \le \frac{\lambda}{\log_2 3} \tag{4}$$

   where λ is the security parameter[1].

An additional condition is required to build a secure signature scheme: essentially the distribution of $\varPsi(\mathbf{s})$, or equivalently of $\varPhi_{\mathcal{C},w}(\mathbf{s})$, must be uniform over $\mathcal{E}_{3;n,w}$ when the input $\mathbf{s}$ is uniformly distributed over $\mathbf{F}_3^{n-k}$. This involves a choice of

[1] *e.g.* $\lambda = 128$ corresponds to a scheme about as secure as AES-128

internal distributions and possibly some rejection sampling. It is shown in [15] that with a gap $g = \lambda / \log_2 3$ it is possible to implement the decoder so that the statistical distance between the distribution of its output and the uniform distribution over $\mathcal{E}_{3;n,w}$ do not exceed $2^{-\lambda}$.

For a security level $\lambda = 128$ the gap must be equal to 81. Depending on the adversarial model and on the security assumptions the value of g may reduce. For instance to $g = \log(Q)/\log_2 3$, where Q is the number of signature queries allowed, for instance $g = 40$ when $Q = 2^{64}$. The gap may even reduce to a small constant. This is hypothetical at the moment, but we will nevertheless consider various scenarios for the value of g and examine how they impact the scheme parameters.

Wave Security Reduction. With the above notations, and assuming that the decoder's output is properly distributed, the Wave signature scheme is EUF-CMA secure under the following assumptions:

1. The unlimited DOOM problem $\mathrm{DP}_\infty(3; n, k, w)$ is hard.
2. Permuted generalized ternary $(U|U+V)$ codes of parameters (n, k_U, k_V) are computationally indistinguishable from random.

2.5 Weight Distribution and $(U|U+V)$-Specific Codewords.

Let $\mathcal{C}$ denote a ternary generalized $(U|U+V)$ code, presumably used as an instance of Wave, associated to $(\mathbf{H}_U, \mathbf{H}_V, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$. We examine below how its weight distribution differs from that of a random code. We denote U and V the codes of respective parity check matrices $\mathbf{H}_U$ and $\mathbf{H}_V$. We call respectively type-U and type-V codewords, the elements of the following subcodes of $\mathcal{C}$

$$\begin{aligned}\mathcal{U}(\mathcal{C}) &= \{(\mathbf{a} \star \mathbf{u}, \mathbf{c} \star \mathbf{u}) \mid \mathbf{u} \in U\},\\ \mathcal{V}(\mathcal{C}) &= \{(\mathbf{b} \star \mathbf{v}, \mathbf{d} \star \mathbf{v}) \mid \mathbf{v} \in V\}.\end{aligned}$$

We refer to $\mathbf{u}$ and $\mathbf{v}$ above as the component words. Except for the contribution of the type-U and type-V codewords, the weight distribution of $\mathcal{C}$ is as for a random ternary linear $[n, k]$ code, see [13] for precise statements. We partition type-U and type-V codewords according to the Hamming weight of their component words and define

$$\begin{aligned}\mathcal{U}(\mathcal{C}, j) &= \{(\mathbf{a} \star \mathbf{u}, \mathbf{c} \star \mathbf{u}) \mid \mathbf{u} \in U, |\mathbf{u}| = j\},\\ \mathcal{V}(\mathcal{C}, j) &= \{(\mathbf{b} \star \mathbf{v}, \mathbf{d} \star \mathbf{v}) \mid \mathbf{v} \in V, |\mathbf{v}| = j\},\end{aligned}$$

whose respective expected cardinalities are

$$\mathbb{E}\left[|\mathcal{U}(\mathcal{C}, j)|\right] = W_j(U) = \frac{\binom{n/2}{j} 2^j}{3^{n/2 - k_U}} \text{ and } \mathbb{E}\left[|\mathcal{V}(\mathcal{C}, j)|\right] = W_j(V) = \frac{\binom{n/2}{j} 2^j}{3^{n/2 - k_V}}.$$

The words of $\mathcal{U}(\mathcal{C}, j)$ all have a Hamming weight $t = 2j$ while the words of $\mathcal{V}(\mathcal{C}, j)$ have a Hamming weight $j \leq t \leq 2j$ depending on the intersection of the component word support with the supports of $\mathbf{b}$ and $\mathbf{d}$. If $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$ are uniformly distributed with the condition (1), then, on average, three quarters of the b_i and d_i are non zero. So the typical weight of a word of $\mathcal{V}(\mathcal{C}, j)$ is $t = 3j/2$.

Codewords of type-U or type-$V$are very few, less than $3^{k_U} + 3^{k_V}$ among $3^k = 3^{k_U+k_V}$ codewords. However, for a particular weight t, it may happen that most codewords of weight t of type-U or type-V. For typical Wave parameters it may happen, see Figure 1, that, either for small or for large values of t, type-U codewords dominate. We also observe in Figure 1 that the type-V codewords never dominate, regardless of the weight. This also holds for other possible choices of code parameters k_U, k_V for Wave.

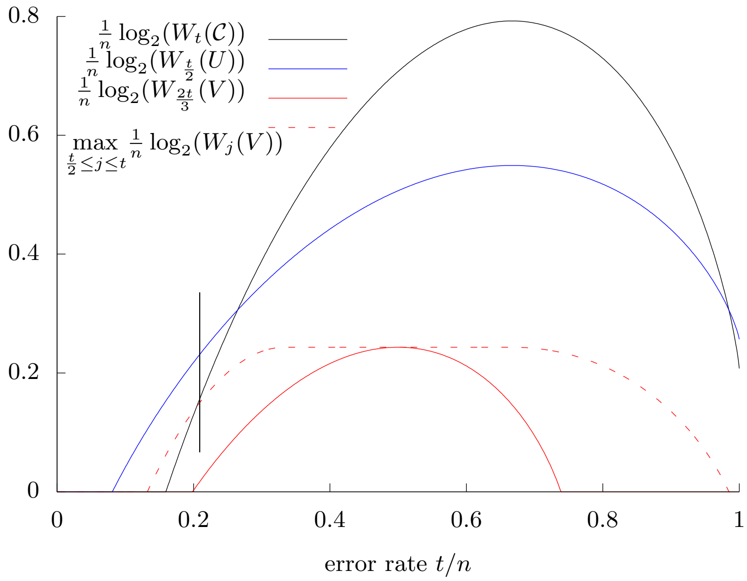

Fig. 1. (Expected) Number of codewords of weight t in $\mathcal{C}$ (black), of type-U (blue), or type-V (red), for Wave parameters $k_U = 0.693 \cdot \dfrac{n}{2}$, $k_V = 0.307 \cdot \dfrac{n}{2}$, $k = \dfrac{n}{2}$ and $w = 0.894 \cdot n$

Example: Lets consider words of weight $t = 0.209 \cdot n$ in a ternary $(U|U + V)$ code $\mathcal{C}$ of length n, dimension $k = 0.5 \cdot n$, and $k_U = 0.693 \cdot \frac{n}{2}$ (the vertical line in Figure 1). We expect to find $2^{0.231 \cdot n}$ words of weight $t/2$ in a random ternary linear $[n/2, k_U]$ code (the blue curve) and “only” $2^{0.156 \cdot n}$ in a random ternary linear $[n, k]$ code (the black curve). From [13], the weight of codewords which are not of type-U or type-V is distributed as for a random code, so, if one can somehow sample a (random) word of weight $t = 0.209 \cdot n$ in $\mathcal{C}$ as above, it will almost certainly be a type-U codeword.

3 q-ary Information Set Decoding (ISD)

Information Set Decoding was first introduced by Prange in 1962 [27] and is one of the main algorithmic techniques to solve the decoding problem. It was later improved in many ways, in particular: birthday paradox [29], representation technique in [23], nearest neighbors in [24]. The current state-of-the-art in the binary case is [5,6,9]. Variants using the Generalized Birthday Algorithm [31] (GBA) are also of interest. GBA was first used for decoding in [11], and for some regime of parameters it can be efficiently used within ISD. This is the case in particular for the decoding of large weights in the non binary case [7].

Few contributions are available in the q-ary case, let us mention [26,8,7]. The interested reader may also refer to [14, Ch. 3] for a comprehensive survey of ISD for any alphabet size. In general the analysis is very similar to the binary case, we briefly revisit that below. Note that, as remarked in [26], there is a factor $\sqrt{q-1}$ to be gained that derives from the linearity of the problem: the identity $\mathbf{e}_0\mathbf{H}_0^{\intercal} = \mathbf{s}_0$ remains true up to a non-zero factor, and this can be used to speed-up the search. We will ignore this in the sequel as we only consider complexity up to a polynomial factor.

3.1 An ISD Framework

We will describe and analyze several ISD variant that fit in the framework described in [18]. This is the case of most known variants, with the notable exception of the latest and best known asymptotic variants [5,6,9]. Those variants have not been generalized yet to the non-binary case, and moreover they are not considered as practical. This could change in the future.

The ISD framework we consider here is described as follows. For a choice of parameters $0 \leq \ell \leq n-k$ and $0 \leq p \leq t$, solve $\mathrm{Dec}(\mathbf{H}_0, \mathbf{s}_0, t)$ by repeating:

1. Pick a permutation σ uniformly in $\mathcal{S}_n$ and compute $(\mathbf{H}, \mathbf{H}', \mathbf{s}, \mathbf{s}') \in \mathbf{F}_q^{\ell \times (k+\ell)} \times \mathbf{F}_q^{(n-k-\ell)\times(k+\ell)} \times \mathbf{F}_q^{\ell} \times \mathbf{F}_q^{n-k-\ell}$ such that for some non singular matrix $\mathbf{U} \in \mathbf{F}_q^{(n-k)\times(n-k)}$,

$$\mathbf{U}\mathbf{H}_0^{\sigma} = \left(\begin{array}{c|c} \mathrm{Id}_{n-k-\ell} & \mathbf{H}' \\ \hline 0 & \mathbf{H} \end{array}\right) = \mathbf{U}\mathbf{s}_0^{\intercal} = \begin{pmatrix} \mathbf{s}'^{\intercal} \\ \mathbf{s}^{\intercal} \end{pmatrix}. \tag{5}$$

2. Use a SubISD routine to compute $\mathcal{E}_{\mathrm{isd}} \subset \mathcal{E} = \{\mathbf{e} \in \mathcal{E}_{q;k+\ell,p}, |\mathbf{e}| = p, \mathbf{e}\mathbf{H}^{\intercal} = \mathbf{s}\}$.
3. For all $\mathbf{e} \in \mathcal{E}_{\mathrm{isd}}$, check $\mathbf{e}_0 = (\mathbf{e}, \mathbf{s}' - \mathbf{e}\mathbf{H}'^{\intercal})$.

Variants of ISD differ in the SubISD routine used at the second step. Every $\mathbf{e}_0$ of weight t at step 3 is a valid solution, the algorithm may either exit there, or continue until a prescribed number of iterations is executed. ISD algorithms are probabilistic and the successive iterations are independent. We will view the sets $\mathcal{E}$ and $\mathcal{E}_{\text{isd}}$ of step 2 as random variables. For convenience, we denote

$$\mathcal{F} = \left\{ (\mathbf{e}, \mathbf{s}' - \mathbf{e}\mathbf{H}'^{\intercal})^{\sigma^{-1}}, \mathbf{e} \in \mathcal{E} \right\} \text{ and } \mathcal{F}_{\text{isd}} = \left\{ (\mathbf{e}, \mathbf{s}' - \mathbf{e}\mathbf{H}'^{\intercal})^{\sigma^{-1}}, \mathbf{e} \in \mathcal{E}_{\text{isd}} \right\} \subset \mathcal{F}.$$

The first set $\mathcal{F}$ above has, a priori, the same size as $\mathcal{E}$ and is the maximal search space for a particular iteration. If $\mathbf{x}$ belongs to $\text{Dec}(\mathbf{H}_0, \mathbf{s}_0, t)$, then the permutation σ is "good for $\mathbf{x}$" if $\mathbf{x} \in \mathcal{F}$. The second set $\mathcal{F}_{\text{isd}}$ is possibly smaller than $\mathcal{F}$ and is the effective search space for a particular iteration and a particular SubISD routine, generally $\mathcal{F}_{\text{isd}}$ is close to $\mathcal{F}$.

Let $\mathcal{A}$ denote a variant of ISD, we define the following three quantities

1. For all $\mathbf{x} \in \mathbf{F}_q^n$ such that $\mathbf{x}\mathbf{H}_0^{\intercal} = \mathbf{s}_0$, we define the probability

$$\pi_{\ell,p}(\mathbf{x}) = \Pr\left(\mathbf{x} \in \mathcal{F}\right) = \frac{\binom{n-k-\ell}{|\mathbf{x}|-p}\binom{k+\ell}{p}}{\binom{n}{|\mathbf{x}|}}, \quad \pi_{\ell,p}(i) = \frac{\binom{n-k-\ell}{i-p}\binom{k+\ell}{p}}{\binom{n}{i}}.$$

   As it only depends of the weight of $\mathbf{x}$ we overload the notation and define $\pi_{\ell,p}(|\mathbf{x}|) = \pi_{\ell,p}(\mathbf{x})$. This quantity is the same for all ISD variants. It is in fact the probability that the permutation σ drawn at step 1 properly splits the support of $\mathbf{x}$ with exactly p non-zero coordinates in the $k + \ell$ rightmost coordinates after permutation.
2. For a given SubISD routine at step 2, we define the *decimation factor* as

$$\mu_{\mathcal{A}} = \frac{\mathbb{E}[|\mathcal{E}_{\text{isd}}|]}{\mathbb{E}[|\mathcal{E}|]} \leq 1.$$

   This is the proportion of solutions of the decoding problem $\text{Dec}(\mathbf{H}, \mathbf{s}, p)$ that are discovered on average by the SubISD routine at step 2. The decimation factor is also equal to $\mathbb{E}[|\mathcal{F}_{\text{isd}}|]/\mathbb{E}[|\mathcal{F}|]$ and this quantity is independent of the target weight t.
3. The average cost of one iteration is denoted $C_{\mathcal{A}}$. It is also independent of the target weight t.

For every particular instance $\mathcal{A}$ of the ISD framework, that is for every choice of (ℓ, p) and of the SubISD routine, the computational cost can be estimated, depending on the task that $\mathcal{A}$ should solve (single or multiple targets, see below). The cost will next be minimized over all choices of (ℓ, p) and possibly other parameters of the SubISD routine.

Single Solution. For any $\mathbf{x} \in \text{Dec}(\mathbf{H}_0, \mathbf{s}_0, t)$, the probability to discover that particular solution in a particular iteration is

$$P = \Pr(\mathbf{x} \in \mathcal{F}_{\text{isd}}) = \mu_{\mathcal{A}} \cdot \Pr(\mathbf{x} \in \mathcal{F}) = \mu_{\mathcal{A}} \cdot \pi_{\ell,p}(t)$$

so we expect to make $1/P$ iterations to find $\mathbf{x}$ and the expected computational effort is

$$\text{WF}_{\mathcal{A}}(n, k, t, 1) = \frac{C_{\mathcal{A}}}{\mu_{\mathcal{A}} \cdot \pi_{\ell,p}(t)} \tag{6}$$

up to a constant factor.

Multiple Solutions. Let $\mathcal{X}$ denote a subset of cardinality N of all solutions of $\mathrm{Dec}(\mathbf{H}_0, \mathbf{s}_0, t)$. The probability to discover an element of $\mathcal{X}$ in a particular iteration is

$$\begin{aligned} P_{\mathcal{A}}(t,N) = \Pr(\mathcal{X} \cap \mathcal{F}_{\text{isd}} \neq \emptyset) &\leq 1 - \prod_{\mathbf{x}\in\mathcal{X}} (1 - \Pr(\mathbf{x} \in \mathcal{F}_{\text{isd}} \neq \emptyset)) \qquad (7) \\ &\leq 1 - (1 - \mu_{\mathcal{A}} \cdot \pi_{\ell,p}(t))^N \\ &\leq \min(1, N \cdot \mu_{\mathcal{A}} \cdot \pi_{\ell,p}(t)) \end{aligned}$$

so the expected computational effort to find an element of $\mathcal{X}$ is

$$\mathrm{WF}_{\mathcal{A}}(n,k,t,N) = \frac{C_{\mathcal{A}}}{P_{\mathcal{A}}(t,N)} \geq \frac{C_{\mathcal{A}}}{\min(1, N \cdot \mu_{\mathcal{A}} \cdot \pi_{\ell,p}(t))}. \qquad (8)$$

The inequality (7) derives from the union bound. The equality holds when the events "$\mathbf{x} \in \mathcal{F}$" are all independent. In some situation, for instance if $\mathcal{X}$ is a set of type-U codewords as in §2.5, the actual probability might be smaller and the actual computational effort larger.

Effective Workfactor. The workfactor formula (6) or (8) depends on the parameters p and ℓ, and possibly other parameters stemming from the subroutine used in the variant. The effective workfactor corresponds to the choice of parameters that will minimize the formula. The corresponding optimization problem can be difficult and has to be solved for each particular problem, that is for given values of n, k, t, and N.

3.2 ISD-MMT

The generalization of the MMT algorithm [23] to the q-ary case is relatively straightforward. We briefly sketch the algorithm and its analysis in the q-ary case, the interested reader may wish to check the relevant literature for a more comprehensive presentation.

We consider the framework of the previous section. After permuting the coordinates and computing a partial Gaussian elimination as in (5), we consider the routine of step 2, which attempts to recover all, or as many as possible, elements of the search space $\mathcal{E} = \left\{\mathbf{e} \in \mathbf{F}_q^{k+\ell}, |\mathbf{e}| = p, \mathbf{e}\mathbf{H}^\intercal = \mathbf{s}\right\}$.

We denote $L = \binom{k+\ell}{p}(q-1)^p$ and $L_0 = \binom{(k+\ell)/2}{p/4}(q-1)^{p/4}$. We call syndrome of $\mathbf{y} \in \mathbf{F}_q^{k+\ell}$ the quantity $\mathbf{y}\mathbf{H}^\intercal$. An additional parameter ℓ_2, $0 \leq \ell_2 \leq \ell$ is introduced. The routine runs as follows:

1. Build 4 lists of size L_0 of words in $\mathbf{F}_q^{k+\ell}$ of weight $p/4$
2. Merge the above lists pairwise to obtain 2 lists of size L_0^2/q^{ℓ_2} of words in $\mathbf{F}_q^{k+\ell}$ of weight $p/2$ with a prescribed value on ℓ_2 coordinates of their syndromes
3. Merge the above 2 lists to obtain a list of size $L_0^4/q^{\ell+\ell_2}$ of words in $\mathbf{F}_q^{k+\ell}$ of weight p with a prescribed value on the other $\ell - \ell_2$ coordinates of their syndromes

The cost of this procedure is, up to a constant factor, the maximum of all the above list sizes. The optimal choice of ℓ_2 is such that $q^{\ell_2} = L/L_0^4$ and L_0 is usually negligible for parameters of interest.

Using the fact that $\binom{k+\ell}{p/2}(q-1)^{p/2} = \Omega(\sqrt{p}) \cdot L_0^2$, the iteration cost becomes

$$C_{\text{MMT}} = \binom{k+\ell}{p}(q-1)^p \cdot \max\left(\frac{1}{\binom{k+\ell}{p/2}(q-1)^{p/2}}, \frac{1}{q^\ell}\right)$$

which is minimal when the two terms in the max are equal. Finally the characteristic quantities of ISD-MMT are

$$C_{\text{MMT}} = \binom{k+\ell}{p}(q-1)^p q^{-\ell} \text{ and } \mu_{\text{MMT}} = 1 \text{ with } q^\ell = \binom{k+\ell}{p/2}(q-1)^{p/2} \qquad (9)$$

up to a polynomial factor.

3.3 ISD-GBA

Wagner's Generalized Birthday Algorithm (GBA) [31] can be used as SubISD routine. Again, we only briefly sketch the algorithm, the interested reader may refer to [25,7] for more details. The GBA of order a builds a binary tree, with 2^a leaves. Lists attached to siblings are merged and attached to the parent node. The list attached to the root is returned.

1. At level a: build 2^a lists of L words of weight $p/2^a$ in $\mathbf{F}_q^{k+\ell}$
2. At level i, $0 \le i < a$: merge pairwise the 2^{i+1} lists of level $i+1$ to obtain 2^i lists of L words of weight $p/2^i$ in $\mathbf{F}_q^{k+\ell}$ with a prescribed value on ℓ/a coordinates of their syndromes
3. At level 0: output the final list of L words of weight p with a prescribed syndrome

With the constraint that $L = q^{\ell/a}$ and $L^{2^a} \le \binom{k+\ell}{p}(q-1)^p$. In practice, a should be an integer ≥ 2, but the algorithm can be smoothed, see [18,7], and we may consider the real value a such that

$$q^{\ell/a} = \left(\binom{k+\ell}{p}(q-1)^p\right)^{\frac{1}{2^a}}. \qquad (10)$$

Even though this value of a does not correspond to an actual GBA tree as above, it provides meaningful bounds. Putting everything together, the characteristic quantities of GBA are the following:

$$C_{\text{GBA}} = (2^{a+1}-1) \cdot q^{\ell/a} \text{ and } \mu_{\text{GBA}} = \frac{q^{\frac{a+1}{a}\ell}}{\binom{k+\ell}{p}(q-1)^p} \qquad (11)$$

up to a polynomial factor.

The DOOM Variant. The DOOM problem, Problem 2, is relevant in the case of the security against forgery of a signature scheme, as the adversary can build any number of messages and be happy to sign only one of them.

The best known way to exploit multiple instance with GBA is to fill one of the 2^a lists of level a with target syndromes. The characteristic quantities of GBA-DOOM verify (11), but instead of (10) the optimal order a will be such that

$$q^{\ell/a} = \left(\binom{k+\ell}{p}(q-1)^p\right)^{\frac{1}{2^a-1}}. \qquad (12)$$

4 Best Known Attacks on Wave

We consider an instance of Wave of main parameters (n, w, k), public key $\mathbf{H}_{\text{pub}} \in \mathbf{F}_3^{(n-k)\times n}$, and secondary parameters (k_U, k_V, g). The public code is $\mathcal{C}_{\text{pub}} = \mathcal{C}^\pi = \langle \mathbf{H}_{\text{pub}} \rangle^\perp$ where π is a secret permutation and $\mathcal{C}$ is a generalized $(U|U+V)$ code. The type-U and type-V codewords of $\mathcal{C}$, denoted $\mathcal{U}(\mathcal{C})$ and $\mathcal{V}(\mathcal{C})$, are defined in §2.5. To ease the statements we will also call type-U and type-V codewords the elements of $\mathcal{C}_{\text{pub}}$ which belong to $\mathcal{U}(\mathcal{C})^\pi$ and $\mathcal{V}(\mathcal{C})^\pi$ respectively.

- **Forgery Attack**. Without knowledge about the secret, the adversary is reduced to solve $\text{DP}_\infty(3; n, k, w)$ on average. The target weight w is large, close to the block length n. In this regime, the best known attack is Information Set Decoding using the Generalized Birthday Algorithm as subroutine, see [7].
- **Key Attack**. The best known method for recovering the secret key, or at least distinguishing the public key from a random matrix, will consist in exhibiting a type-U codeword in $\mathcal{C}_{\text{pub}}$ or in $\mathcal{C}_{\text{pub}}^\perp$.

4.1 Forgery Attack

This problem was addressed in [7]. We want to solve the unlimited DOOM problem $\text{DP}_\infty(3; n, k, w)$ when w is close to n. The workfactor is (larger than)

$$\frac{C_{\text{GBA}}}{\min(1, N \cdot \mu_{\text{GBA}} \cdot \pi_{\ell,p}(w))}$$

with $N = \binom{n}{w} 2^w / 3^{n-k}$ the expected number of solutions, C_{GBA} and μ_{GBA} defined in (11), minimized over all choices of ℓ, p, a such that (12).

Optimization. It is optimal to choose $p = k + \ell$ in this regime, this was already remarked in [7]. We also notice that the workfactor is minimal when the success probability of an iteration is constant (*i.e.* one or a small number of iterations is enough). This happens, asymptotically, when $N \cdot \mu_{\text{GBA}} \cdot \pi_{\ell,p}(t) = 1$. Finally, the minimal workfactor is $\text{WF}_{\text{GBA}} = q^{\ell/a}$ when ℓ and a verify

$$3^{\ell/a} = 2^{(k+\ell)/(2^a-1)} = \frac{3^{n-k-\ell}}{\binom{n-k-\ell}{w-k-\ell} 2^{w-k-\ell}}.$$

4.2 Key Attack

The key attack we consider here consist in searching a type-U or a type-V codeword in either the public $[n, k]$ code $\mathcal{C}_{\text{pub}}$ or its dual. This is done by applying a generic decoding technique (*e.g.* ISD and variants) for a target error weight t and the zero syndrome.

Limiting the Codeword Search:

- We will not consider type-V codewords. In fact, for all Wave parameters of interest, the situation is similar to what we observe in Figure 1, the type-V codewords (red curve) are always dominated by the type-U codewords. So it will always be easier to find a type-U.
- We will not consider type-U codewords of high weight. We observe in Figure 1 that, for large weights, the blue curve of type-U codewords is above the expected number of codewords of that weight, in black. This phenomenon even amplifies when the code rate k decrease. Even though, those words are always harder to find in practice.

Note that to be thorough, the designer must check the above cases a posteriori, which can easily be done when the parameters are known. The purpose of the above limitation is to reduce the parameters selection to a trade-off between the forgery attack of §4.1 and the search of type-U codewords of small weight.

Finding type-U Codewords of Small Weight. This search will consist in finding a word of weight t, to be determined, in the code $\mathcal{C}_{\text{pub}}$ or $\mathcal{C}_{\text{pub}}^{\perp}$. The adversary wins if it was able to obtain at least one type-U codeword of such weight. That is:

- either search a type-U codeword in $\mathcal{C}_{\text{pub}}$, one among $N_U(t) = \binom{n/2}{t/2} 2^{\frac{t}{2}} / 3^{\frac{n}{2} - k_U}$;
- or search a type-U codeword in $\mathcal{C}_{\text{pub}}^{\perp}$, one among $N_V(t) = \binom{n/2}{t/2} 2^{\frac{t}{2}} / 3^{k_V}$.

The workfactor is minimized over those two cases and over all possible even values of t. The corresponding computational effort when the generic decoding algorithm is ISD-MMT [23] is

$$\min\left(\min_{t,\ell}\left(\mathrm{WF}_{\mathrm{MMT}}(n, k, t, N_U(t))\right), \min_{t,\ell}\left(\mathrm{WF}_{\mathrm{MMT}}(n, n-k, t, N_V(t))\right)\right) \quad (13)$$

whose value derives from (8) and (9). The optimization parameters p and ℓ are related when the optimum is reached in ISD-MMT, see (9), here we write p as a function of ℓ, and ℓ is used to minimize the expression.

Remark: the optimal value of t, that is the weight for which it is easiest to find a type-U codeword, is a non trivial trade-off. For instance for $(k/n, w/n) = (0.5, 0.894)$ and $k_U = 0.693\, n/2$, the easiest target weight is not, as one could expect, the Gilbert-Varshamov distance ($t/n = 0.081$ where the blue curve meets the horizontal axis in Figure 1) but is much larger ($t/n = 0.209$ shown by the small vertical black line in Figure 1).

4.3 Wave Parameter Selection

We consider here asymptotic estimates. Relative parameters are considered, *e.g.* k/n, $w/n\dots$, and polynomial factors are not considered. Our purpose is to find the good proportion between the scheme parameters, main and secondary. As the security scales linearly with code-length n, the target security is obtained as a final step by scaling n.

1. Let λ be the target (classical) security or the "number of security bits", *e.g.* $\lambda = 128$.
   We want that all attacks against the scheme require a computational effort at least equal to 2^{λ}.
2. Select the main parameters (k, w).
   (guidelines: $0.35 \leq k/n \leq 0.7$ and $0.85 \leq w/n \leq 0.95$)
   The forgery attack requires a computational effort
   $$\mathrm{WF}_{\mathrm{GBA}}(n, k, w) \geq 2^{c \cdot n}$$
   where c only depends on k/n and w/n.
   We may now relate the code-length and the security as we want to reach $c \cdot n = \lambda$ where λ is the "number of (classical) security bits".
3. The secondary parameters (k_U, k_V) derive from the main ones:
   $$k_U = g + \frac{3}{2}w - n \text{ and } k_V = k - k_U.$$
4. The gap g is used in [15] to unconditionally ensure a uniformity condition needed to formally reach the EUF-CMA security, its value is $g = \lambda / \log_2 3 = 0.631 c \cdot n$ for "λ bits of (classical) security". Using $g = 0$ is probably safe but, at the moment, would require an additional heuristic assumption. We will favor an intermediate value $g = 0.315\, c \cdot n$ and admit that it corresponds to an adversary limited to $2^{\lambda/2}$ queries to a signing oracle, *e.g.* a scheme with 128 bits of security with an adversary allowed to at most 2^{64} queries to a signing oracle, corresponding to a gap $g \approx 40$.
5. The corresponding computational effort for the key attack derives from (13) and §4.2. If we denote $\mathrm{WF}_{\mathrm{key}}$ this quantity, the corresponding asymptotic exponent is $c' = \frac{1}{n} \log_2(\mathrm{WF}_{\mathrm{key}})$ which only depends of the relative value k/n and k_U/n. Finally k_U is related to w and to the gap g.

Example: In Figure 2, we consider $k/n = 0.5$. The asymptotic exponent is given for forgery attack, the increasing curve in black, and for the key attack, the decreasing curves in dotted red, solid blue, dashed green, corresponding to various choices for the gap. The optimal choice will correspond to the curves intersections:

asymptotic parameters			Wave parameters, $\lambda = 128$					
	w/n	c	n	k	w	k_U	k_V	g
no gap	0.89665	0.0162184	7904	3952	7077	2763	1183	0
half gap	0.89412	0.0150016	8532	4266	7629	2951	1315	40
full gap	0.89193	0.0139887	9150	4575	8161	3132	1443	81

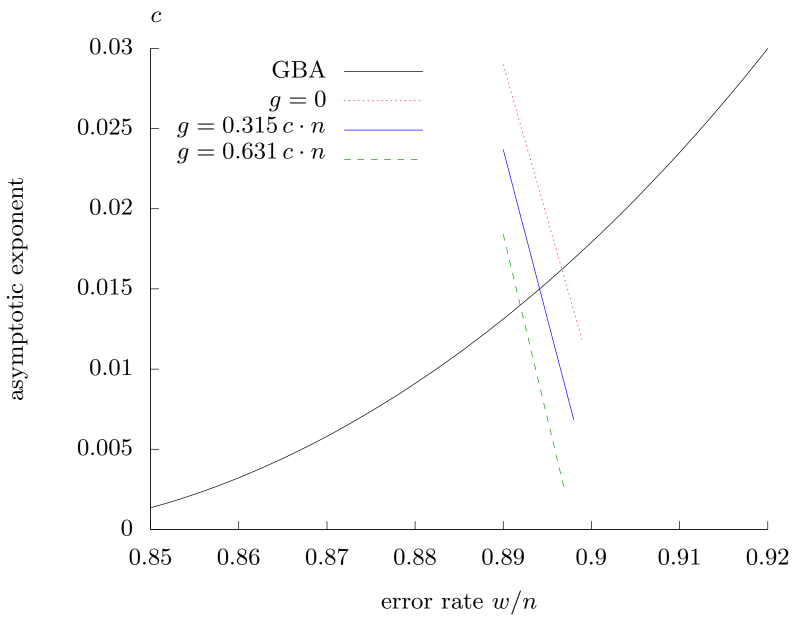

Fig. 2. Security Exponent for Best Known Forgery Attacks (black) and Key Attacks (blue) *vs.* Error Weight for a Code Rate $k = 0.5$

We observe that, even though the gap does not change the order of magnitude, it has a significant impact on the system parameters. In the sequel we will consider the "half gap" scenario, corresponding to $g = 40$ when the target security is $\lambda = 128$.

4.4 Sizes

The public key can be provided in systematic form, so its size is equal to $k(n-k)\log_2 3$ bits. The signature is a ternary vector $\mathbf{e}$ of length n and weight w. As explained in [2, §2.5], when the public key is in systematic form, it is enough to provide the last k coordinates of $\mathbf{e}$. Moreover those coordinates can be compressed to a size $k \cdot h_3(w/n)$ bits, where $h_3(x) = -x\log_2 x - (1-x)\log_2(1-x) + x$ is the ternary entropy function.

For a security level $\lambda = 128$ and the intermediate gap $g = 40$ we give in Table 1 Wave parameters for a code rate between 0.38 and 0.70. We also give in Figure 3 the signature length and the public key size for the same range of code rates. We observe that if k/n decrease below 0.43 or increases above 0.66 then both the signature length and the key size increase. So the effective range for the code rate is rather narrow. In practice $k = 0.5$ seems to offer a rather good trade-off, the signature length is very close the minimum, 737 bytes for a minimum of 723 bytes, while the public key has a size of 3.6MB, not much larger than the minimal 3.04MB.

Table 1. Wave parameters for $(\lambda, g) = 128, 40$

k/n	n	k	w	k_U	k_V	signature length (bytes)	key size (MB)
0.38	10464	3976	8918	2953	1023	724	5.11
0.41	9792	4015	8454	2928	1087	723	4.60
0.44	9262	4075	8095	2920	1155	724	4.19
0.47	8846	4158	7822	2927	1231	729	3.86
0.50	8526	4263	7623	2949	1314	737	3.60
0.53	8318	4409	7516	2995	1414	751	3.41
0.56	8184	4583	7469	3059	1524	768	3.27
0.59	8124	4793	7485	3143	1650	791	3.16
0.62	8136	5044	7564	3249	1795	818	3.09
0.65	8226	5347	7713	3383	1964	852	3.05
0.68	8402	5713	7942	3550	2163	894	3.04
0.70	8574	6002	8146	3685	2317	928	3.06

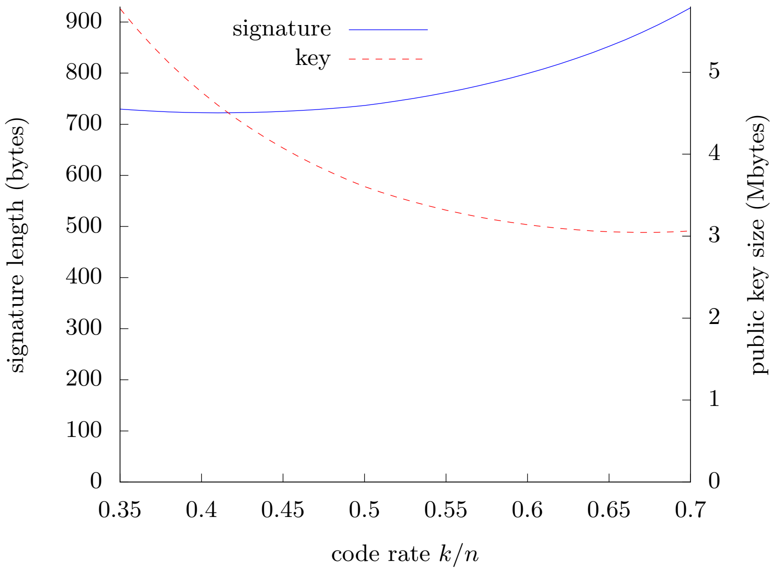

Fig. 3. Signature Length and Public Key Size *vs.* Code Rate for a Security $\lambda = 128$ and a Gap $g = 40$

4.5 Scaling Security

All parameters scale linearly with the security. The signature length also scales linearly, the key size however is quadratic. So Table 1 can be used to deduce parameters offering a higher security level, *e.g.* $\lambda \in \{192, 256\}$. This scaling will also increase the gap while in some situations, for instance in the NIST call for postquantum primitives, some capabilities of the adversary are limited regardless of the target security. Let us assume then that the adversary is limited to 2^{64} calls to a signing oracle and we want to reach security levels 192 or 256. It is possible to do slightly better than just scaling Table 1

Table 2. Scaling Wave Parameters with 2^{64} Queries ($g = 40$) to a Signing Oracle

λ	g	n	k	w	k_U	k_V	signature length	key size
128	40	8526	4263	7623	2949	1314	737 bytes	3.60 MB
192	40	12476	6238	11165	4311	1927	1076 bytes	7.71 MB
256	40	16424	8212	14705	5673	2539	1416 bytes	13.36 MB

4.6 Quantum Security

All attacks that are involved in the parameter selection process derive from ISD. So far quantum speedup of ISD variants [3,22,10] never managed to reduce the security exponent by more than a factor 2 and thus, from §4.5, scaling the scheme to resist to quantum cryptanalysis will be straightforward.

5 Conclusion

Wave is the first scalable hash-and-sign signature scheme based on codes. It enjoys short signature and fast verification, whereas signing time is average, hundreds of milliseconds, and the public key large, several megabytes.

We propose here some practical parameters for the scheme, based on the current knowledge of the underlying hard problems. Our proposed methodology would easily adapt to any evolution of the best known solvers for those problems.

Note that all known attacks eventually require the use of solvers for the generic decoding problem, and even though those solvers are used in unusual regime of parameters, the generic decoding problem has been studied for more than half a century without any significant improvement. For instance, decoding at the Gilbert-Varshamov distance at rate 0.5, *i.e.* $t = 0.11 \cdot n$ errors in a binary code of length n and dimension $n/2$, features a security exponent of 0.12 with Prange algorithm [27] which only reduces to about 0.09 with today's best known solver [5]. In other words, after sixty years, any code-based scheme corresponding to this regime of parameters, as for the instance the Stern authentication scheme [30] and related signatures, as [16] for instance, need an increase of only 25%

of its block size to maintain the same security. Very few problems in public key cryptography can claim such a stability for the computational cost of their best known attacks.

References

1. Carlos Aguilar, Philippe Gaborit, and Julien Schrek. A new zero-knowledge code based identification scheme with reduced communication. In *Proc. IEEE Inf. Theory Workshop- ITW 2011*, pages 648–652. IEEE, October 2011.
2. Gustavo Banegas, Thomas Debris-Alazard, Milena Nedeljkovi, and Benjamin Smith. Wavelet: Code-based postquantum signatures with fast verification on microcontrollers. Cryptology ePrint Archive, Report 2021/1432, 2021. `https://ia.cr/2021/1432`.
3. Daniel J. Bernstein. Grover vs. McEliece. In Nicolas Sendrier, editor, *Post-Quantum Cryptography 2010*, volume 6061 of *LNCS*, pages 73–80. Springer, 2010.
4. Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is more: Code-based signatures without syndromes. In Abderrahmane Nitaj and Amr M. Youssef, editors, *Progress in Cryptology - AFRICACRYPT 2020*, volume 12174 of *LNCS*, pages 45–65. Springer, 2020.
5. Leif Both and Alexander May. Optimizing BJMM with Nearest Neighbors: Full Decoding in $2^{2/21n}$ and McEliece Security. In *WCC Workshop on Coding and Cryptography*, September 2017.
6. Leif Both and Alexander May. Decoding linear codes with high error rate and its impact for LPN security. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography 2018*, volume 10786 of *LNCS*, pages 25–46, Fort Lauderdale, FL, USA, April 2018. Springer.
7. Rémi Bricout, André Chailloux, Thomas Debris-Alazard, and Matthieu Lequesne. Ternary syndrome decoding with large weights. In Kenneth G. Paterson and Douglas Stebila, editors, *Selected Areas in Cryptography - SAC 2019 - 26th International Conference, Waterloo, ON, Canada, August 12-16, 2019, Revised Selected Papers*, volume 11959 of *Lecture Notes in Computer Science*, pages 437–466. Springer, 2019.
8. Rodolfo Canto Torres. Asymptotic analysis of ISD algorithms for the $q-$ary case. In *Proceedings of the Tenth International Workshop on Coding and Cryptography WCC 2017*, September 2017.
9. Kevin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. Statistical decoding 2.0: Reducing decoding to LPN. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022*, volume 13794 of *LNCS*, pages 477–507. Springer, 2022.
10. André Chailloux, Thomas Debris-Alazard, and Simona Etinski. Classical and quantum algorithms for generic syndrome decoding problems and applications to the lee metric. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20-22, 2021, Proceedings*, volume 12841 of *Lecture Notes in Computer Science*, pages 44–62. Springer, 2021.
11. Jean-Sebastien Coron and Antoine Joux. Cryptanalysis of a provably secure cryptographic hash function. IACR Cryptology ePrint Archive, Report 2004/013, 2004. `http://eprint.iacr.org/`.

12. Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 157–174, Gold Coast, Australia, 2001. Springer.
13. Thomas Debris-Alazard. *Cryptographie fondée sur les codes : nouvelles approches pour constructions et preuves ; contribution en cryptanalyse.* Theses, Sorbonne Université, December 2019.
14. Thomas Debris-Alazard. Code-based cryptography: Lecture notes. arXiv:2304.03541, April 2023.
15. Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 21–51, Kobe, Japan, December 2019. Springer.
16. Thibauld Feneuil, Antoine Joux, and Matthieu Rivain. Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2023*, volume 13508 of *LNCS*, pages 541–572. Springer, 2022.
17. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A.M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86*, volume 263 of *LNCS*, pages 186–194. Springer, 1987.
18. Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In M. Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 88–105. Springer, 2009.
19. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.
20. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *STOC '07*, pages 21–30. ACM, 2007.
21. Thomas Johansson and Fredrik Jönsson. On the complexity of some cryptographic problems based on the general decoding problem. *IEEE Trans. Inform. Theory*, 48(10):2669–2678, October 2002.
22. Ghazal Kachigar and Jean-Pierre Tillich. Quantum information set decoding algorithms. In *Post-Quantum Cryptography 2017*, volume 10346 of *LNCS*, pages 69–89, Utrecht, The Netherlands, June 2017. Springer.
23. Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $O(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 107–124. Springer, 2011.
24. Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 203–228. Springer, 2015.
25. L. Minder and A. Sinclair. The extended k-tree algorithm. In C. Mathieu, editor, *Proceedings of SODA 2009*, pages 586–595. SIAM, 2009.
26. Christiane Peters. Information-set decoding for linear codes over $\mathbf{F}_q$. In *Post-Quantum Cryptography 2010*, volume 6061 of *LNCS*, pages 81–94. Springer, 2010.
27. Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

28. Nicolas Sendrier. Decoding one out of many. In *Post-Quantum Cryptography 2011*, volume 7071 of *LNCS*, pages 51–67, 2011.

29. Jacques Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *LNCS*, pages 106–113. Springer, 1988.

30. Jacques Stern. A new identification scheme based on syndrome decoding. In D.R. Stinson, editor, *Advances in Cryptology - CRYPTO'93*, volume 773 of *LNCS*, pages 13–21. Springer, 1993.

31. David Wagner. A generalized birthday problem. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, 2002.