Quantum security analysis of Wave

Johanna Loyer

Inria, Saclay, France

Abstract. Wave is a code-based digital signature scheme. Its hardness relies on the unforgeability of the signature and the indistinguishability of its public key, a parity check matrix of a ternary $(U, U+V)$-code. The best known attacks involve solving the Decoding Problem using the Information Set Decoding algorithm (ISD) to defeat these two problems. Our main contribution is the description of a quantum smoothed Wagner's algorithm within the ISD framework, which improves the forgery attack on Wave in the quantum setting. We also recap the best known key and forgery attacks against Wave in both classical and quantum settings. For each one, we express their time complexity as function of Wave parameters and deduce its claimed security.

Keywords. Decoding problem, Code-based cryptography, Information Set Decoding, Quantum cryptanalysis.

1 Introduction

Wave [BCC+23] is a hash-and-sign digital signature scheme introduced in [DAST19]. It instantiates the theoretical framework of [GPV08], replacing the traditional lattice-based trapdoor, used in schemes like Falcon [FHK+18], with a trapdoor based on coding theory.

Wave offers several advantages: notably short signatures (less than 1 kilobyte for 128-bit security) and fast verification (under one millisecond) [BDANS21]. However, these benefits come at the cost of a large public key (4 megabytes). The scheme was submitted as a candidate in the NIST post-quantum cryptography standardization process.

On the security of this scheme, Wave is provably EUF-CMA secure (existentially unforgeable under chosen-message attacks) under code-based hardness assumptions. Specifically, it relies on the hardness of decoding and the indistinguishability of permuted generalized ternary $(U, U+V)$-codes. The best known attack strategies target these problems via the Decoding Problem (DP). This problem is defined as follows: given a parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$, and a weight w, the goal is to find an $\boldsymbol{e} \in \mathbb{F}_q^n$ of Hamming weight w such that $\boldsymbol{e}\mathbf{H}^\top = \boldsymbol{s}$.

The DP problem admits worst-case to average-case reductions [BMvT78] and is believed to be hard against both classical and quantum attacks, even after more than forty years of strong interest from the research community [Pra62, Ste88, Dum91, FS09, Ber10, BLP11, MMT11, BJMM12, MO15, BM17, KT17, BM18, Kir18, BBSS20a, CDMT22]. It is the foundation of many code-based cryptographic systems, including Classic McEliece [McE78], Alekhnovich's scheme [Ale03], and several NIST candidates such as BIKE [ABB+21] and HQC [MAB+].

However, most prior work has focused on binary fields ($\mathbb{F}_2$). The Decoding Problem over $\mathbb{F}_q$ for $q > 2$, especially in the context of Wave where $q = 3$, has only recently received attention [BCDAL20, CDAE21, KL22, Sen23]. Unlike in $\mathbb{F}_2$, where the decoding hardness symmetrically peaks at mid-low and mid-high weights, the hardness in $\mathbb{F}_3$ is maximal at

E-mail: firstname.lastname@inria.fr (Johanna Loyer)

large weights. This highest computational hardness can be used to claim cryptographic security.

Attacks on Wave fall into two main categories:

- *Key Attacks* aim to solve the Distinguishing Wave Keys (DWK) problem, that asks to distinguish the public key, a permutated generalized ternary $(U, U+V)$-code from the a uniformly random matrix. The best known classical key attack [Sen23] searches for a codeword with a specific target weight, exploiting the fact that such words are easier to find in a $(U, U+V)$-code than in a random code. If this algorithm successfully exhibits such a word in less time than expected for a random code, it identifies the code as a $(U, U+V)$-code; otherwise, it concludes it is random. Consequently, this provides a distinguisher that tackles the DWK problem. This attack utilizes the Information Set Decoding (ISD) framework algorithm with Dumer's algorithm [Dum91] as a subroutine. Prior to this work, there was no quantum cryptanalysis specifically targeting the DWK problem.

- *Message Attacks*, which target the Decoding One Out of Many (DOOM) problem [Sen11]: given a set of syndromes S, the goal is to forge a signature $(\boldsymbol{e}, \boldsymbol{s}) \in \mathbb{F}_3^n \times \mathbb{F}_3^{n-k}$ such that $\boldsymbol{e}\mathbf{H}^\top = \boldsymbol{s} \in S$ and $\boldsymbol{e}$ has Hamming weight w. The best known classical attack [BCDAL20] uses ISD with Wagner's algorithm [Wag02] as a subroutine. The best known quantum attack [CDAE21] applies a quantum variant of Wagner's algorithm within a quantum ISD.

[Sen23] described a method to select optimal Wave parameters by balancing the complexities of these two attacks. Because they impose opposite constraints, parameters are chosen so that both classical attacks are of highest complexity while closely approaching the desired security threshold.

As a post-quantum candidate, it is important to estimate Wave's quantum security. The notion of 'measure' of security depends on the chosen model. Consider an algorithmic attack that, for a input of dimension n, runs in time $A(n) \cdot 2^{\alpha n}$, where $A(n) = \text{poly}(n) >> 1$ corresponds to the polynomial overhead. One model defines the number of security bits simply as α, ignoring the polynomial factor. This is standard and allows a conservative claim on the security that does not depend on the small improvements over these operations. Alternatively, the NIST standard adopts a comparative model: it assesses the cost of breaking a cryptographic scheme relative to that of breaking AES-λ, for $\lambda \in 128, 192, 256$. These define the three security levels (I, III, V). Though this model enables a finer security estimate, it is less conservative.

Contributions.

- We describe a quantum smoothed version of Wagner's algorithm based on the combined approaches of [Sen11], [BCDAL20] and [CDAE21]. Our new algorithm provides an improved message attack on Wave.

- For each of the four best known attacks on Wave, we perform a time complexity analysis and provide transparent expressions as function of the Wave parameters. So the claimed security level can easily be updated with new sets of parameters using our formulas. Table 1 summarizes Wave security bits against all the attacks studied in this work, using the parameter selection from [BCC+23].

- We complete the analysis to incorporate the polynomial overheads of the attacks to the number of security bits, summarized in Table 2. This allows to check whether the selected parameters of Wave [BCC+23] pass the NIST security levels for large-depth quantum computers.

Table 1: Number of security bits of Wave instances. α security bits indicate that the most efficient known attacks require at least time $A(n) \cdot 2^{\alpha n}$ to execute, with $A(n)$ some polynomial overhead.

	Classical		Quantum	
Setting	Key attack Thm. 6	Message attack Thm. 8	Key attack Thm. 7	Message attack Thm. 9
(I)	138	129	82	78
(III)	206	194	123	117
(V)	274	258	164	156

Table 2: Number of security bits of Wave instances, including polynomial overheads (Computed in Section 5). β security bits in this table means that the corresponding attack runs in time 2^{β}.

	Classical		Quantum	
Setting	Key attack	Message attack	Key attack	Message attack
(I)	158	154	109	102
(III)	227	220	151	142
(V)	297	285	193	182

The results in Table 1 reveal a significant gap between the minimum security thresholds and the time of key-distinguishing attacks. This is a consequence of the discreteness of the Wave integer parameters, which makes it impossible to tune the parameters to match the exact minimum number of security bits simultaneously for both types of attacks.

For the message attacks described in Section 4, our analysis of the classical algorithm shows that it achieves a time complexity close to the theoretical lower bound from [FS09]. Regarding the quantum case, we correct the claimed security level given in [BCC$^+$23] for quantum message attacks, finding a slight difference of one security bit with their estimation.

We also observe that key-distinguishing attacks benefit more from the quantum setting than message attacks. This is because Grover's algorithm has a stronger impact when searching in a large range, as in Dumer's algorithm, rather than in several fragmented small ones, as occurs in Wagner's algorithm for message attacks.

Under the standard asymptotic security model, quantum attacks do not appear to be a limiting factor for Wave: even the best quantum attacks do not achieve a full quadratic speedup, and there remains a significant safety margin above the minimum security thresholds. In the NIST model, the Wave parameter selection of levels (I) and (III) satisfy the corresponding thresholds in scenarios where the adversary is assumed to have access to high-depth quantum circuits. However instance for level (V) seems not to meet the required threshold.

Therefore, while classical attacks remain the primary concern for parameter selection under the asymptotic model, this may not be sufficient to meet the official NIST security criteria, especially for higher security level and varying constraints on maximum allowed quantum circuit depth.

Organization of the paper. We recall in Section 2 preliminaries about quantum computing, code-based problems particularly in the case of Wave, the ISD framework, and list merging. Section 3 presents key-distinguishing attacks and Section 4 presents message attacks. Section 5 discusses which NIST security levels Wave instances reach.

Scripts. All our numerical optimizations have been obtained using Python scripts available with the submission.

2 Preliminaries

Notations.

$\mathbb{F}_q$ denotes a q-ary finite field. In this paper, we consider the case $q = 3$. Vectors are in row notation, usually written in bold and their coordinates are in plain, with $\boldsymbol{x} = (x_i)_i$. The weight considered in this work is the Hamming weight denoted $|\boldsymbol{x}| := |\{i : x_i \neq 0\}|$. For a vector $\boldsymbol{x} = (x_0, \dots, x_{n-1})$, we denote by $\boldsymbol{x}_{|[i,j]}$ the restricted vector $(x_i, \dots, x_j)$. For a matrix $\mathbf{H}$, we denote by $\mathbf{H}^\top$ its transpose. For $\boldsymbol{x} = (x_0, \dots, x_{n-1}) \in \mathbb{F}_q^n$ and $\mathbf{M} = (M_{i,j})_{0\leq i<r, 0\leq j<n} \in \mathbb{F}_q^{r\times n}$, we define $\boldsymbol{x}\star\mathbf{M} := (x_j M_{i,j})_{0\leq i<r,0\leq j<n}$ the row-wise star product. We use the usual Landau notation $\mathcal{O}$, and we write $g(n) \in \widetilde{\mathcal{O}}(f(n))$ iff. there exists a constant $k \geq 0$ such that $g(n) \in \mathcal{O}(f(n) \cdot \log^k(n))$.

2.1 Quantum information

The quantum part of this work stands in the quantum circuit model with the assumption that a Quantum Random Access to Classical Memory (QRACM) operation can be efficiently implemented. If we consider registers $x_1, ..., x_n \in \{0,1\}^n$ classically stored, a QRACM operation consists in applying the unitary $|i\rangle |y\rangle \to |i\rangle |x_i \oplus y\rangle$. If a list L is classically stored and quantumly accessible then there exists an efficient quantum circuit that constructs the quantum superposition of its elements $|\psi_L\rangle := \frac{1}{\sqrt{|L|}} \sum_{x\in L} \left|\text{ind}_L(x)\right\rangle |x\rangle$, where $\text{ind}_L(x)$ denotes the index of the element x in the list L. We point out [Kup13] to the reader for more information about the different types of memories in the QRAM model.

The two following theorems are at the foundation of adapting attack algorithms to the quantum setting.

Theorem 1 (Grover's algorithm [Gro96]). *Consider a function $f : L \to \{0,1\}$ that runs in time T. Given quantum access to the list L classically stored, Grover's algorithm returns $x \in L$ such that $f(x) = 1$ in time $\mathcal{O}(T \cdot \sqrt{|L|})$.*

Theorem 2 (Amplitude amplification [BHMT02]). *Let $\mathcal{A}$ be an algorithm without measurements that finds a solution $x \in L$ such that $f(x) = 1$ with a success probability p in time T. Quantum amplitude amplification returns a solution with probability 1 in time $\mathcal{O}(T \cdot 1/\sqrt{p})$.*

2.2 Code-based cryptography

Generic codes.

Definition 1 ($[n,k]_q$-code)**.** A linear code C of length n and dimension k over $\mathbb{F}_q$ is a k-dimensional subspace of $\mathbb{F}_q^n$. The elements of C are called *codewords*. The *rate* of C is defined as k/n. A generator matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ of C verifies $C = \{\mathbf{xG} \mid \mathbf{x} \in \mathbb{F}_q^k\}$ and a parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ of C verifies $C = \{\mathbf{y} \in \mathbb{F}_q^n \mid \mathbf{yH}^\top = \mathbf{0}\}$. For any $\mathbf{y} \in \mathbb{F}_q^n$, the vector $\mathbf{yH}^\top$ is called the syndrome of $\mathbf{y}$ (relatively to $\mathbf{H}$). The dual code of C is $C^\perp = \{\mathbf{xH} \mid \mathbf{x} \in \mathbb{F}_q^{n-k}\}$.

Problem 1 (Decoding Problem - $\mathsf{DP}_{\mathbf{H},\mathbf{s},w}$)**.** *Given a parity check matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$ and a target weight $w \in [|0,n|]$, find a vector $\boldsymbol{e} \in \mathbb{F}_q^n$ such that $|\boldsymbol{e}| = w$ and $\boldsymbol{eH}^\top = \boldsymbol{s}$.*

The problem $\mathsf{DP}_{\mathbf{H},\mathbf{s},w}$ is believed to be hard on average for $\mathbf{H}$ distributed uniformly at random in $\mathbb{F}_q^{(n-k)\times n}$ and $\boldsymbol{s} = \boldsymbol{e}\mathbf{H}^\top$ with $\boldsymbol{e}$ a uniformly random vector in $\mathbb{F}_q^n$ of weight $|\boldsymbol{e}| = w$.

The best known algorithms to solve DP have a polynomial complexity when $\frac{q-1}{q}(n-k) \leq w < k + \frac{q-1}{q}(n-k)$, and exponential otherwise. Notice that to find a codeword with a given target weight, one can solve an instance of $\mathsf{DP}_{\mathbf{H},\mathbf{s}=\mathbf{0},w}$. Setting the syndrome $\mathbf{s}$ at 0 does not affect the hardness of the instance of DP in comparison with an arbitrary $\mathbf{s}$.

Proposition 1. *[DA23] For a uniformly random matrix $\boldsymbol{H} \in \mathbb{F}_3^{(n-k)\times n}$, we expect the solutions to the $\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}$ problem to be on average $\binom{n}{w}\frac{2^w}{3^{n-k}}$.*

Remark. There does not necessarily exist a solution to generic instances of the DP problem. However, Wave parameters ensure that there always exists at least one solution.

Wave.

Definition 2 (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 $0 < k_V < n/2$. For i from 0 to $n/2$, let $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{d}$ denote vectors in $\mathbb{F}_3^{n/2}$ such that $\forall i \in [0, n/2]$, $a_i c_i \neq 0$ and $a_i d_i - b_i c_i \neq 0$.

The ternary linear codes U (resp. V) are of length $n/2$ and dimension k_U (resp. k_V) and admits generator matrix $\mathbf{G}_U$ and parity check matrix $\mathbf{H}_U \in \mathbb{F}_3^{(n/2-k_U)\times n/2}$ (resp. $\mathbf{G}_V$ and $\mathbf{H}_V \in \mathbb{F}_3^{(n/2-k_V)\times n/2}$). Then, the generalized ternary $(U, U+V)$-code C associated to $(\mathbf{H}_U, \mathbf{H}_V, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$ has the following parity check matrix

$$\mathbf{H} = \begin{pmatrix} \mathbf{d} \star \mathbf{H}_U & -\mathbf{b} \star \mathbf{H}_U \\ -\mathbf{c} \star \mathbf{H}_V & \mathbf{a} \star \mathbf{H}_V \end{pmatrix}.$$

The dual of C is a $(U, U+V)$-code associated to $(\mathbf{G}_U, \mathbf{G}_V, -\mathbf{c}, \mathbf{d}, \mathbf{a}, -\mathbf{b})$.

The Wave signature scheme [DAST19] uses a permuted generalized ternary $(U, U+V)$-code of length n and dimension k, whose parity check matrix $\mathbf{H} \in \mathbb{F}_3^{(n-k)\times n}$ constitutes the public key. A weight w is also fixed, as well as the dimensions k_U for code U and k_V for V. The signature of a message m in Wave is an $\boldsymbol{e} \in \mathbb{F}_3^n$ such that $|\boldsymbol{e}| = w$ and $\boldsymbol{e}\mathbf{H}^\top = h(m) \in \mathbb{F}_3^{(n-k)}$, where h is a hash function. A signer with their secret key U, V can use them to efficiently compute such an $\boldsymbol{e}$ to sign their message m.

Table 3 recalls Wave parameters of the three instances submitted to the NIST in [BCC+23].

Table 3: Sets of Wave parameters.

	n	k	w	k_U
(I)	8576	4288	7668	2966
(III)	12544	6272	11226	4335
(V)	16512	8256	14784	5704

From an attacker's perspective, certain special codewords in such a structured code will have significance, namely, the type-U and type-V codewords.

Definition 3 (Type-U and type-V codewords)**.** We consider a generalized ternary $(U, U+V)$-code C and reuse the above notations. We call a type-U codeword in C a word of form $(\mathbf{a} \star \mathbf{u} \,\|\mathbf{c} \star \mathbf{u})$ with $\mathbf{u} \in U$. And a type-V codeword in C is a word of form $(\mathbf{b} \star \mathbf{v} \,\|\mathbf{d} \star \mathbf{v})$ with $\mathbf{v} \in V$.

Proposition 2 ([Sen23] p.6)**.** *Consider a generalized ternary $(U, U+V)$-code C whose code U has dimension k_U and V dimension k_V. For a target weight t, we expect the number of type-U codewords of C to be on average $\binom{n/2}{t/2}\frac{2^{t/2}}{3^{n/2-k_U}}$, and the number of type-V codewords of C to be on average $\binom{n/2}{t/2}\frac{2^{t/2}}{3^{n/2-k_V}}$.*

Key-distinguishing attacks. From the proposition just above, for some weights t, the expected number of type-U codewords of this weight is higher than those expected for a random code, recalled in Proposition 1. Then, one can use this fact to exhibit a type-U or type-V codeword, that provides a distinguisher of the Wave public key from the uniform, namely solving the DWK_{n,k_U,k_V} problem. This is the goal of *key-distinguishing attacks* on Wave. Notice that one can run this attack directly on the $(U, U+V)$-code, but also on its dual code.

Problem 2 (Distinguishing Wave Keys DWK_{n,k_U,k_V})**.** *Given a column-permutation of $\boldsymbol{H} \in \mathbb{F}_3^{(n-(k_U+k_V))\times n}$, decide whether $\boldsymbol{H}$ has been chosen uniformly at random or among parity-check matrices of permuted generalized $(U, U+V)$-codes where U has dimension k_U, and V dimension k_V.*

Message attacks. Another way to attack Wave is by forging a message-signature pair $(\boldsymbol{e}, \mathbf{s}) \in \mathbb{F}_3^n \times \mathbb{F}_3^{n-k}$ such that $|\boldsymbol{e}| = w$ and $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$. This consists of solving the $\mathsf{DOOM}_{n,k,w}$ problem, which is hard if $\mathsf{DP}_{n,k,w}$ is hard. This problem was introduced in [JJ02] for $\mathbb{F}_2$, and [Sen11] presented an approach for solving it.

Problem 3 (Decoding One Out of Many $\mathsf{DOOM}_{n,k,w}$)**.** *Given an arbitrary large list S of syndromes in $\mathbb{F}_q^{n-k}$, a parity check matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ and a target weight w, find $\boldsymbol{s} \in S$ and $\boldsymbol{e} \in \mathbb{F}_q^n$ such that $|\boldsymbol{e}| = w$ and $\boldsymbol{eH}^\top = \boldsymbol{s}$.*

2.3 Information Set Decoding (ISD) Framework

Attacks on the Decoding Problem are commonly[1] based on the Information Set Decoding (ISD) framework that received several refinements since its introduction by Prange [Pra62]. Stern and Dumer [Ste88, Dum91] improved it by taking advantage of the Generalized Birthday Paradox, and Wagner's approach [Wag02] extended this idea. We use here a framework similar to [FS09], which uses the parity check matrix instead of the generator matrix. Another variant uses representation techniques [MMT11, BJMM12], but this refinement only provides a very slight gain in the Wave setting as shown in [BCDAL20]. With nearest-neighbour techniques [MO15, BM17, BM18, CDMT22], the gain in asymptotic factors is compensated by a high overhead. For these reasons, we will not deal with these techniques in this work and restrict our cryptanalysis to algorithms [Pra62, Dum91, Wag02].

To solve the problem $\mathsf{DP}_{\mathbf{H},\mathbf{s},w}$, the idea behind ISD is to rewrite $\mathbf{H}$ into a systematic form $\mathbf{H} = \left(\begin{array}{c|c} I_{n-k-\ell} & \mathbf{H}' \\ \hline \mathbf{0} & \mathbf{H}'' \end{array}\right) \in \mathbb{F}_q^{n\times(n-k)}$ where $\mathbf{H}' \in \mathbb{F}_q^{(k+\ell)\times(n-k-\ell)}$ and $\mathbf{H}'' \in \mathbb{F}_q^{(k+\ell)\times\ell}$, and then to solve the easier instance $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$with $\mathbf{s}'' \in \mathbb{F}_q^\ell$ for parameters ℓ the length of $\mathbf{s}''$ and p the target weight of $\boldsymbol{e}''$. We need to find many solutions $\boldsymbol{e}'' \in \mathbb{F}_q^{k+\ell}$ to the subproblem $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$ to hope to get one of them that gives a complete solution $\boldsymbol{e} = (\boldsymbol{e}' \| \boldsymbol{e}'') \in \mathbb{F}_3^n$ to the $\mathsf{DP}_{\mathbf{H},\mathbf{s},w}$ problem.

[1] A quite recent paper [CDMT22] presented a way to make the statistical decoding [Jab01] perform better than ISD algorithms in some regimes. Except for this algorithm, all the known attacks on DP for the sixty last years were based on the ISD framework.

To analyze the complexity of the ISD framework, we will need the following lemma.

Lemma 1. *Let $\boldsymbol{e} \in \mathbb{F}_3^n$ be a vector of weight w and integers ℓ, p. Let us define the subvectors $\boldsymbol{e}' \in \mathbb{F}_3^{n-k-\ell}$ and $\boldsymbol{e}'' \in \mathbb{F}_3^{k+\ell}$ such that $\boldsymbol{e} = (\boldsymbol{e}' \| \boldsymbol{e}'')$. We say that $\boldsymbol{e}$ is well cut if $|\boldsymbol{e}'| = w - p$ and $|\boldsymbol{e}''| = p$. The probability that a random $\boldsymbol{e} \in \mathbb{F}_3^n$ of weight w is well cut is*

$$Pr_{GoodCut} = \frac{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}{\binom{n}{w}}. \tag{1}$$

Classical ISD algorithm.

We reuse the above notations for $\mathbf{H}, \mathbf{H}'', \mathbf{s}, \mathbf{s}''$.

Theorem 3. *[FS09] We are given a classical procedure that finds $N_{SolFound}$ solutions to $\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}$ in time $T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}$, out of a total of $N_{Sol}(\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p})$ solutions to $\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}$. Let $Pr_{GoodCut}$ be defined as in Lemma 1, and let $N_{Sol}(\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w})$ denote the total number of solutions to the $\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}$ problem. Then the classical Information Set Decoding framework (Algorithm 1) solves the $\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}$ problem in average time*

$$T_{DP} = \max\left\{ T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}, \frac{T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}}{N_{Sol}(\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}) \cdot Pr_{GoodCut} \cdot \frac{N_{SolFound}}{N_{Sol}(\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p})}} \right\}$$

up to a polynomial factor in n.

The term on the left inside the max is the complexity when only one iteration of Steps 1-4 in the ISD suffices, while the term on the right is the one for several iterations.

Algorithm 1 Classical ISD [FS09]

```
Input: $\mathbf{H}_0 \in \mathbb{F}_3^{n \times (n-k)}$, syndrome $\mathbf{s} \in \mathbb{F}_3^{n-k}$, weight $w$.
Parameters $\ell \in [0, n-k]$ and $p \in [\max\{0, w-(n-k-\ell)\}, \min\{w, k+\ell\}]$
Output: $\boldsymbol{e}_0 \in \mathbb{F}_3^n$ such that $\boldsymbol{e}_0\mathbf{H}_0^\top = \mathbf{s}$ and $|\boldsymbol{e}_0| = w$.
1: Pick a random permutation of columns $\pi$ and apply $\mathbf{H} \leftarrow \pi(\mathbf{H}_0)$
2: Apply a partial Gaussian Elimination on $\mathbf{H}$ to transform it into a systematic form
   $\mathbf{H} = \left(\begin{array}{c|c} I_{n-k-\ell} & \mathbf{H}' \\ \hline \mathbf{0} & \mathbf{H}'' \end{array}\right) \in \mathbb{F}_q^{n \times (n-k)}$ where $\mathbf{H}' \in \mathbb{F}_q^{(k+\ell) \times (n-k-\ell)}$ and $\mathbf{H}'' \in \mathbb{F}_q^{(k+\ell) \times \ell}$,
   and a syndrome $\mathbf{s} = (\mathbf{s}' \| \mathbf{s}'') \in \mathbb{F}_q^{n-k}$ with $\mathbf{s}' \in \mathbb{F}_q^{n-k-\ell}$ and $\mathbf{s}'' \in \mathbb{F}_q^{\ell}$.
3: Solve the subproblem $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$: Construct a list of vectors $(\boldsymbol{e}'', \boldsymbol{e}''\mathbf{H}''^\top) \in \mathbb{F}_q^{k+\ell} \times \mathbb{F}_q^{\ell}$
   such that $|\boldsymbol{e}''| = p$ and $\boldsymbol{e}''\mathbf{H}''^\top = \mathbf{s}''$.
4: Test step: For each $\boldsymbol{e}''$ found during Step 3, recover the complete vector $\boldsymbol{e} = (\boldsymbol{e}' \| \boldsymbol{e}'')$
   that satisfies $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$, and check if $|\boldsymbol{e}| = w$.
5: Repeat Steps 1-4 until Step 4 succeeds and gives a $\boldsymbol{e} \in \mathbb{F}_3^n$ such that $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$ and
   $|\boldsymbol{e}| = w$.
6: return $\boldsymbol{e}_0 = \pi^{-1}(\boldsymbol{e})$.
```

Let us informally explain the rationale behind the complexity expression in Theorem 3 for the Algorithm 1.

Steps 1 and 2. Applying a random permutation of columns and a partial Gaussian elimination on $\mathbf{H}$ can be done in polynomial time.

Step 3. This step takes time $T_{\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}}$ that depends on the choice of the subroutine. It returns $N_{SolFound}$ solutions to the $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$ subproblem.

Step 4. From an $\boldsymbol{e}'' \in \mathbb{F}_3^{k+\ell}$ such that $\boldsymbol{e}''\mathbf{H}''^{\top} = \mathbf{s}''$, one can efficiently compute $\boldsymbol{e}' = \mathbf{s}' - \boldsymbol{e}''\mathbf{H}'^{\top} \in \mathbb{F}_3^{n-k-\ell}$. The vector $\boldsymbol{e} = (\boldsymbol{e}' \| \boldsymbol{e}'')$ then satisfies $\boldsymbol{e}\mathbf{H}^{\top} = \mathbf{s}$. There are $N_{SolFound}$ solutions that need to be checked for weight, which can be done for each one in polynomial time.

Step 5. Suppose there is a precise solution $\boldsymbol{e}$ that we want to find, where $\boldsymbol{e} = (\boldsymbol{e}' \| \boldsymbol{e}'')$ with $\boldsymbol{e}' \in \mathbb{F}_3^{k+\ell}$ and $\boldsymbol{e}'' \in \mathbb{F}_3^{n-k-\ell}$. The probability that $\boldsymbol{e}$ is "well cut", *i.e.*, for $\boldsymbol{e}$ there is $|\boldsymbol{e}'| = w - p$ and $|\boldsymbol{e}''| = p$, is given by Lemma 1. Then, supposing $\boldsymbol{e}$ is well cut, one iteration of Steps 1-4 returns a list containing a fraction $\frac{N_{SolFound}}{N_{Sol}(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p})}$ of the solutions to the $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$ subproblem. As there are $N_{Sol}(\mathsf{DP}_{\mathbf{H},\mathbf{s},w})$ such solutions $\boldsymbol{e}$, the probability that one iteration returns a solution is

$$Pr_{FindSol} = \min \left\{ 1, N_{Sol}(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) \cdot Pr_{GoodCut} \cdot \frac{N_{SolFound}}{N_{Sol}(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p})} \right\}. \quad (2)$$

After $1/Pr_{FindSol}$ iterations, one finds a solution with a constant probability of success $1-\epsilon$, where $\epsilon = (1 - 1/Pr_{FindSol})^{1/Pr_{FindSol}} < 1/e = o(1)$. Once we have found an $\boldsymbol{e}$ such that $\boldsymbol{e}\mathbf{H}^{\top} = \mathbf{s}$, we return $\boldsymbol{e}_0 := \pi^{-1}(\boldsymbol{e})$, where π^{-1} is the reversed permutation applied in Step 1. We have $\pi^{-1}(\boldsymbol{e})\pi^{-1}(\mathbf{H})^{\top} = \boldsymbol{e}_0\mathbf{H}_0^{\top} = \mathbf{s}$ and $|\boldsymbol{e}_0| = |\boldsymbol{e}| = w$, so $\boldsymbol{e}_0$ is indeed a solution to the problem. Applying π^{-1} can be done in polynomial time.

Quantum ISD algorithm.

A quantum algorithm for Information Set Decoding is described in [Ber10]. The following algorithm is similar, and rewritten to match the notations we will use throughout the rest of the article.

Notations. We recall that given a quantumly accessible list L, $\mathrm{ind}_L(\boldsymbol{x})$ denotes the index of element $\boldsymbol{x}$ in the list L. The quantum superposition of the elements of list L is the quantum state $|\psi_L\rangle = \frac{1}{\sqrt{|L|}} \sum_{\boldsymbol{x} \in L} |\mathrm{ind}_L(\boldsymbol{x})\rangle |\boldsymbol{x}\rangle$ (See Section 2.1).

Theorem 4 ([Ber10]). *We are given a procedure that constructs a quantum superposition of $N_{SolFound}$ solutions to $\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}$ in time $T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}$, out of a total of $N_{Sol}(\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p})$ solutions to $\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}$. Let $Pr_{GoodCut}$ be defined as in Lemma 1, and let $N_{Sol}(\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w})$ denote the total number of solutions to the $\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}$ problem. Then the quantum Information Set Decoding framework (Algorithm 2) solves the $\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}$ problem in average time*

$$T_{DP} = \max \left\{ T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}, \frac{T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}}{\sqrt{N_{Sol}(\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}) \cdot Pr_{GoodCut} \cdot \frac{N_{SolFound}}{N_{Sol}(\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p})}}} \right\}$$

up to a polynomial factor in n.

Let us explain the rationale behind this formula.

Steps 1 and 2. These steps do not change from the classical version and are efficiently done.

Step 3. This operation takes time $T_{\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}}$ that depends on the choice of the subroutine solving the problem. It returns a quantum superposition over $|L| = N_{SolFound}$ solutions to the $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$ subproblem. In the quantum state $|\psi_L\rangle$, all the $\boldsymbol{e}''\mathbf{H}''^{\top}$ in the tuples of L are equal to $\mathbf{s}''$ as it is an output condition of the solver subroutine. Therefore, the last register now can be considered classical, and possibly be discarded.

Step 4. We add a zero quantum register to the constructed quantum state to get

$$\frac{1}{\sqrt{|L|}} \sum_{\boldsymbol{e}'' \in L} |\mathrm{ind}_L(\boldsymbol{e}'')\rangle |0\rangle |\boldsymbol{e}''\rangle$$

Algorithm 2 Quantum ISD

```
Input: $\mathbf{H}_0 \in \mathbb{F}_3^{n \times (n-k)}$, syndrome $\mathbf{s} \in \mathbb{F}_3^{n-k}$, weight $w$.
Parameters $\ell \in [0, n-k]$ and $p \in [\max\{0, w - (n-k-\ell)\}, \min\{w, k+\ell\}]$
Output: $\boldsymbol{e}_0 \in \mathbb{F}_3^n$ such that $\boldsymbol{e}_0 \mathbf{H}_0^\top = \mathbf{s}$ and $|\boldsymbol{e}_0| = w$.
1: Pick a random permutation of columns $\pi$ and apply $\mathbf{H} \leftarrow \pi(\mathbf{H}_0)$
2: Apply a partial Gaussian Elimination on $\mathbf{H}$ to transform it into a systematic form
   $\mathbf{H} = \left(\begin{array}{c|c} I_{n-k-\ell} & \mathbf{H}' \\ \hline \mathbf{0} & \mathbf{H}'' \end{array}\right) \in \mathbb{F}_q^{n \times (n-k)}$ where $\mathbf{H}' \in \mathbb{F}_q^{(k+\ell) \times (n-k-\ell)}$ and $\mathbf{H}'' \in \mathbb{F}_q^{(k+\ell) \times \ell}$,
   and a syndrome $\mathbf{s} = (\mathbf{s}' \| \mathbf{s}'') \in \mathbb{F}_q^{n-k}$ with $\mathbf{s}' \in \mathbb{F}_q^{n-k-\ell}$ and $\mathbf{s}'' \in \mathbb{F}_q^{\ell}$.
3: Solve the subproblem $\mathsf{DP}_{\mathbf{H}'', \mathbf{s}'', p}$: Apply the procedure that constructs in quantum
   superposition over the elements of a list $L$ containing tuples $(\boldsymbol{e}'', \boldsymbol{e}''\mathbf{H}''^\top) \in \mathbb{F}_3^{k+\ell} \times \mathbb{F}_3^{\ell}$
   such that $|\boldsymbol{e}''| = p$ and $\boldsymbol{e}''\mathbf{H}''^\top = \mathbf{s}''$, i.e.,
       $$|0\rangle \rightarrow |\psi_L\rangle := \frac{1}{\sqrt{|L|}} \sum_{(\boldsymbol{e}'', \mathbf{s}'') \in L} \left|\mathrm{ind}_L(\boldsymbol{e}'')\right\rangle \left|\boldsymbol{e}''\right\rangle \left|\mathbf{s}''\right\rangle$$
   where $\mathrm{ind}_L(\boldsymbol{e}'')$ denotes the index of $\boldsymbol{e}''$ in list $L$. Discard the classical register $\left|\mathbf{s}''\right\rangle$.
4: Test step: From this previous state, by computing the complete candidate solution
   $\boldsymbol{e} \in \mathbb{F}_3^n$ such that $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$, construct
       $$|\Psi\rangle := \frac{1}{\sqrt{|L|}} \sum_{\substack{(\boldsymbol{e}'', \boldsymbol{y}) \in L \\ \boldsymbol{e} = (\mathbf{s}' - \boldsymbol{e}''\mathbf{H}'^\top \| \boldsymbol{e}'')}} \left|\mathrm{ind}_L(\boldsymbol{e}'')\right\rangle |\boldsymbol{e}\rangle .$$
   The vectors $\boldsymbol{e}$ in the second register then satisfy $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$. Apply Grover, iterating
   on the operation $|0\rangle \rightarrow |\Psi\rangle$, to only keep the $\boldsymbol{e}$'s in the superposition which are of
   weight $w$.
5: Apply an Amplitude Amplification on Steps 1-4 to find a good permutation $\pi$ in
   Step 1 with high probability.
6: Measure $\boldsymbol{e}$. Return $\boldsymbol{e}_0 = \pi^{-1}(\boldsymbol{e})$.
```

where $\mathrm{ind}_L(\boldsymbol{e}'')$ is the index of the tuple $(\boldsymbol{e}'', \boldsymbol{e}''\mathbf{H}''^\top)$ in list L. We apply the efficient quantum circuit $|0\rangle \left|\boldsymbol{e}''\right\rangle \mapsto \left|\mathbf{s}' - \mathbf{H}'\boldsymbol{e}''\right\rangle \left|\boldsymbol{e}''\right\rangle$ on its two last registers to get the state

$$|\Psi\rangle := \frac{1}{\sqrt{|L|}} \sum_{\substack{(\boldsymbol{e}'', \boldsymbol{y}) \in L \\ \boldsymbol{e}' = \mathbf{s}' - \mathbf{H}'\boldsymbol{e}''}} \left|\mathrm{ind}_L(\boldsymbol{e}'')\right\rangle \left|\boldsymbol{e}'\right\rangle \left|\boldsymbol{e}''\right\rangle = \frac{1}{\sqrt{|L|}} \sum_{\substack{\boldsymbol{e}'' \in L \\ \boldsymbol{e} = (\mathbf{s}' - \mathbf{H}'\boldsymbol{e}'' \| \boldsymbol{e}'')}} \left|\mathrm{ind}_L(\boldsymbol{e}'')\right\rangle |\boldsymbol{e}\rangle .$$

This state is a uniform quantum superposition over candidate solutions $\boldsymbol{e} \in \mathbb{F}_3^n$ that satisfy $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$. We need to only keep those that are of good weight w, so we apply a Grover search [Gro96] that iterates the procedure $|0\rangle \rightarrow |\Psi\rangle$. Its check function takes $\boldsymbol{e}$ and returns 1 if $|\boldsymbol{e}| = w$, and 0 otherwise. It returns the following quantum superposition over solutions to $\mathsf{DP}_{\mathbf{H}, \mathbf{s}, w}$, which is

$$\frac{1}{\sqrt{Z}} \sum_{\substack{(\boldsymbol{e}'', \boldsymbol{y}) \in L \\ \boldsymbol{e} = (\mathbf{s}' - \mathbf{H}'\boldsymbol{e}'' \| \boldsymbol{e}'') \\ |\boldsymbol{e}| = w}} \left|\mathrm{ind}_L(\boldsymbol{e}'')\right\rangle |\boldsymbol{e}\rangle ,$$

where Z is the number of such solutions $\boldsymbol{e}$ generated from the vectors $\boldsymbol{e}''$ computed during Step 3 and of good weight. This requires at most $\sqrt{|L|} = \sqrt{N_{SolFound}}$ Grover iterations.

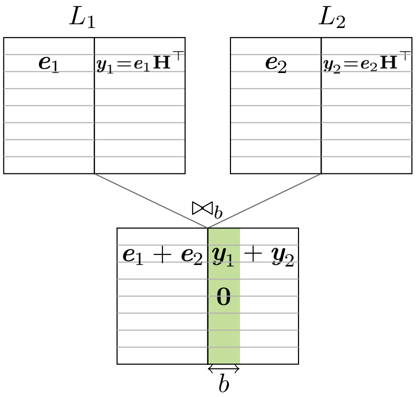
L_1
L_2
$\boldsymbol{e}_1$
$\boldsymbol{y}_1 = \boldsymbol{e}_1 \mathbf{H}^\top$
$\boldsymbol{e}_2$
$\boldsymbol{y}_2 = \boldsymbol{e}_2 \mathbf{H}^\top$
$\bowtie_b$
$\boldsymbol{e}_1 + \boldsymbol{e}_2$
$\boldsymbol{y}_1 + \boldsymbol{y}_2$
$\mathbf{0}$
b

Figure 1: Merging lists L_1 and L_2 on the b first coordinates.

Step 5. Suppose there is a precise solution $\boldsymbol{e}$ that we want to find where $\boldsymbol{e} = (\boldsymbol{e}' \| \boldsymbol{e}'')$ with $\boldsymbol{e}' \in \mathbb{F}_3^{k+\ell}$ and $\boldsymbol{e}'' \in \mathbb{F}_3^{n-k-\ell}$. Lemma 1 gives the probability that $\boldsymbol{e}$ is "well cut", *i.e.*, $|\boldsymbol{e}'| = w - p$ and $|\boldsymbol{e}''| = p$. Then, supposing $\boldsymbol{e}$ is well cut, one iteration of Steps 1-4 returns a list in quantum superposition containing a fraction $\frac{N_{SolFound}}{N_{Sol}(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p})}$ of the solutions to the $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$ subproblem. One iteration of Steps 1-4 returns a solution with probability $Pr_{FindSol}$ whose expression is in Equation (2). To get a solution with a probability close to 1, we apply an amplitude amplification on this process, which takes $1/\sqrt{Pr_{FindSol}}$ iterations. Then we measure and find a solution to $\mathsf{DP}_{\mathbf{H},\mathbf{s},w}$.

DOOM variant of the ISD.

[Sen11] presented an approach for solving more efficiently the DOOM problem. Instead of having only one syndrome $\mathbf{s}$ in the input of the ISD frameworks 1 and 2, the adversary takes an arbitrarily large list S of syndromes. At the end of the algorithm, the adversary wins if they get a pair $(\boldsymbol{e}_0, \mathbf{s}) \in \mathbb{F}_3^n \times S$ such that $\mathbf{e}_0 \mathbf{H}_0 = \mathbf{s}$. The subroutine of the third step is also adapted in function: it takes in input a list S'' of the restricted syndromes $\mathbf{s}''$, and outputs solutions $(\boldsymbol{e}'', \mathbf{s}'') \in \mathbb{F}_3^{k+\ell} \times \mathbb{F}_3^{\ell}$ to the subproblem, where $\mathbf{s}''$ are restrictions of the $\mathbf{s} \in S$ on their ℓ last coordinates. The time complexity of this variant stays the same as the one given in Theorem 3 for classical and in Theorem 4 for quantum. This approach will be applied and explained in more detail in Section 4 in the context of message attacks on Wave.

2.4 List merging

Subroutines within the framework of ISD will make great use of list merging. In this context, we will consider lists L_1 and L_2 where each element is a couple of error $\boldsymbol{e}$ and syndrome $\boldsymbol{y}$. The objective is to merge the lists by summing the elements so that the obtained syndromes have a part of 0. Merging two lists L_1 and L_2 on the b first coordinates means constructing the following list.

$$L_1 \bowtie_b L_2 := \Big\{ (\boldsymbol{e}_1 + \boldsymbol{e}_2, \boldsymbol{y}_1 + \boldsymbol{y}_2) \ : \ (\boldsymbol{e}_1, \boldsymbol{y}_1) \in L_1, \ (\boldsymbol{e}_2, \boldsymbol{y}_2) \in L_2, \\ (\boldsymbol{y}_1 + \boldsymbol{y}_2)_{|[0:b-1]} = \mathbf{0} \Big\} \tag{3}$$

Size of the merged list.

For lists L_1 and L_2 randomly sampled in $\mathbb{F}_3^n \times \mathbb{F}_3^\ell$, their merged list is of expected size $|L_1 \bowtie_b L_2| = \frac{|L_1|\cdot|L_2|}{3^b}$ on average. Then for lists L_1, L_2 already merged so that their vectors have already their b_0 first coordinates at zero, we have on average for $b \geq b_0$,

$$|L_1 \bowtie_b L_2| = \frac{|L_1| \cdot |L_2|}{3^{b-b_0}}. \tag{4}$$

Classical merging.

We want to construct the merged list $L = L_1 \bowtie_b L_2$. We sort L_1 by lexicographic order according to its second tuple elements $\boldsymbol{y}_1$, which takes time $|L_1| \cdot \log(|L_1|)$. Then, for each $(\boldsymbol{e}_2, \boldsymbol{y}_2) \in E_2$, we search $(\boldsymbol{e}_1, \boldsymbol{y}_1) \in L_1$ such that $\boldsymbol{y}_1 + \boldsymbol{y}_2$ values $\mathbf{0}$ on its b first coordinates. Thanks to the sorting, for each $\boldsymbol{e}_2$ one can find a solution in L_2 (if it exists) in time $\log|L_2|$ by dichotomic search. For each collision found on $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, we add $(\boldsymbol{e}_1 + \boldsymbol{e}_2, \boldsymbol{y}_1 + \boldsymbol{y}_2)$ to L. So the classical merging takes time $\big(|L_1| + |L_2|\big) \cdot \log|L_1|$. Hence the following lemma.

Lemma 2 ([Wag02])**.** *Given lists L_1 and L_2, one can construct the list $L_1 \bowtie_b L_2$ for an arbitrary b in time $\widetilde{\mathcal{O}}(|L_1|, |L_2|, |L_1 \bowtie_b L_2|)$.*

Quantum merging.

We are given a list L_1 classically stored and assumed quantumly accessible, and a procedure $|0\rangle \to |\psi_{L_2}\rangle$ that returns in time T the uniform quantum superposition on the list L_2,

$$|\psi_{L_2}\rangle = \frac{1}{\sqrt{|L_2|}} \sum_{(\boldsymbol{e}_2, \boldsymbol{y}_2) \in L_2} \big|\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\big\rangle\, |\boldsymbol{e}_2\rangle\, |\boldsymbol{y}_2\rangle\,. \tag{5}$$

We sort L_1 in the lexicographic order according to its second tuple elements $\boldsymbol{y}_1$, which takes time $|L_1| \cdot \log(|L_1|)$. We define the following function:

$$\mathit{match}_{L_1}(\boldsymbol{e}_2, \boldsymbol{y}_2) = \begin{cases} (\boldsymbol{e}_1, \boldsymbol{y}_1) \in L_1 \text{ such that } (\boldsymbol{y}_1 + \boldsymbol{y}_2)_{|[0:b]} = \mathbf{0} \text{ if it exists,} \\ \perp \text{ otherwise.} \end{cases}$$

If several such $(\boldsymbol{e}_1, \boldsymbol{y}_1)$'s match, the function will arbitrarily return the first one by lexicographic order. However, if lists L_1, L_2 are random and there is $|L_1| \leq 3^{b-b_0}$, then there will be on average at most one match in L_1 for any given element from L_2. So we make this assumption by simplicity from here[2].

The function match_{L_1} is efficiently implementable by performing a dichotomic search as L_1 is sorted and assumed quantumly accessible. We then apply this circuit on $|\psi_{L_2}\rangle$ using an auxiliary register:

$$\frac{1}{\sqrt{|L_2|}} \sum_{(\boldsymbol{e}_2, \boldsymbol{y}_2) \in L_2} \big|\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\big\rangle \big|\mathit{match}_{L_1}(\boldsymbol{e}_2, \boldsymbol{y}_2)\big\rangle\, |\boldsymbol{e}_2\rangle\, |\boldsymbol{y}_2\rangle\,.$$

While the classical merging ran a loop on L_2 to search for matching elements, in the quantum model we replace it with Grover's search [Gro96] that iterates on the procedure $|0\rangle \to |\psi_{L_2}\rangle$. We define the Grover check function as follows: given $(\boldsymbol{e}_2, \boldsymbol{y}_2)$, it returns 1 if

[2] When we will apply quantum merging further in this work, we will manipulate random lists L_1, L_2 such that $|L_1| \leq 3^{b-b_0}$ and $|L_2| = |L_1|^2$, so there will be at most one solution with very high probability. This allows us to consider that this quantum merging process constructs a quantum superposition over the list $L_1 \bowtie_b L_2$ without missing any element.

$match_{L_1}(\boldsymbol{e}_2, \boldsymbol{y}_2) \neq \perp$, and 0 else. Applying Grover requires at most $\sqrt{|L_2|}$ iterations, and returns a state as close as we want to the following state.

$$\frac{1}{\sqrt{|L_1 \bowtie_b L_2|}} \sum_{\substack{(\boldsymbol{e}_2,\boldsymbol{y}_2)\in L_2 \\ match_{L_1}(\boldsymbol{e}_2,\boldsymbol{y}_2)=(\boldsymbol{e}_1,\boldsymbol{y}_1)\neq\perp}} \left|\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\right\rangle |\boldsymbol{e}_1\rangle |\boldsymbol{y}_1\rangle |\boldsymbol{e}_2\rangle |\boldsymbol{y}_2\rangle .$$

And finally, by simply summing, swapping and reassembling the registers, we get the state

$$\frac{1}{\sqrt{|L_1 \bowtie_b L_2|}} \sum_{\substack{(\boldsymbol{e}_2,\boldsymbol{y}_2)\in L_2 \\ match_{L_1}(\boldsymbol{e}_2,\boldsymbol{y}_2)=(\boldsymbol{e}_1,\boldsymbol{y}_1)\neq\perp}} \left|\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\right\rangle |\boldsymbol{e}_1 + \boldsymbol{e}_2\rangle |\boldsymbol{y}_1 + \boldsymbol{y}_2\rangle |\boldsymbol{e}_2, \boldsymbol{y}_2\rangle ,$$

where the last register $|\boldsymbol{e}_2, \boldsymbol{y}_2\rangle$ cannot be discarded because of the requirement of the reversibility of the process, but it will not be used anymore. The previous state then can be rewritten

$$\left|\psi_{L_1 \bowtie_b L_2}\right\rangle |\mathrm{Aux}\rangle := \frac{1}{\sqrt{|L_1 \bowtie_b L_2|}} \sum_{\substack{(\boldsymbol{e},\boldsymbol{y})\in L_1\bowtie_b L_2 \\ \boldsymbol{e}=\boldsymbol{e}_1+\boldsymbol{e}_2, \boldsymbol{e}_1\in L_1, \boldsymbol{e}_2\in L_2}} \left|\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\right\rangle |\boldsymbol{e}\rangle |\boldsymbol{y}\rangle |\mathrm{Aux}\rangle . \quad (6)$$

This whole process takes time $(|L_1| + T\sqrt{|L_2|}) \cdot \log |L_1|$. This leads to the following lemma.

Lemma 3 ([Sch22])**.** *Given a list L_1 classically stored and quantumly accessible, and a procedure that returns the quantum state $|\psi_{L_2}\rangle$ (defined in Equation (5)) in time T, for $|L_1| \leq |L_2|$, there exists a quantum algorithm that returns the state $\left|\psi_{L_1 \bowtie_b L_2}\right\rangle$ (Eq. 6) for an arbitrary b in time $\widetilde{\mathcal{O}}(|L_1|, T \cdot \sqrt{|L_2|})$.*

3 Key-distinguishing attacks

We are given a public $(U, U+V)$-code with generator matrix $\mathbf{G} \in \mathbb{F}_3^{(k_U+k_V)\times n}$ and parity check matrix $\mathbf{H} \in \mathbb{F}_3^{(n-(k_U+k_V))\times n}$. The point of this attack is to solve the problem of Distinguishing Wave Keys (Problem 2), which can be done by finding a type-U or type-V word $\boldsymbol{e}$ of weight t in the public code or its dual. It was shown in [Sen23] that type-U words outnumber type-V ones, so the attacker can restrain their search to only type-U words as they are easier to find. (See Definition 3 for type-U and -V words.) The target weight t can be chosen as the attacker wants under the condition that the number of such words has to be higher than in a random code. The former condition, by combining Propositions 1 and 2, is equivalent to

$$3^{n-2\cdot k_V} > \binom{n}{t} 2^t. \quad (7)$$

So the time complexity of this key-distinguishing attack is the minimum between the time of solving the problems $\mathsf{DP}_{\mathbf{H},\mathbf{0},t}$ and $\mathsf{DP}_{\mathbf{G},\mathbf{0},t'}$, respectively to find a type-U word in the public code and in its dual, where t and t' are freely chosen as long as they respect Equation (7). In this section, we present how to solve $\mathsf{DP}_{\mathbf{H},\mathbf{0},t}$ and these algorithms can straightforwardly be adapted to the dual version.

3.1 Classical key-distinguishing attack

The best known classical key attack on Wave is due to [Sen23], who applies Dumer's algorithm [Dum91] within the ISD framework [FS09]. We describe here their algorithm to introduce the ideas and notations we will need for our quantum version of this attack. We start by constructing the following lists.

$$\begin{aligned} E_1 &:= \{(\boldsymbol{x}_1 \parallel \mathbf{0}^{\frac{k+\ell}{2}}) \mid \boldsymbol{x}_1 \in \mathbb{F}_3^{\frac{k+\ell}{2}}, |\boldsymbol{x}_1| = p/2\} \quad ; \quad L_1 := \{(\boldsymbol{e}_1'', \boldsymbol{e}_1''\mathbf{H}''^\top) : \boldsymbol{e}_1'' \in E_1\} \\ E_2 &:= \{(\mathbf{0}^{\frac{k+\ell}{2}} \parallel \boldsymbol{x}_2) \mid \boldsymbol{x}_2 \in \mathbb{F}_3^{\frac{k+\ell}{2}}, |\boldsymbol{x}_2| = p/2\} \quad ; \quad L_2 := \{(\boldsymbol{e}_2'', \boldsymbol{e}_2''\mathbf{H}''^\top) : \boldsymbol{e}_2'' \in E_2\} \end{aligned} \tag{8}$$

Both these initial lists are of size

$$\binom{(k+\ell)/2}{p/2} 2^{p/2} = \widetilde{\mathcal{O}}\left(\binom{k+\ell}{p}^{1/2} 2^{p/2}\right). \tag{9}$$

We apply classical merging from Lemma 2 on the lists L_1 and L_2 to get the merged list $L_1 \bowtie_\ell L_2$ filled with elements of form $(\boldsymbol{e}'', \boldsymbol{e}''\mathbf{H}''^\top) = (\boldsymbol{e}'', \mathbf{0})$. In Equation (8) we see that vectors $\boldsymbol{e}_1'' \in E_1$ and $\boldsymbol{e}_2'' \in E_2$ have disjoint sets of non-zero coordinates. Therefore, their sum $\boldsymbol{e}''$ that we get through the merged list is in form $\boldsymbol{e}'' = \boldsymbol{e}_1'' + \boldsymbol{e}_2'' = (\boldsymbol{x}_1 \| \boldsymbol{x}_2)$ with $|\boldsymbol{x}_1| = |\boldsymbol{x}_2| = p/2$. So all the $\boldsymbol{e}''$ for $(\boldsymbol{e}'', \boldsymbol{e}''\mathbf{H}'^\top) \in L$ obtained are of weight p by construction, ensuring the algorithm's correctness. Using this process as a subroutine within the ISD framework leads to the following theorem.

Theorem 5 ([Sen23])**.** *We are given a generalized ternary $(U, U+V)$-code C of parameters (n, k, k_U, k_V). Fix ISD parameters ℓ, p, and a target weight t. There exists a classical algorithm that solves the DWK_{n,k_U,k_V} problem for code C in time*

$$T = \widetilde{\mathcal{O}}\left(\max\left\{\binom{k+\ell}{p}^{1/2} 2^{p/2}, \frac{3^{n/2-k_U}\binom{n}{t}^{1/2}}{2^{(t-p)/2}\binom{k+\ell}{p}^{1/2}\binom{n-k-\ell}{t-p}}\right\}\right).$$

Remark. We have observed that Wagner's algorithm [Wag02] does not perform better than Dumer [Dum91] in this setting, *i.e.*, one does not get profit from taking additional levels in the merging tree. The reason is that the condition in Equation (7) forces the target weight t to remain quite small. And this impacts the number of vectors one can generate with this weight, which is low in comparison to those with higher weights (as we are in ternary). Additional merging levels are useful when there are sufficiently many vectors, which is not the case here, but it will have an advantage in a different setting, for the message attacks, as we will show in Section 4.

Numerical results.

The time complexity is optimal when both initial lists are of maximal size, *i.e.*, by fixing p such that $\binom{k+\ell}{p}^{1/2} 2^{p/2} = 3^\ell$. Parameters ℓ and t are obtained by numerical optimization to minimize the time complexity of the attack. As a result, we obtain as optimal parameters $\ell \approx 0.01n$, $t \approx 0.21n$ and $p \approx 0.003n$. With these values, the ISD algorithm with a Dumer subroutine solves DWK_{n,k_U,k_V} in time $2^{0.0161n+o(n)}$, *i.e.*, 2^{138} for the set of Wave parameters (I); in time $2^{0.0165n+o(n)}$, *i.e.*, 2^{206} for set (III); and in time $2^{0.0167n+o(n)}$, *i.e.*, 2^{274} for (V).

The sets of Wave parameters (I), (III) and (V) are recalled in Table 3.

With this choice of parameters, the algorithm finds $|L|$ solutions in time $|L|$, so in amortized time $\mathcal{O}(1)$ per solution. The $o(n)$ terms above encapsulate the hidden polynomial terms, as our analysis only focused on the asymptotic complexity. These results are summarized in the first column of Table 1, in the introduction.

3.2 Quantum key-distinguishing attack

Our quantum key-distinguishing attack algorithm reuses the general structure of the classical one by [Sen23], but with adaptations to get the best profit in the quantum setting. We use the quantum version of the ISD framework (Algorithm 2), which performs a quantum Amplitude Amplification (Theorem 2 [BHMT02]) instead of a classical 'repeat' loop. This makes a quadratic gain over the number of iterations of the algorithm. Within the ISD framework, we also replace the classical Dumer subroutine with its quantum merging variant. Remember that the input parity check matrix $\mathbf{H}$ is written in a systematic form $\mathbf{H} = \left(\begin{array}{c|c} I_{n-k-\ell} & \mathbf{H}' \\ \hline \mathbf{0} & \mathbf{H}'' \end{array}\right) \in \mathbb{F}_q^{n\times(n-k)}$, where $\mathbf{H}' \in \mathbb{F}_q^{(k+\ell)\times(n-k-\ell)}$ and $\mathbf{H}'' \in \mathbb{F}_q^{(k+\ell)\times\ell}$, and here once again we deal with the subproblem DP on matrix $\mathbf{H}''$ instead of $\mathbf{H}$.

Let us define the following lists. Note that the list L_2 does not need to be classically written at any moment of the algorithm.

$$\begin{aligned} E_1 &:= \{(\boldsymbol{x}_1 \parallel \mathbf{0}^{\frac{2(k+\ell)}{3}}) \mid \boldsymbol{x}_1 \in \mathbb{F}_3^{\frac{k+\ell}{3}}, |\boldsymbol{x}_1| = p/3\} \; ; \; L_1 := \{(\boldsymbol{e}''_1, \boldsymbol{e}''_1\mathbf{H}''^{\top}) : \boldsymbol{e}''_1 \in E_1\} \\ E_2 &:= \{(\mathbf{0}^{\frac{k+\ell}{3}} \parallel \boldsymbol{x}_2) \mid \boldsymbol{x}_2 \in \mathbb{F}_3^{\frac{2(k+\ell)}{3}}, |\boldsymbol{x}_2| = 2p/3\} \; ; \; L_2 := \{(\boldsymbol{e}''_2, \boldsymbol{e}''_2\mathbf{H}''^{\top}) : \boldsymbol{e}''_2 \in E_2\} \end{aligned} \tag{10}$$

The lists are no longer of equal size. Indeed, to balance the running times, the list in quantum superposition is taken quadratically larger than the classical one:

$$|L_1| = \widetilde{\mathcal{O}}\left(\binom{k+\ell}{p}^{1/3} 2^{p/3}\right) \quad \text{and} \quad |L_2| = |L_1|^2. \tag{11}$$

The algorithm starts by classically constructing the list L_1. It also constructs the uniform quantum superposition over the elements of L_2: $|\psi_{L_2}\rangle = \frac{1}{\sqrt{|L_2|}} \sum_{(\boldsymbol{e}_2,\boldsymbol{y}_2)\in L_2} \left|\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\right\rangle |\boldsymbol{e}_2\rangle |\boldsymbol{y}_2\rangle$, where $\mathrm{ind}_{L_2}(\boldsymbol{e}_2)$ is the index of the tuple $(\boldsymbol{e}_2, \boldsymbol{y})$ in the list L_2. We apply a quantum merging (Lemma 3) on L_1 and $|\psi_{L_2}\rangle$ to get the state $\left|\psi_{L_1 \bowtie_\ell L_2}\right\rangle$, which contains the quantum superposition of all the $\boldsymbol{e}'' = (\boldsymbol{e}''_1 + \boldsymbol{e}''_2)$ for $\boldsymbol{e}''_1 \in E_1$ and $\boldsymbol{e}''_2 \in E_2$ such that $\boldsymbol{e}''\mathbf{H}''^{\top} = \mathbf{0}$. Please refer to Equation (6) for an explicit expression of this quantum state. As by construction $\boldsymbol{e}''_1 \in E_1$ and $\boldsymbol{e}''_1 \in E_1$ have disjoint set of non-zero coordinates, then $\boldsymbol{e}''$ is of weight $|\boldsymbol{e}''| = |\boldsymbol{e}''_1| + |\boldsymbol{e}''_2| = p/3 + 2p/3 = p$. So we end up with a quantum superposition of $|L_1 \bowtie_\ell L_2|$ solutions to the $\mathsf{DP}_{\mathbf{H}'',\mathbf{0},p}$ subproblem. Using this as a subroutine within the ISD framework leads to the following theorem.

Theorem 6 ([Sen23] restated with explicit formula)**.** *We are given a generalized ternary $(U, U+V)$-code C of dimensions (n, k, k_U, k_V). Fix ISD parameters ℓ, p, and a target weight t. There exists a classical algorithm that solves the DWK_{n,k_U,k_V} problem for code C in time*

$$T = \widetilde{\mathcal{O}}\left(\max\left\{ \binom{k+\ell}{p}^{1/2} 2^{p/2}, \frac{3^{n/2-k_U}\binom{n}{t}^{1/2}}{2^{(t-p)/2}\binom{k+\ell}{p}^{1/2}\binom{n-k-\ell}{t-p}} \right\}\right).$$

Proof. L_1 and L_2 are of the same size given by Equation (9). Constructing the initial lists takes time $\mathcal{O}(|L_1|)$ and merging $L_1 \bowtie_\ell L_2$ takes time $\widetilde{\mathcal{O}}(|L_1|)$ by Lemma 2. So Dumer's subroutine runs in time

$$T_{\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}} = \widetilde{\mathcal{O}}(|L_1|) = \widetilde{\mathcal{O}}\left(\binom{k+\ell}{p}^{1/2} 2^{p/2}\right).$$

The merged list is of expected size $|L_1| \cdot |L_2| \cdot 3^{-\ell} = \widetilde{\mathcal{O}}\left(\binom{k+\ell}{p} 2^p \cdot 3^{-\ell}\right)$, by Equation (4). Proposition 1 gives the number of solutions to the DP subproblem: $N_{Sol}(\mathsf{DP}_{\mathbf{H}'',\mathbf{0},p}) = \binom{k+\ell}{p} 2^p$. And Proposition 2 gives the number of solutions to the DP problem in the $(U, U+V)$-code: $N_{Sol}(\mathsf{DP}_{\mathbf{H},\mathbf{0},t}) = \binom{n/2}{t/2} \frac{2^{t/2}}{3^{n/2-k_V}}$.

Applying Theorem 3 with these amounts gives the time complexity of the ISD framework with a Dumer subroutine. Simplifying the expression directly gives the result. □

Remark. We have observed that Wagner's algorithm does not perform better than Dumer in this setting, *i.e.*, one does not get profit from taking additional levels in the merging tree. The reason is that the condition in Equation (7) forces the target weight t to remain quite small. And this impacts the number of vectors one can generate with this weight, which is low in comparison to those with higher weights (as we are in ternary). Additional merging levels are useful when there are sufficiently many vectors, which is not the case here, but it will have an advantage in a different setting, for the message attacks, as we will show in Section 4.

Numerical results.

The time complexity is optimal when both initial lists are of maximal size, *i.e.*, by fixing p such that $\binom{k+\ell}{p}^{1/2} 2^{p/2} = 3^\ell$. Parameters ℓ and t are obtained by numerical optimization to minimize the time complexity of the attack. As a result, we obtain as optimal parameters $\ell \approx 0.01n$, $t \approx 0.21n$ and $p \approx 0.003n$. With these values, the ISD algorithm with a Dumer subroutine solves DWK_{n,k_U,k_V} in time $2^{0.0161n+o(n)}$, *i.e.*, 2^{138} for the set of Wave parameters (I); in time $2^{0.0165n+o(n)}$, *i.e.*, 2^{206} for set (III); and in time $2^{0.0167n+o(n)}$, *i.e.*, 2^{274} for (V).

The sets of Wave parameters (I), (III) and (V) are recalled in Table 3.

With this choice of parameters, the algorithm finds $|L|$ solutions in time $|L|$, so in amortized time $\mathcal{O}(1)$ per solution. The $o(n)$ terms above encapsulate the hidden polynomial terms, as our analysis only focused on the asymptotic complexity. These results are summarized in the first column of Table 1, in the introduction.

3.3 Quantum key-distinguishing attack

Our quantum key-distinguishing attack algorithm reuses the general structure of the classical one by [Sen23], but with adaptations to get the best profit in the quantum setting. We use the quantum version of the ISD framework (Algorithm 2), which performs a quantum Amplitude Amplification (Theorem 2 [BHMT02]) instead of a classical 'repeat' loop. This makes a quadratic gain over the number of iterations of the algorithm. Within the ISD framework, we also replace the classical Dumer subroutine with its quantum merging variant. Remember that the input parity check matrix $\mathbf{H}$ is written in a systematic form $\mathbf{H} = \left(\begin{array}{c|c} I_{n-k-\ell} & \mathbf{H}' \\ \hline \mathbf{0} & \mathbf{H}'' \end{array}\right) \in \mathbb{F}_q^{n\times(n-k)}$, where $\mathbf{H}' \in \mathbb{F}_q^{(k+\ell)\times(n-k-\ell)}$ and $\mathbf{H}'' \in \mathbb{F}_q^{(k+\ell)\times\ell}$, and here once again we deal with the subproblem DP on matrix $\mathbf{H}''$ instead of $\mathbf{H}$.

Let us define the following lists. Note that the list L_2 does not need to be classically written at any moment of the algorithm.

$$\begin{aligned} E_1 &:= \{(\boldsymbol{x}_1 \parallel \mathbf{0}^{\frac{2(k+\ell)}{3}}) \mid \boldsymbol{x}_1 \in \mathbb{F}_3^{\frac{k+\ell}{3}}, |\boldsymbol{x}_1| = p/3\} \ ; \ L_1 := \{(\boldsymbol{e}_1'', \boldsymbol{e}_1''\mathbf{H}''^{\top}) : \boldsymbol{e}_1'' \in E_1\} \\ E_2 &:= \{(\mathbf{0}^{\frac{k+\ell}{3}} \parallel \boldsymbol{x}_2) \mid \boldsymbol{x}_2 \in \mathbb{F}_3^{\frac{2(k+\ell)}{3}}, |\boldsymbol{x}_2| = 2p/3\} \ ; \ L_2 := \{(\boldsymbol{e}_2'', \boldsymbol{e}_2''\mathbf{H}''^{\top}) : \boldsymbol{e}_2'' \in E_2\} \end{aligned} \tag{12}$$

The lists are no longer of equal size. Indeed, to balance the running times, the list in quantum superposition is taken quadratically larger than the classical one:

$$|L_1| = \widetilde{\mathcal{O}}\left(\binom{k+\ell}{p}^{1/3} 2^{p/3}\right) \quad \text{and} \quad |L_2| = |L_1|^2. \tag{13}$$

The algorithm starts by classically constructing the list L_1. It also constructs the uniform quantum superposition over the elements of L_2: $|\psi_{L_2}\rangle = \frac{1}{\sqrt{|L_2|}} \sum_{(e_2, \boldsymbol{y}_2) \in L_2} \left|\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\right\rangle |\boldsymbol{e}_2\rangle |\boldsymbol{y}_2\rangle$, where $\mathrm{ind}_{L_2}(\boldsymbol{e}_2)$ is the index of the tuple $(\boldsymbol{e}_2, \boldsymbol{y})$ in the list L_2. We apply a quantum merging (Lemma 3) on L_1 and $|\psi_{L_2}\rangle$ to get the state $\left|\psi_{L_1 \bowtie_\ell L_2}\right\rangle$, which contains the quantum superposition of all the $\boldsymbol{e}'' = (\boldsymbol{e}_1'' + \boldsymbol{e}_2'')$ for $\boldsymbol{e}_1'' \in E_1$ and $\boldsymbol{e}_2'' \in E_2$ such that $\boldsymbol{e}''\mathbf{H}''^{\top} = \mathbf{0}$. Please refer to Equation (6) for an explicit expression of this quantum state. As by construction $\boldsymbol{e}_1'' \in E_1$ and $\boldsymbol{e}_1'' \in E_1$ have disjoint set of non-zero coordinates, then $\boldsymbol{e}''$ is of weight $|\boldsymbol{e}''| = |\boldsymbol{e}_1''| + |\boldsymbol{e}_2''| = p/3 + 2p/3 = p$. So we end up with a quantum superposition of $|L_1 \bowtie_\ell L_2|$ solutions to the $\mathsf{DP}_{\mathbf{H}'', \mathbf{0}, p}$ subproblem. Using this as a subroutine within the ISD framework leads to the following theorem.

Theorem 7. *Let us fix parameters ℓ, p, t and set $k := k_U + k_V$. There exists a quantum algorithm under the QRAM model assumption that solves DWK_{n,k_U,k_V} in time*

$$T = \max\left\{ \binom{k+\ell}{p}^{1/2} 2^{p/2}, \frac{3^{n/4 - k_U/2}\binom{n}{t}^{1/4}}{2^{t/4 - p/2}\binom{n-k-\ell}{t-p}^{1/2}} \right\}.$$

Proof. Sizes of lists L_1 and L_2 are given in Equation (13). Constructing the initial classical list takes time $|L_1|$, and constructing the initial quantum state $|\psi_{L_2}\rangle$ can be done in efficient time by a Quantum Fourier Transform and then applying the circuit $\left|\boldsymbol{e}_2''\right\rangle |0\rangle \mapsto \left|\boldsymbol{e}_2''\right\rangle \left|\boldsymbol{e}_2''\mathbf{H}''^{\top}\right\rangle$.

The quantum merging takes time $|L_1| + \sqrt{|L_2|}$ by Lemma 3. On average we can expect $|L_1 \bowtie_\ell L_2| = \binom{k+\ell}{p} 2^p \cdot 3^{-\ell} := N_{SolFound}$ by Equation (4). This is also equal, up to a polynomial factor, to $N_{Sol}(\mathsf{DP}_{\mathbf{H}'', \mathbf{0}, p})$ the number of solutions to the DP subproblem, by Proposition 1. And Proposition 2 gives the number of solutions to the DP problem in the $(U, U+V)$-code which is $N_{Sol}(\mathsf{DP}_{\mathbf{H}, \mathbf{0}, t}) = \binom{n/2}{t/2} \frac{2^{t/2}}{3^{n/2 - k_U}}$. Plugging these values into the Theorem 4 with the same notations provides the result. □

3.4 Numerical results

The time complexity is optimal when the list L_2 is of maximal size. Parameters ℓ and t are obtained by numerical optimization to minimize the time complexity of the attack. We give here the optimal ISD parameters and the associated time complexities for each set of parameters of Wave given in Table 3.

The time complexity is optimal for $t \approx 0.21n$, $\ell \approx 0.0052n$ and $p \approx 0.0024n$. The ISD algorithm with a quantum Dumer subroutine solves DWK_{n,k_U,k_V} for the set of Wave parameters (I) in time $2^{0.0094n + o(n)}$, *i.e.*, 2^{80} bits of quantum security; for the set (III) in time $2^{0.0096n + o(n)}$, *i.e.*, 2^{120}; and for the set (V) in time $2^{0.0097n + o(n)}$, *i.e.*, 2^{160}. These results are summarized in the second column of Table 1.

4 Message attacks

These attacks consist in forging a signature by solving the problem $\mathsf{DOOM}_{n,k,w}$ (Problem 3) that we remind here: given a list S of syndromes in $\mathbb{F}_3^{n-k}$ and a matrix $\mathbf{H}$, find $\boldsymbol{e} \in \mathbb{F}_3^n$ and $\mathbf{s} \in S$ such that $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$. Once again, we use the ISD framework, but here with Wagner's algorithm [Wag02] as a subroutine instead of just Dumer's [Dum91]. We first recall the classical message attack to introduce the useful notions and notations we will use in our quantum version of the attack.

4.1 Classical message attack

The best known classical message attack algorithm is the smoothed Wagner's algorithm [MS12, BCDAL20] based on the approach of [Sen11] to solve DOOM. A parameter a, to be chosen, will be the tree depth of Wagner's algorithm. Wagner's algorithm [Wag02] can be seen as a generalization of Dumer [Dum91], where taking Wagner with $a = 1$ exactly describes Dumer's algorithm.

First lists.

We start by constructing the first-level lists $L_1^{(0)}, \dots, L_{2^a-1}^{(0)}$ of size $|L_i^{(0)}| = 3^{\ell/a}$, where for $i = 1$ to $2^a - 1$ we sample

$$E_i^{(0)} \subseteq \left\{ (\mathbf{0}^{\frac{k+\ell}{2^a-1}} \| ... \| \mathbf{0}^{\frac{k+\ell}{2^a-1}} \| \underbrace{\boldsymbol{x}}_{i\text{th block}} \| \mathbf{0}^{\frac{k+\ell}{2^a-1}} \| ... \| \mathbf{0}^{\frac{k+\ell}{2^a-1}}) \mid \boldsymbol{x} \in \mathbb{F}_3^{\frac{k+\ell}{2^a-1}}, |\boldsymbol{x}| = \frac{p}{2^a-1} \right\}.$$

$$L_i^{(0)} := \left\{ ((\boldsymbol{e}'', \mathbf{0}), \boldsymbol{e}''\mathbf{H}''^\top) : \boldsymbol{e}'' \in E_i^{(0)} \right\} \tag{14}$$

And with the DOOM approach, the last list is filled with $3^{\ell/a}$ syndromes restricted on their ℓ last coordinates

$$L_{2^a}^{(0)} \subseteq \left\{ ((\mathbf{0}, \mathbf{s}''), -\mathbf{s}'') : \mathbf{s} = (\mathbf{s}' \| \mathbf{s}'') \in S, \mathbf{s}' \in \mathbb{F}_3^{n-k-\ell}, \mathbf{s}'' \in \mathbb{F}_3^\ell \right\}. \tag{15}$$

The aim to store elements in the form $((\boldsymbol{e}'', \mathbf{s}'), \boldsymbol{e}'\mathbf{H}^\top - \mathbf{s}'')$ is to merge them on their last elements and get at the end some for which $\boldsymbol{e}''\mathbf{H}^\top - \mathbf{s}'' = \mathbf{0}$ and be able to recover the corresponding $\boldsymbol{e}''$ and $\mathbf{s}''$. To be formal, let us precise the tuple addition $(\boldsymbol{e}_1, \mathbf{s}_1) + (\boldsymbol{e}_2, \mathbf{s}_2) := (\boldsymbol{e}_1 + \boldsymbol{e}_2, \mathbf{s}_1 + \mathbf{s}_2)$.

Notice that we need the list size to be lower than the number of words of weight p we can generate from $\mathbb{F}_3^{\frac{k+\ell}{2^a-1}}$, *i.e.*, we require

$$3^{\ell/a} \leq \binom{(k+\ell)/(2^a-1)}{p/(2^a-1)} 2^{p/(2^a-1)}.$$

Actually, as [BCDAL20] has already suggested and that we recover in our numerical optimizations, the optimal choice for p in high weight w is to take it at the maximum, *i.e.*, $p = k + \ell$. Then by rewriting the condition on a with this value of p gives this simplified formula:

$$3^{\ell/a} \leq 2^{\frac{k+\ell}{2^a-1}}. \tag{16}$$

Merging tree.

For Wagner's algorithm, we consider a binary merging tree with at the first level the lists $L_1^{(0)}, \ldots, L_{2^a-1}^{(0)}$ and $L_{2^a}^{(0)}$ defined in Equations 14 and 15. To pass from the j-th level to the $(j+1)$-th we merge pairwise lists using Lemma 2, for odd i:

$$L_{(i+1)/2}^{(j+1)} := L_i^{(j)} \bowtie_{(j+1)\ell/a} L_{i+1}^{(j)} \tag{17}$$

By construction, every $((\boldsymbol{e}'', \mathbf{s}''), \boldsymbol{y}) \in L_{(i+1)/2}^{(j)}$ will satisfy $\boldsymbol{y} = \boldsymbol{e}''\mathbf{H}''^{\top} - \mathbf{s}''$ and $|\boldsymbol{e}''| = j\frac{p}{2^a-1}$.

At each floor, the size of this newly merged list is $|L_{(i+1)/2}^{(j+1)}| = \frac{|L_i^{(j)}| \cdot |L_{i+1}^{(j)}|}{3^{\ell/a}}$, so by recurrence it remains constant at $3^{\ell/a}$ on average. Please refer to Figure 2 to visualize the merging process.

At the end of Wagner's algorithm, we get a final list $L_1^{(a)}$ filled with tuples in form $(\boldsymbol{e}'', \mathbf{s}'', \boldsymbol{e}''\mathbf{H}^{\top} - \mathbf{s}'' = \mathbf{0})$, and by construction, we have $|\boldsymbol{e}''| = p$. At each level in the merging tree, the list sizes are $3^{\ell/a}$ on average, so at the end we find as many solutions $(\boldsymbol{e}'', \mathbf{s}'')$ to the $\mathsf{DP}_{\mathbf{H}'', \mathbf{s}'', p}$ subproblem.

Algorithm 3 Classical Wagner's algorithm for DOOM [BCDAL20]

Input: $\mathbf{H}'' \in \mathbb{F}_3^{(k+\ell)\times\ell}$, a list S of target syndromes $\mathbf{s}''_1, \ldots, \mathbf{s}''_{3^{\ell/a}} \in \mathbb{F}_3^{\ell}$ length ℓ, target weight p, tree depth a.
Output: List of $(\boldsymbol{e}'', \mathbf{s}'') \in \mathbb{F}_3^{k+\ell} \times S$ such that $|\boldsymbol{e}''| = p$ and $\boldsymbol{e}''\mathbf{H}''^{\top} = \mathbf{s}''$

```
Sample lists $E_i^{(0)}$, $L_i^{(0)}$ for $i = 1$ to $2^a$ using Equation (14).
for $j = 0$ to $a - 1$ do
    for $i = 1$ to $2^{(a-j)}$ do
        Merge $L_{(i+1)/2}^{(j+1)} = L_i^{(j)} \bowtie_{(j+1)\ell/a} L_{i+1}^{(j)}$.
return $L_1^{(a)}$
```

Proposition 3 ([BCDAL20])**.** *Let us fix parameters ℓ, p, a such that $3^{\ell/a} \leq 2^{\frac{k+\ell}{2^a-1}}$ (See Equation (16)). There exists a classical algorithm that solves $\mathsf{DOOM}_{n,k,w}$ in time*

$$T = \max\left\{ 3^{\ell/a}, \frac{3^{n-k-\ell}}{2^{w-p}\binom{n-k-\ell}{w-p}} \right\}.$$

Proof. By Lemma 2, each merging step takes time $3^{\ell/a}$, thus Wagner's subroutine 3 takes time $T_{\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}} = 3^{\ell/a}$ to find $N_{SolFound} = 3^{\ell/a}$ solutions. By Proposition 1, the solutions to the DP problem are at number $N_{Sol}(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) = \binom{n}{w}\frac{2^w}{3^{n-k}}$, and the solutions to the subproblem are at number $N_{Sol}(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}) = \binom{k+\ell}{p}2^p$. We apply the classical ISD algorithm 1 with a Wagner subroutine, and the Theorem 3 with these values directly conducts to the result. □

Smoothing.

The discreteness of integer parameter a makes the time complexity of Wagner's algorithm evolve by stairs, which is not optimal for the majority of its points. [MS12] introduced a

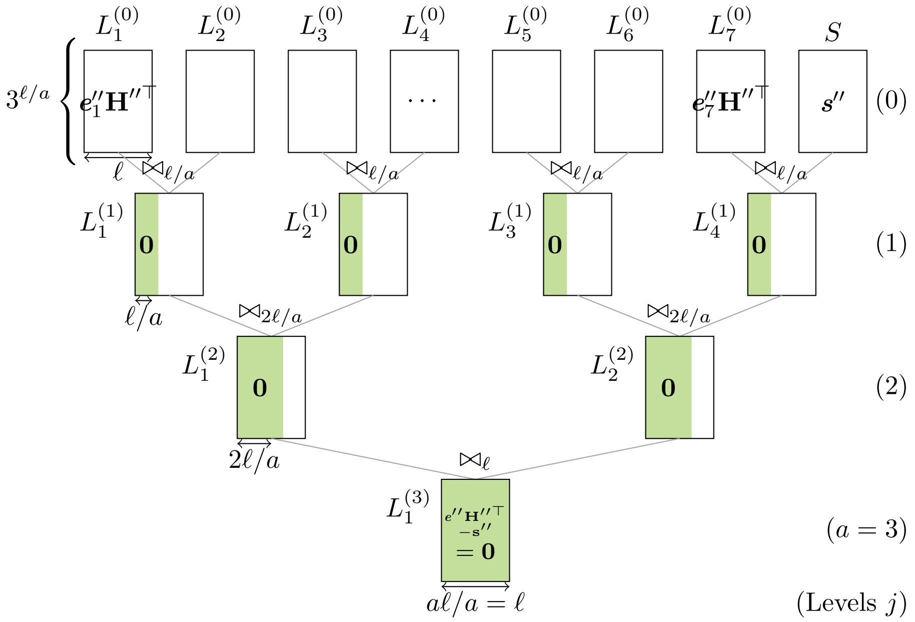

Figure 2: Wagner subroutine for $a = 3$. There are $2^a - 1 = 7$ initial lists of $\boldsymbol{e}''\mathbf{H}''^\top$, plus the syndromes list S. At each level, the lists are merged on ℓ/a more coordinates, until the final list $L_1^{(a)}$ filled with elements in the form $((\boldsymbol{e}'', \mathbf{s}''), \mathbf{0})$, where pairs $(\boldsymbol{e}'', \mathbf{s}'')$ are solutions to the $\mathsf{DOOM}_{k+\ell,k,p}$ subproblem.

smoothed binary Wagner algorithm, whose idea is to start with longer lists and a stricter first merging, and [BCDAL20] then extended the idea into the ternary setting. The lists $L_i^{(0)}$ are merged pairwise on m bits such that these merged lists are of size 2^λ for well-chosen m and λ. From there we merge on $\lambda/\log_2 3$ more coordinates at each level, until merging on all the ℓ coordinates.

Algorithm 4 Classical smoothed Wagner's algorithm for DOOM [BCDAL20]

Input: $\mathbf{H}'' \in \mathbb{F}_3^{(k+\ell)\times\ell}$, target syndromes $\mathbf{s}''_1, \ldots, \mathbf{s}_{3^{\ell/a}} \in \mathbb{F}_3^\ell$
length ℓ, target weight p, tree depth a.
Output: List of $(\boldsymbol{e}'', \mathbf{s}'') \in \mathbb{F}_3^{k+\ell} \times S$ such that $|\boldsymbol{e}''| = p$ and $\boldsymbol{e}''\mathbf{H}''^\top = \mathbf{s}''$

```
Compute $\lambda$ and $m$ using Equations 18 and 20.
Sample lists $E_i^{(0)}$, $L_i^{(0)}$ for $i = 1$ to $2^a - 1$, and $L_{2^a}^{(0)}$ using Equation (14).
for $i = 1$ to $2^a$ do
    Merge $L_{(i+1)/2}^{(1)} = L_i^{(0)} \bowtie_m L_{i+1}^{(0)}$
for $j = 1$ to $a - 1$ do
    for $i = 1$ to $2^{(a-j)}$ do
        Merge $L_{(i+1)/2}^{(j+1)} = L_i^{(j)} \bowtie_{m+j\frac{\lambda}{\log_2(3)}} L_{i+1}^{(j)}$
return $L_1^{(a)}$
```

Proposition 4 ([BCDAL20]). *Let a be the largest integer such that $3^{\ell/a} < 2^{(k+\ell)/(2^a-1)}$. If $a \geq 3$, the classical smoothed Wagner's algorithm can find 2^λ solutions to $\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}$ in time $\mathcal{O}(2^\lambda)$ with*

$$\lambda = \frac{1}{a-1}\left(\ell\log(3) - 2 \cdot \frac{k+\ell}{2^a - 1}\right). \tag{18}$$

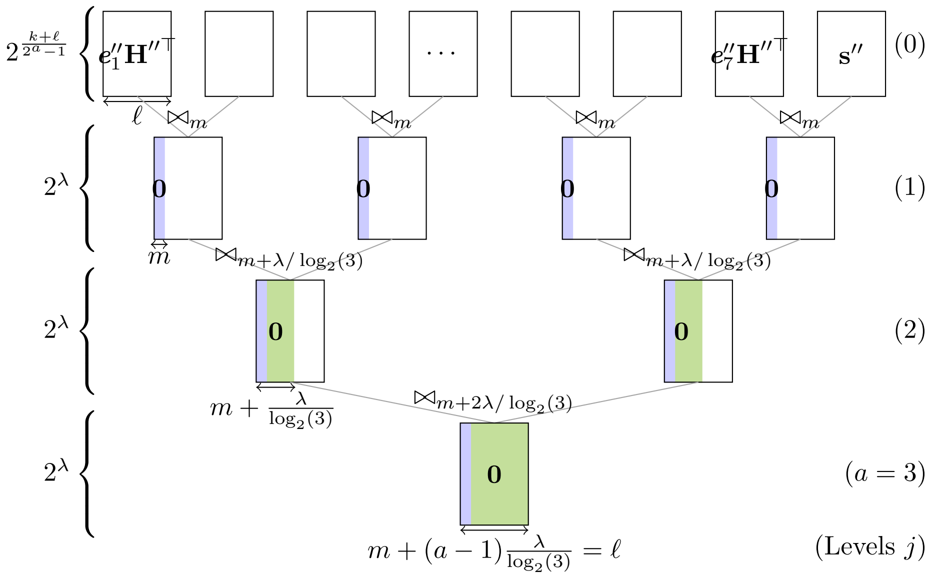

Figure 3: Smoothed Wagner subroutine for $a = 3$. The first merging is operated on a small number of coordinates m, and then we merge on $\lambda/\log_2 3$ more coordinates at each level.

Proof. We restate the proof from [BCDAL20] adapted in the context of DOOM (it only changes 2^a to $2^a - 1$ in the formulae).

We are given parameters k and ℓ, and we fix a at the largest integer such that $3^{\ell/a} < 2^{\frac{k+\ell}{2^a-1}}$ to respect the requirement stated in Equation (16), and we suppose that $a \geq 3$. At the first level in the tree, we take lists $L_i^{(0)}$ with the maximum possible size $|L_i^{(0)}| = 2^{\frac{k+\ell}{2^a-1}}$. We first merge on $m \leq \ell/a$ coordinates (Steps 2-4 in Algorithm 4). In order to obtain lists of size 2^λ at the second level, we have to choose m such that

$$\frac{\left(2^{\frac{k+\ell}{2^a-1}}\right)^2}{3^m} = 2^\lambda \quad i.e. \quad \lambda = \frac{2(k+\ell)}{2^a-1} - m\log_2 3 \tag{19}$$

The $(a-1)$ next merging steps are designed such that merging two lists of size 2^λ gives a new list of size 2^λ, which means that we merge on $\lambda/\log_2 3$ coordinates. In the final list, we have to put a constraint on all coordinates, therefore λ and m have to verify:

$$m + (a-1)\frac{\lambda}{\log_2 3} = \ell. \tag{20}$$

By combining Equations 19 and 20, We get the expression of λ as given in the statement of the proposition, and we deduce m from above. The order a is chosen to be the largest integer such that $3^{\ell/(a-1)} < 2^{\frac{k+\ell}{2^a-1}}$, so λ and deduce m are positive and $2^\lambda \leq 2^{\frac{k+\ell}{2^a-1}}$. □

Theorem 8 ([BCDAL20])**.** *There exists a classical algorithm that solves* $\mathsf{DOOM}_{n,k,w}$ *in time*

$$T = \max\left\{\left(\frac{3^\ell}{2^{\frac{k+\ell}{2^a-1}}}\right)^{\frac{1}{a-2}}, \frac{3^{n-k-\ell}}{2^{w-p}\binom{n-k-\ell}{w-p}}\right\}.$$

The left term in the max is improved in comparison with Proposition 3 for ISD with non-smoothed Wagner. This corresponds to the case of a single iteration of the ISD algorithm.

Proof. By Proposition 3, the classical smoothed Wagner's algorithm takes times $2^\lambda = \left(\frac{3^\ell}{2^{\frac{k+\ell}{2^{a-1}}}}\right)^{\frac{1}{a-2}}$. There are $N_{Sol}(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) = \binom{n}{w}\frac{2^w}{3^{n-k}}$ solutions for a random code. The time complexity of the ISD classical algorithm 1 with smoothed Wagner subroutine is given by Theorem 3 that directly conducts to the result. □

Numerical results.

As we said before, taking $p = k + \ell$ is optimal. Parameters ℓ and a are then chosen by numerical optimization. The optimal a in this setting is here $a = 5$ and $\ell \approx 0.05n$. The optimal values of ℓ may slightly vary as function of Wave parameters since they are not exactly linear.

Without smoothing. The ISD algorithm with Wagner's subroutine with the set of Wave parameters (I) solves DWK_{n,k_U,k_V} in time $2^{0.0153n+o(n)}$, *i.e.*, 2^{130}. For set (III) it solves it in time $2^{0.0156n+o(n)}$, *i.e.*, 2^{196}, and for set (V) in time $2^{0.0158n+o(n)}$, *i.e.*, 2^{261}.

With smoothing. The ISD algorithm with smoothed Wagner's subroutine for set (I) solves DWK_{n,k_U,k_V} in time $2^{0.0151n+o(n)}$, *i.e.*, 2^{129}. For set (III) it solves it in time $2^{0.0155n+o(n)}$, *i.e.*, 2^{194}, and for set (V) in time $2^{0.0157n+o(n)}$, *i.e.*, 2^{258}.

We see that the smoothing slightly improves the message attack on Wave and grabs a few security bits. The results with the smoothing are summarized in the third column of Table 1.

Previous work [FS09] considered the tree depth a as a float instead of an integer, in order to give a complexity approximation of a smoothed Wagner algorithm. When we optimize the time complexity of the non-smoothed Wagner from Proposition 3 with a allowed to be a float, the difference is of only one or two security bits less in comparison with the analysis of the smoothed Wagner algorithm from [BCDAL20]. Indeed, for set (I), the number of security bits is 128, for (III) it is 192, and for (V), 256. So considering a float a provides a tight lower bound in this setting.

4.2 Quantum message attack

We recall that given a quantumly accessible list L, $\mathrm{ind}_L(\boldsymbol{x})$ denotes the index of element $\boldsymbol{x}$ in the list L. The state $|\psi_L\rangle = \frac{1}{\sqrt{|L|}}\sum_{\boldsymbol{x}\in L}\left|\mathrm{ind}_L(\boldsymbol{x})\right\rangle|\boldsymbol{x}\rangle$ denotes the quantum superposition of the elements of list L (See Section 2.1).

For the quantum message attack, we combine DOOM approach from [Sen11], quantum Wagner's algorithm[3] of [CDAE21] and smoothing from [BCDAL20]. The merging tree has the same structure as in the smoothed classical algorithm presented in the previous section. Quantum mergings (see Lemma 3) are performed on the right-most side of the tree, and classical mergings (see Lemma 2) are performed everywhere else.

Proposition 5. *We are given n, k, w, and we fix parameters ℓ, p and a such that $3^{\ell/a} \leq 2^{\frac{k+\ell}{2^{a-1}}}$. There exists a quantum algorithm that solves $\mathsf{DOOM}_{n,k,w}$ in time*

$$T = \max\left\{3^{\ell/a}, \sqrt{\frac{3^{n-k-\ell}}{2^{w-p}\binom{n-k-\ell}{w-p}}}\right\}.$$

[3] [BBSS20b] presented a binary quantum Wagner's algorithm in the context of the subset sum problem. The version introduced in [CDAE21] tackles the syndrome decoding problem with Lee metric in $\mathbb{F}_q$, which is equal to Hamming metric in the case $q = 3$.

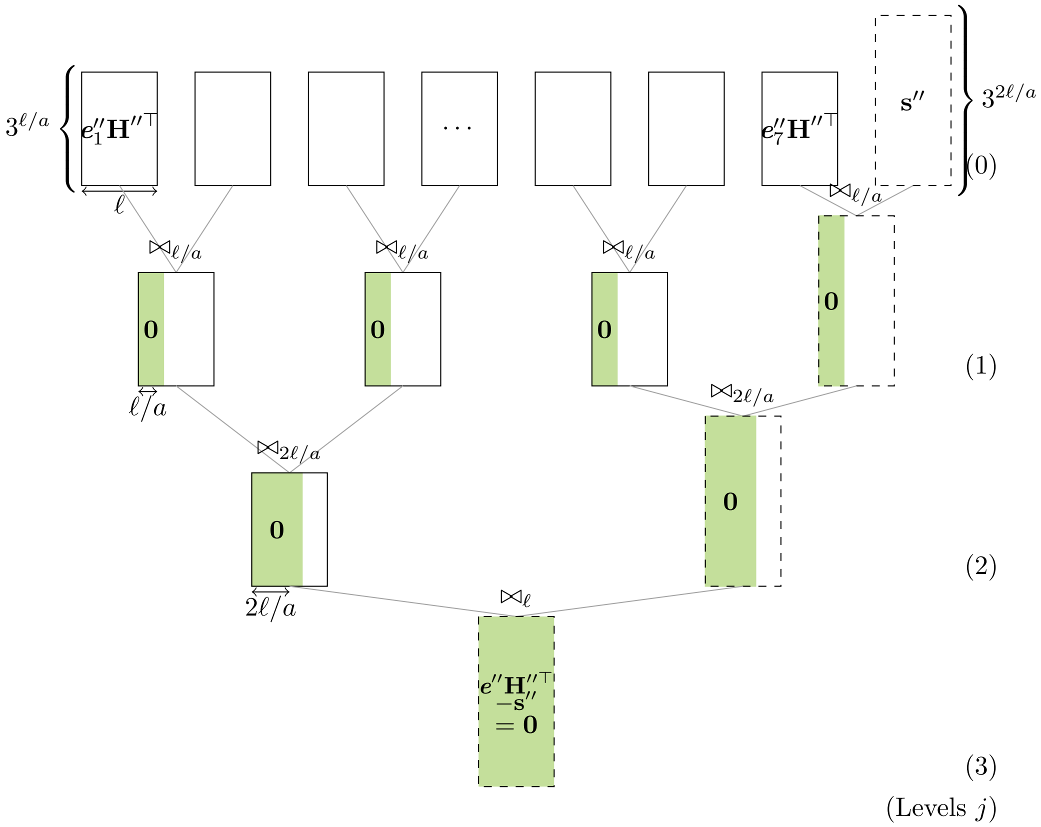

Figure 4: Quantum Wagner subroutine. Dashed-line boxes represent lists that are not classically constructed but of which we have an algorithm that constructs a quantum superposition of the elements.

Proof. Quantum Wagner's algorithm does the classical merges in time $3^{\ell/a}$, and the quantum ones in time $\frac{3^{\ell/a}\times\sqrt{3^{2\ell/a}}}{3^{\ell/a}} = 3^{\ell/a}$. So the whole Wagner's algorithm takes time $T_{\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}} = 3^{\ell/a}$ Proposition 1 gives the number of solutions to the DP problem $N_{Sol}(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) = \binom{n}{w}\frac{2^w}{3^{n-k}}$, and to the DP subproblem $N_{Sol}(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}) = \binom{k+\ell}{p} 2^p \cdot 3^{-\ell}$. Applying Theorem 4 with these amounts gives the time complexity of the ISD algorithm with Wagner's algorithm as a subroutine, and this directly leads to the result. □

Algorithm 5 Quantum smoothed Wagner's algorithm for DOOM

Input: $\mathbf{H}'' \in \mathbb{F}_3^{(k+\ell)\times\ell}$, list S of target syndromes in $\mathbb{F}_3^{\ell}$
length ℓ, target weight p, tree depth a.
Output: List in quantum superposition of $(\boldsymbol{e}'', \mathbf{s}'') \in \mathbb{F}_3^{k+\ell} \times S$ such that $|\boldsymbol{e}''| = p$ and $\boldsymbol{e}''\mathbf{H}''^{\top} = \mathbf{s}''$

```
Compute $\lambda$ and $m$ using Theorem 9.
Sample lists $L_i^{(0)}$ for $i = 1$ to $2^a - 1$ using Equations 14 and 15.
Construct state $\left|\psi_{L_{2^a}^{(0)}}\right\rangle$, quantum superposition of $3^{2\ell/a}$ syndromes $\mathbf{s}'' \in \mathbb{F}_3^{\ell}$
for $i = 1$ to $2^a - 2$ do
  Classically merge $L_{(i+1)/2}^{(1)} = L_i^{(0)} \bowtie_m L_{i+1}^{(0)}$
Quantumly merge $L_{2^a-1}^{(1)} = L_{2^a-1}^{(0)} \bowtie_m L_{2^a}^{(0)}$
for $j = 1$ to $a - 1$ do
  for $i = 1$ to $2^{(a-j)} - 1$ do
    Classically merge $L_{(i+1)/2}^{(j+1)} = L_i^{(j)} \bowtie_{m+j\frac{\lambda}{\log_2(3)}} L_{i+1}^{(j)}$
  Quantumly merge $L_{(2^{a-j}+1)/2}^{(j+1)} = L_{2^{a-j}-1}^{(j)} \bowtie_{m+j\frac{\lambda}{\log_2(3)}} L_{2^{a-j}}^{(j)}$
return $\left|\psi_{L_1^{(a)}}\right\rangle$
```

Theorem 9. *We are given n, k, w, and we fix parameters ℓ, p and $a \geq 3$ such that $3^{\ell/a} \leq 2^{\frac{k+\ell}{2^a-1}}$. There exists a quantum algorithm that solves $\mathsf{DOOM}_{n,k,w}$ in time*

$$T = \max\left\{ \left(\frac{3^{\ell}}{2^{\frac{k+\ell}{2^a-1}}}\right)^{\frac{1}{a-2}}, \sqrt{\frac{3^{n-k-\ell}}{2^{w-p}\binom{n-k-\ell}{w-p}}} \right\}.$$

Proof. The logic is the same as in the classical smoothed Wagner algorithm, but the optimal list sizes obey a different balance. The order a is chosen at the largest integer such that $3^{\ell/a} < 2^{\frac{k+\ell}{2^a-1}}$ to respect the condition set in the Equation (16). The classical lists $L_i^{(0)}$ for $i = 1$ to $2^a - 1$ are chosen of maximal size $2^{\frac{k+\ell}{2^a-1}} =: 2^{\gamma}$. The list $L_{2^a}^{(0)}$ in quantum superposition is of size $2^{\gamma'}$. The classical list $L_i^{(j)}$ for $j > 0$ and $0 \leq i < 2^a$ are of size 2^{λ}, and the quantum list for levels $(j > 0)$ are of size $2^{2\lambda}$. The classical merging from level (0) to level (1) is done on the m first elements.

Now we need to choose γ, γ', λ and m. The classical merging from level (0) to level (1) puts the constraint $\lambda = 2\gamma - m\log_2 3$. And the quantum merging requires $2\lambda = \gamma' + \gamma - m\log_2 3$. So we can deduce from the above that $\lambda = \gamma' - \gamma$. For the classical part of the merging tree for level (1) and more, the constraints on λ and m remain the same as in Proposition 4 so their expressions are already given (respectively) by Equations 18 and 20, where $\lambda = \frac{1}{a-2}\left(\ell\log(3) - \frac{2(k+\ell)}{2^a-1}\right)$.

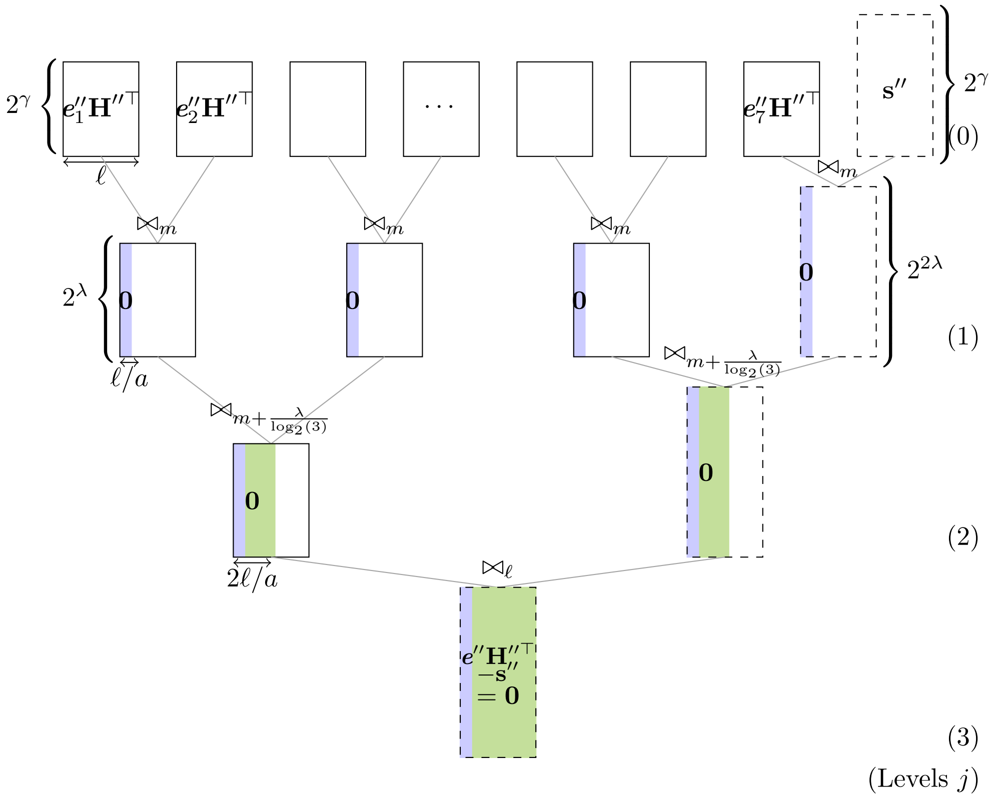

Figure 5: Quantum smoothed Wagner subroutine. Dashed-line boxes represent lists that are not classically constructed but of which we have an algorithm that constructs a quantum superposition of the elements.

The first-level classical merges take time 2^γ, the first quantum merges take time $2^{\max\{\gamma, \frac{\lambda+\gamma}{2}\}}$, and all the other merges take time 2^λ, which dominates as $\lambda \geq \gamma$. So the quantum smoothed Wagner subroutine takes time $T_{\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}} = 2^\lambda$ to construct a list in quantum superposition with 2^{2^λ} solutions to the $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$ subproblem.

Proposition 1 gives the number of solutions to the DP problem $N_{Sol}(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) = \binom{n}{w}\frac{2^w}{3^{n-k}}$, and to the DP subproblem $N_{Sol}(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}) = \binom{k+\ell}{p} 2^p \cdot 3^{-\ell}$. Theorem 4 with these amounts gives the time complexity of the quantum ISD algorithm with smoothed Wagner algorithm as a subroutine, and this directly leads to the result. □

Numerical results.

As said before, taking $p = k + \ell$ is optimal. Parameters ℓ and a are then chosen by numerical optimization.

Without smoothing. Taking $\ell \approx 0.032n$ and $a = 6$ is optimal. The quantum ISD algorithm with quantum Wagner's subroutine with the set of Wave parameters (I) solves DWK_{n,k_U,k_V} in time $2^{0.0093n+o(n)}$, *i.e.*, 2^{79}. For set (III) it solves it in time $2^{0.0096n+o(n)}$, *i.e.*, 2^{120}, and for set (V) in time $2^{0.0098n+o(n)}$, *i.e.*, 2^{161}.

With smoothing. Taking $\ell \approx 0.034n$ and $a = 6$ is optimal. The quantum ISD algorithm with smoothed quantum Wagner's subroutine with the set of Wave parameters (I) solves DWK_{n,k_U,k_V} in time $2^{0.0091n+o(n)}$, *i.e.*, 2^{78}. For set (III) it solves it in time $2^{0.0094n+o(n)}$, *i.e.*, 2^{117}, and for set (V) in 2^{156}.

These results are summarized in the fourth column of Table 1, and below with the comparison of claimed security from previous works.

Table 4: Number of quantum security bits for message attacks.

Algorithm	(I)	(III)	(V)
ISD + Wagner [CDAE21]	79	120	161
Estimation [BCC+23]	77	117	157
ISD + smoothed Wagner (This work, Thm. 9)	78	117	156

We see that quantum Wagner's algorithm benefits from smoothing, by respectively decreasing the security by respectively 1, 3 and 5 security bits for sets (I), (III) and (V). And we correct the claimed quantum security level of [BCC+23], whose estimation did not rely on analyzing an explicitly described algorithm, which is now formalized by our theorem 9. Their estimation only differs by plus or minus one security bit, and the slight underestimation in case (V) maintains the security level far from the required threshold set at 128 security bits.

5 Security in NIST model

List-merging overhead. Let us briefly complete our analysis by quantifying the polynomial overheads due to list merging (See Section 2.4). Classically merging two lists of size $|L|$ has a polynomial overhead $\log_2(|L|) \cdot n$ factor. A similar overhead arises when quantumly merging a classical list of size $|L|$ with a list in quantum superposition of size $|L|^2$ elements. In the context of our attack algorithms, the list sizes $|L|$ can be found in Equation 9 and 13 for key-distinguishing attacks; and we have $|L| = 2^\lambda$ where λ is defined in Equation 18 for message attacks. Moreover, the smoothed Wagner algorithm with tree depth a performs 2^a list merges at the first level, and $\sum_{i=0}^{a-1} 2^i = 2^a - 1$ merges for all the rest of the merging tree. Adding these overheads into the security bit count results to the values reported in Table 2 in the introduction.

NIST security levels. The NIST defines their own measure of security for their post-quantum cryptography standardization process [NIS]. Under this model, a cryptographic scheme validates a given level if no (known) attack can be performed with fewer logical gates than required to break AES-λ for that level. In particular, the NIST fix different quantum settings that put a constraint on the maximal depth allowed in quantum attacks, denoted as 'MAXDEPTH'.

Table 5: NIST gates thresholds for classical and quantum security levels. Numbers in parentheses correspond to $\log_2$(MAXDEPTH).

Levels		Classical	Quantum		
			(96)	(64)	(40)
(I)	AES-128	143	74	106	130
(III)	AES-192	207	137	169	193
(V)	AES-256	272	202	234	258

To determine whether a Wave parameter set satisfies a NIST level, one compares the estimated time complexity of the best known attacks (Table 2 in the introduction) against the thresholds from (Table 5), for each respective setting.

The parameters proposed in [Sen23] were initially selected to reach the classical thresholds under the asymptotic security model. We observe that the scheme also validates the thresholds in the security NIST model.

Our quantum security analyses did not consider depth constraints, as we aim for conservativity of the results. Indeed, 'realistic' quantum circuit depth and other technical constraints are aimed to change with time. Nonetheless, we observe that the attacks costs with an attacker that disposes of theoretical unlimited depth is already above the required threshold for MAXDEPTH at 2^{96} for levels (I) and (III). However, for level (V), our key-distinguishing attack runs in time 2^{193}, and our message attack runs in time 2^{182}, while the threshold is at 2^{202} for MAXDEPTH at 2^{96}. As an indication, the set of parameters $n = 18504$, $k = 9252$, $w = 16568$, $k_U = 6392$, and $k_V = 2860$ ensures both quantum attacks to be above the threshold. This increases the dimension by 12%, which would impact quadratically the key size, and linearly the signature size. It is unclear if in practice, the original parameter selection would be reached if we add depth constraint and all the small overheads of each operation.

Restricting the depth means shrinking also the list sizes, and the performance of quantum attacks using Grover's search heavily relies on the size of the searching spaces. For levels (I) and (III) parameter selections, the large margin between the attack costs and the classical thresholds makes think that having low-depth Grover's search would not make the security number pass below the threshold in the low depth setting. Proving this intuition would require a deeper analysis of depth-restricted variants of the algorithms in this specific model, which would be relevant in case Wave is submitted again to the NIST standardization process.

Furthermore, note that in our analyzes we assume efficient QRAM, while in practice it could be way more costly to implement.

Leads for further research

It is still an open problem to determine if there exists a better key-distinguishing attack that uses the structure of the $(U, U + V)$-code, potentially by avoiding going through the

Decoding Problem. For the message quantum attack, the merging tree could be optimized using the results of [NPS20] by solving a k-sum problem in $\mathbb{F}_3$. There may also exist a solution to smooth the trade-off between key and message attack complexities to refine the choice of parameters and, in doing so, to gain on the key size marginally. Finally, one could analyze variants of the algorithms with restrictions on depth or QRAM.

References

[ABB+21] N Aragon, P. L. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Ghosh, S. Gueron, T. GÃ¼neysu, C. Aguilar Melchor, R. Misoczki, E. Persichetti, J. Richter-Brockmann, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. ZÃ©mor. Bike - bit flipping key encapsulation. *NIST PQC*, 2021. URL: https://bikesuite.org/files/v5.0/BIKE_Spec.2022.10.10.1.pdf.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 298–307. IEEE, 2003.

[BBSS20a] Xavier Bonnetain, Rémi Bricout, André Schrottenloher, and Yixin Shen. Improved classical and quantum algorithms for subset-sum. In *ASIACRYPT*, 2020. doi:10.1007/978-3-030-64834-3_22.

[BBSS20b] Xavier Bonnetain, Rémi Bricout, André Schrottenloher, and Yixin Shen. Improved classical and quantum algorithms for subset-sum. In *ASIACRYPT*, 2020. URL: https://doi.org/10.1007/978-3-030-64834-3_22.

[BCC+23] Gustavo Banegas, KÃ©vin Carrier, André Chailloux, Alain Couvreur, Thomas Debris-Alazard, Philippe Gaborit, Pierre Karpman, Johanna Loyer, Ruben Niederhagen, Nicolas Sendrier, Benjamin Smith, and Jean-Pierre Tillich. Wave support documentation. *NIST PQC*, 2023. URL: https://wave-sign.org/wave_documentation.pdf.

[BCDAL20] RÃ©mi Bricout, André Chailloux, Thomas Debris-Alazard, and Matthieu Lequesne. Ternary syndrome decoding with large weight. *SAC*, 2020. URL: https://arxiv.org/pdf/1903.07464.pdf.

[BDANS21] Gustavo Banegas, Thomas Debris-Alazard, Milena Nedeljkovi, and Benjamin Smith. Wavelet: Code-based postquantum signatures with fast verification on microcontrollers. *Cryptology ePrint Archive*, 2021. URL: https://ia.cr/2021/1432.

[Ber10] Daniel J. Bernstein. Grover vs. mceliece. *PQCrypto*, 2010. URL: https://cr.yp.to/codes/grovercode-20100303.pdf.

[BHMT02] Gilles Brassard, Peter HÃ¸yer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *QCQI*, page 305:53â€“74, 2002. URL: https://arxiv.org/abs/quant-ph/0005055.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1+1=0$ improves information set decoding. In *EUROCRYPT*, Lecture Notes in Computer Science. Springer, 2012. URL: https://eprint.iacr.org/2012/026.pdf.

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: ball-collision decoding. In *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 743–760, 2011. URL: https://eprint.iacr.org/2010/585.pdf.

[BM17] Leif Both and Alexander May. Optimizing BJMM with Nearest Neighbors: Full Decoding in $2^{2/21n}$ and McEliece Security. In *WCC*, September 2017. URL: http://wcc2017.suai.ru/Proceedings{_}WCC2017.zip.

[BM18] 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, *PQCrypto*, volume 10786 of *Lecture Notes in Computer Science*, pages 25–46, Fort Lauderdale, FL, USA, April 2018. Springer. doi:10.1007/978-3-319-79063-3{_}2.

[BMvT78] E. R. Berlekamp, R. J. McEliece, and H. C. van Tilborg. On the inherent intractability of certain coding problems. *IEEE*, 24, no.3, 1978. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1055873.

[CDAE21] André Chailloux, Thomas Debris-Alazard, and Simona Etinski. Classical and quantum algorithms for generic syndrome decoding problems and applications to the lee metric. *PQCrypto*, 2021. URL: https://arxiv.org/pdf/2104.12810.pdf.

[CDMT22] Kevin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. Statistical decoding 2.0: Reducing decoding to LPN. In *ASIACRYPT*, Lecture Notes in Computer Science. Springer, 2022. URL: https://eprint.iacr.org/2022/1000.

[DA23] Thomas Debris-Alazard. Code-based cryptography: Lecture notes. 2023. URL: https://arxiv.org/pdf/2304.03541.pdf.

[DAST19] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. *ASIACRYPT*, 2019. URL: https://arxiv.org/pdf/1810.07554.pdf.

[Dum91] Ilya Dumer. On minimum distance decoding of linear codes. *Workshop Inform. Theory*, pages 50–52, 1991.

[FHK+18] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-basedcompact signatures over ntru. *NIST*, 2018. URL: https://www.di.ens.fr/~prest/Publications/falcon.pdf.

[FS09] Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In M. Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 88–105. Springer, 2009. URL: https://eprint.iacr.org/2009/414.pdf.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. ACM, 2008. URL: https://dl.acm.org/doi/pdf/10.1145/1374376.1374407.

[Gro96] Lov Grover. A fast quantum mechanical algorithm for database search. *STOC*, pages 212 – 219, 1996. URL: https://dl.acm.org/doi/pdf/10.1145/237814.237866.

[Jab01] Abdulrahman Al Jabri. A statistical decoding algorithm for general linear block codes. *LNCS*, 2001. URL: https://link.springer.com/content/pdf/10.1007/3-540-45325-3_1.pdf?pdf=inline%20link.

[JJ02] Thomas Johansson and Fredrik Jönsson. On the complexity of some cryptographic problems based on the general decoding problem. *IEEE Transactions on Information Theory*, 2002. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1035119.

[Kir18] Elena Kirshanova. Improved quantum information set decoding. In Tanja Lange and Rainer Steinwandt, editors, *PQCrypto*, volume 10786 of *Lecture Notes in Computer Science*, pages 507–527. Springer, 2018. doi:10.1007/978-3-319-79063-3_24.

[KL22] Pierre Karpman and Charlotte Lefevre. Time-memory tradeoffs for large-weight syndrome decoding in ternary codes. In *PKC*, 2022. URL: https://doi.org/10.1007/978-3-030-97121-2_4.

[KT17] Ghazal Kachigar and Jean-Pierre Tillich. Quantum information set decoding algorithms. In *PQCrypto*, 2017. URL: https://arxiv.org/pdf/1703.00263.pdf.

[Kup13] G. Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *TQC*, 2013. URL: https://arxiv.org/pdf/1112.3333.pdf.

[MAB$^+$] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jurjen Bos, Jean-Christophe Deneuville, Arnaud Dion, Philippe Gaborit, Jérôme Lacan, Edoardo Persichetti, Jean-Marc Robert, Pascal Véron, and Gilles Zémor. Hqc (hamming quasi-cyclic). *NIST PQC*.

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN*, 1978. URL: https://ntrs.nasa.gov/api/citations/19780016269/downloads/19780016269.pdf#page=123.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $O(2^{0.054n})$. In *ASIACRYPT*, 2011. URL: https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/paper/ac11_decoding.pdf.

[MO15] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In E. Oswald and M. Fischlin, editors, *EUROCRYPT*, volume 9056 of *Lecture Notes in Computer Science*, pages 203–228. Springer, 2015. URL: https://www.crypto.ruhr-uni-bochum.de/imperia/md/content/may/paper/codes.pdf.

[MS12] Lorenz Minder and Alistair Sinclair. The extended k-tree algorithm. *Journal of Cryptology*, 2012. URL: https://link.springer.com/content/pdf/10.1007/s00145-011-9097-y.pdf.

[NIS] NIST. Post-quantum cryptography evaluation criteria. *NIST*. URL: https://csrc.nist.rip/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-%28evaluation-criteria%29?

[NPS20] María Naya-Plasencia and André Schrottenloher. Optimal merging in quantum k-xor and k-sum algorithms. *EUROCRYPT*, 2020. URL: https://eprint.iacr.org/2019/501.pdf.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. IT*, 8(5):5–9, 1962. URL: http://dx.doi.org/10.1109/TIT.1962.1057777, doi:10.1109/TIT.1962.1057777.

[Sch22] André Schrottenloher. Improved quantum algorithms for the k-xor problem. *SAC*, 2022. URL: https://eprint.iacr.org/2021/407.pdf.

[Sen11] Nicolas Sendrier. Decoding one out of many. *PQCrypto*, 2011. URL: https://eprint.iacr.org/2011/367.pdf.

[Sen23] Nicolas Sendrier. Wave parameter selection. *PQCrypto*, 2023. URL: https://eprint.iacr.org/2023/588.pdf.

[Ste88] Jacques Stern. A method for finding codewords of small weight. In *Coding Theory*, 1988. URL: https://link.springer.com/chapter/10.1007/BFb0019850.

[Wag02] David Wagner. A generalized birthday problem. *CRYPTO*, 2002. URL: https://www.enseignement.polytechnique.fr/informatique/profs/Francois.Morain/Master1/Crypto/projects/Wagner02.pdf.