Full Cryptanalysis of Wave by the Information Set Decoding: classical and quantum security

Johanna Loyer
johanna.loyer@inria.fr

Inria Paris, EPI COSMIQ

Abstract. Wave is a code-based digital signature scheme. Its hardness relies on the unforgeability of 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, which improves the forgery attack on Wave in the quantum model. We also recap the best known key and forgery attacks against Wave in the classical and quantum models. For each one, we explicitly express their time complexity in the function of Wave parameters and deduce the claimed security of Wave.

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

1 Introduction

Wave [Ban+23] is a hash-and-sign digital signature scheme candidate for the NIST post-quantum standardization process. Introduced in [DST19], it instantiates the theoretical framework of [GPV08] with a new trapdoor based on coding theory instead of lattices as it was usually done before, for example in Falcon [Fou+18]. Wave has short signatures (less than 1 kilobyte for 128 security bits) and fast verification (less than one millisecond) [Ban+21]. But as a drawback, it has a very large public key (several megabytes).

Wave is provably EUF-CMA (existential unforgeability under adaptive chosen message attacks) under code-based hardness assumptions, namely the hardness of decoding and the indistinguishability of permuted generalized ternary $(U, U+V)$-codes. The best known method for an attacker to solve these two problems goes through solving the Decoding Problem (DP). This problems is as follows: Given input $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$ and w, DP asks to find an $\boldsymbol{e} \in \mathbb{F}_q^n$ of Hamming weight w such that $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$. This problem is NP-complete [BMT78] and its average case is believed to be hard both classically and quantumly, even after more than forty years of strong interest from the research community [Pra62; Ste88; Dum91; FS09; Ber10; BLP11; MMT11; Bec+12; MO15; BM17; KT17; BM18; Kir18; Bon+20; Car+22]. The Decoding Problem is at the foundation of many code-based cryptographic schemes, including McEliece [McE78], BIKE [Ara+21] or HQC [Mel+21] from the previous NIST call for post-quantum schemes. However, the Decoding Problem in $\mathbb{F}_q$ for $q > 2$ has been less studied. Specifically, DP with Wave settings, where $q = 3$ and for high weight w, has only got recent attention [Bri+20; CDE21; KL22; Sen23]. A main difference is that for $\mathbb{F}_2$, there is one peak of computational hardness when the Hamming weight is low, while in $\mathbb{F}_3$ a second peak appears for high weight.

The ways to defeat Wave fall into two categories. *Key attacks* aim to solve the DWK (Distinguishing Wave Keys) problem that asks to distinguish the public key of a permuted generalized ternary $(U, U+V)$-code from a uniformly random matrix. The best known key classical attack [Sen23] constructs a distinguisher which exhibits a word in the code with a certain weight, where such words are simpler to find in a $(U, U+V)$-code than in a random code. This attack uses the ISD (Information Set Decoding) framework [Pra62] with Dumer's algorithm [Dum91] as a subroutine. Before this present work, there was no quantum cryptanalysis specifically for the DWK problem to our knowledge.

The other way to attack Wave is by *message attacks*, which aim to solve the DOOM (Decoding One Out of Many) problem [Sen11], *i.e.* to forge a signature $(\boldsymbol{e}, \mathbf{s}) \in \mathbb{F}_3^n \times \mathbb{F}_3^{n-k}$ that satisfies $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$ and $\boldsymbol{e}$ of weight w. The best known classical attack [Bri+20] also uses an instance of the ISD [Pra62] with Wagner's algorithm [SS81] as a subroutine. The best quantum attack [CDE21] applies Wagner's algorithm in a quantum ISD.

[Sen23] computed the time trade-off between these two classical attacks and get optimized parameters of Wave for a given security level. These parameters were then updated in [Ban+23] to take into account the best classical message attack.

Contributions.

- We describe a quantum smoothed Wagner's algorithm based on the combined approaches of [Sen11], [Bri+20] and [CDE21]. Our new algorithm provides an improved message attack on Wave. The technical difficulty laid in sizing the additional smoothing level in Wagner's merging tree. Indeed, contrary to the classical, the optimal quantum algorithm does not have all its list sizes equal even at the same tree level.
- For each of the four best known attacks on Wave, we do a complete time complexity analysis and provide transparent expressions in the function of the Wave parameters. So the claimed security level can easily be updated with new sets of parameters using our formulas. We then apply our theorems to the Round-1 parameter selection and the results are summarized in Table 1.

Organisation 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 attacks based on ISD and Dumer's algorithm and Section 4 presents message attacks based on ISD and Wagner's algorithm. In Section 5, we conclude and comment on the obtained results.

Table 1. Security of Wave instances. λ bits of security indicate that the most efficient known attacks require a time 2^{λ} to execute. You may notice that λ does not equal the minimum required number of security bits and this will be explained in Section 5.

	Classical		Quantum	
Setting	Key attack	Message attack	Key attack	Message attack
	Thm. 5	Thm. 7	Thm. 6	Thm. 8
(I)	138	129	80	78
(III)	206	194	120	117
(V)	274	258	160	156

2 Preliminaries

Notations. $\mathbb{F}_q$ denotes a q-ary finite field. In this paper, we consider the case $q = 3$. Vectors are in raw 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 $|\boldsymbol{x}| := |\{x_i : x_i \neq 0\}|$. For a vector $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$, we denote by $\boldsymbol{x}_{|[\![i,j]\!]}$ the restricted vector $(x_i, \ldots, x_j)$. For a matrix $\mathbf{H}$ we denote by $\mathbf{H}^{\top}$ its transpose. For $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{F}_q^n$ and $\mathbf{M} = (M_{i,j})_{0 \le i < r, 0 \le j < n-1} \in \mathbb{F}_q^{r \times n}$, we define $\boldsymbol{x} \star \mathbf{M} := (x_j M_{i,j})_{0 \le i < r, 0 \le j < n}$ the row-wise star product. We use the notation $\widetilde{\mathcal{O}}$ to denote complexities $\widetilde{\mathcal{O}}\left(2^{cn}\right)$ which ignores sub-exponential factors in 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 Memory (QRAM) operation can be efficiently implemented. If we consider registers $x_1, ..., x_N \in \{0,1\}^n$ classically stored, a QRAM operation consists in applying the unitary $|i\rangle |y\rangle \to |i\rangle |x_i \oplus y\rangle$.

Definition 1 (Quantum superposition). *Given a list L, we call the quantum superposition of L the state $|\psi_L\rangle := \frac{1}{\sqrt{|L|}} \sum_{x \in L} |\mathrm{ind}_L(x)\rangle |x\rangle$, where $\mathrm{ind}_L(x)$ denotes the index of the element x in the list L. In the QRAM model, if L is classically stored and quantumly accessible then there exists an efficient quantum circuit that constructs the state $|\psi_L\rangle$.*

Theorem 1 (Grover's algorithm [Gro96]). *Consider a function $f : L \to \{0,1\}$. Given quantum access to the list L classically stored, Grover's algorithm returns $x \in L$ such that $f(x) = 1$ in time $\mathcal{O}(\sqrt{|L|})$.*

Theorem 2 (Amplitude amplification [Bra+02]). *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. Quantum amplitude amplification returns a solution with probability 1 in time $\mathcal{O}(1/\sqrt{p})$.*

We point out [NC00] to the reader for a complete introduction to quantum information theory.

2.2 Code-based cryptography

Generic codes.

Definition 2 **($[n,k]_q$-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 $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$ of C verifies $C = \{\boldsymbol{xG} \mid \boldsymbol{x} \in \mathbb{F}_q^k\}$ and a parity check matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ of C verifies $C = \{\boldsymbol{y} \in \mathbb{F}_q^n \mid \boldsymbol{yH}^\top = \boldsymbol{0}\}$. For any $\boldsymbol{y} \in \mathbb{F}_q^n$, the vector $\boldsymbol{yH}^\top$ is called the syndrome of $\boldsymbol{y}$ (relatively to $\boldsymbol{H}$). The dual code of C is $C^\perp = \{\boldsymbol{xH} \mid \boldsymbol{x} \in \mathbb{F}_q^{n-k}\}$.*

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

The problem $\mathsf{DP}_{\mathbf{H},\mathbf{s},w}$ is hard on average for $\mathbf{H}$ uniformly distributed in $\mathbb{F}_q^{(n-k)\times n}$ and $\mathbf{s} = \boldsymbol{e}\mathbf{H}^\top$ with $\boldsymbol{e}$ uniform vector in $\mathbb{F}_q^n$ of weight $|\boldsymbol{e}| = w$. The best known algorithms 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 $\mathsf{DP}_{\mathbf{H},\mathbf{s}=\mathbf{0},w}$. This problem is as hard as $\mathsf{DP}_{\mathbf{H},\mathbf{s},w}$ with an arbitrary $\mathbf{s}$.

Proposition 1. *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 at number $\binom{n}{w}\frac{2^w}{3^{n-k}}$.*

Proof. There are $\binom{n}{w} 2^w$ words of length n and weight w in $\mathbb{F}_3$. For some $\boldsymbol{e} \in \mathbb{F}_3^n$ and $\mathbf{H} \in \mathbb{F}_3^{(n-k)\times n}$, the vector $\boldsymbol{e}\mathbf{H}^\top$ has 3^{n-k} possible values, so the probability that for a given $\boldsymbol{e}$ it gives the correct one is $\frac{1}{3^{n-k}}$. □

Remark. There does not necessarily exist a solution to generic instances of the DP problem. But in the Wave settings, there is always at least one solution on average, and even there are exponentially many ones.

Wave.

Definition 3 **(Generalized ternary $(U, U+V)$-code).** *We consider integers n, k, k_U, k_V with n even such that $n > k > 0$, $k = k_U + k_V$, $0 < k_U < n/2$ and $0 < k_V < n/2$. For i from 0 to $n/2$, let $\boldsymbol{a} = (a_i)_i$, $\boldsymbol{b} = (b_i)_i$, $\boldsymbol{c} = (c_i)_i$ and $\boldsymbol{d} = (d_i)_i$ denote vectors in $\mathbb{F}_3^{n/2}$ such that $\forall i \in [0, n/2]$, $a_ic_i \neq 0$ and $a_id_i - b_ic_i \neq 0$.*

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

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

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

Definition 4 (Type-U and type-V codewords). *We consider a generalized ternary $(U, U+V)$-code C and retake the above notations. Given a chosen target weight t, we call a type-U codeword in C a word $\boldsymbol{u} \in U$ of form $\boldsymbol{u} = (\boldsymbol{a}\star \boldsymbol{u} \| \boldsymbol{c} \star \boldsymbol{u})$ and of weight $|\boldsymbol{u}| = t/2$. And a type-V codeword in C is a word $\boldsymbol{v} \in V$ of form $\boldsymbol{v} = (\boldsymbol{b} \star \boldsymbol{v} \| \boldsymbol{d} \star \boldsymbol{v})$ and of weight $|\boldsymbol{v}| = t/2$.*

The Wave signature scheme [DST19] uses a permuted generalized ternary $(U, U+V)$-code of length n and dimension k admitting a parity check matrix $\mathbf{H} \in \mathbb{F}_3^{(n-k)\times n}$ that constitutes the public key. Are also fixed a weight w, and dimensions k_U for code U and k_V for V. The signature of a message m by 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 a $\boldsymbol{e}$ to sign their message m.

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 od C to be on average at number $\binom{n/2}{t/2}\frac{2^{t/2}}{3^{n/2-k_U}}$, and the number of type-V codewords of C to be on average at number $\binom{n/2}{t/2}\frac{2^{t/2}}{3^{n/2-k_V}}$.*

Key attacks. From the proposition just above, for some values of weight t the number of type-U codewords is higher than those expected for a random code, given by 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 attacks* on Wave. Notice that one can run this attack directly on the $(U, U+V)$-code but also on its dual code. We draw the attention of the reader to the work of [Sen23] for further details on type-U and type-V words.

Problem 2 (The Distinguishing Wave Keys Problem DWK_{n,k_U,k_V}). Given $\mathbf{H} \in \mathbb{F}_3^{(n-(k_U+k_V))\times n}$, decide whether $\mathbf{H}$ 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 $(\boldsymbol{e}, \mathbf{s}) \in \mathbb{F}_3 \times \mathbb{F}_3^{n-k}$ such that $|e| = w$ and $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$. This consists in 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 $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ and a target weight w, find $\mathbf{s} \in S$ and $\boldsymbol{e} \in \mathbb{F}_q^n$ such that $|\boldsymbol{e}| = w$ and $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$.

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 Schroeppel and Shamir [SS81] extended this idea by using Wagner's approach [Wag02]. 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; Bec+12], but this refinement only provides a very slight gain in the Wave setting as shown in [Bri+20]. With nearest-neighbour techniques [MO15; BM17; BM18; Car+22] 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; SS81].

To solve the problem $\mathsf{DP}_{\mathbf{H},\mathbf{s},w}$, the idea behind ISD is to rewrite $\mathbf{H}$ into a systematic form and then to solve an easier instance $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$, where $\mathbf{H}'' \in \mathbb{F}_q^{(k+\ell)\times \ell}$ and $\mathbf{s}'' \in \mathbb{F}_q^{\ell}$ for parameters ℓ the length of the $\mathbf{s}''$ 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.

In order to analyse 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 parameters 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 according 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*

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

[1] A recent paper [Car+22] 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.

Classical ISD algorithm.

Theorem 3. *We are given a classical algorithm that finds $NbSolFound$ solutions to $\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}$ in time $T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}$, among the $NbSol(\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p})$ total solutions to $\mathsf{DP}_{k+\ell,\ell,p}$. PrGoodCut is defined as in Proposition 1, and $NbSol(\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w})$ denotes the number of solutions to the $\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}$ problem. Then the classical Information Set Decoding framework (Algorithm 1) solves $\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}$ in time*

$$T_{DP} = \max\left\{T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}, \frac{T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}}{NbSol(\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}) \cdot PrGoodCut \cdot \frac{NbSolFound}{NbSol(\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p})}}\right\}.$$

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

```
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: $\boldsymbol{e} \in \mathbb{F}_3^n$ such that $\boldsymbol{e}\mathbf{H}_0^\top = \mathbf{s}$ and $|\boldsymbol{e}| = w$.
1: Pick a random permutation $\pi$ 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 $\boldsymbol{e}_0 = \pi^{-1}(\boldsymbol{e})$. It verifies $\boldsymbol{e}_0\mathbf{H}_0^\top = \mathbf{s}$ and $|\boldsymbol{e}_0| = w$.
```

Proof. We use the same notations as above.

Steps 1-2. Applying a random permutation and a partial Gaussian elimination on $\mathbf{H}$ can be done in a 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 *NbSolFound* 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 *NbSolFound* solutions that need to be checked for weight. The check time be dominated in the sum by the time of Step 3.

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 $\frac{NbSolFound}{NbSol(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p})}$ of the solutions to the $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$ subproblem. As there are $NbSol(\mathsf{DP}_{\mathbf{H},\mathbf{s},w})$ such solutions $\boldsymbol{e}$, the probability that one iteration returns a solution is

$$PrFindSol = \min\left\{1, NbSol(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) \cdot PrGoodCut \cdot \frac{NbSolFound}{NbSol(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p})}\right\}. \tag{2}$$

To get a solution with probability $1-o(1)$, one has to repeat steps (1-4) a number $1/PrFindSol$ of iterations. Once we have found a solution $\boldsymbol{e}$ such that $\boldsymbol{e}\mathbf{H}^{\top} = \mathbf{s}$, then $\pi^{-1}(\boldsymbol{e})\mathbf{H}_0^{\top}$, and this achieves the algorithm.

Putting everything together gives the result. □

Quantum ISD algorithm.

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 a 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 Definition 1).

Theorem 4. *We are given an algorithm that constructs a quantum superposition of NbSolFound solutions to $\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}$ in time $T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}$, among the $NbSol(\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p})$ total solutions to this subproblem. PrGoodCut is defined in Proposition 1, and $NbSol(\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w})$ denotes the number of solutions to $\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}$. Then in the quantum mode, Algorithm 2 solves $\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}$ in time*

$$T_{DP} = \max\left\{T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}, \frac{T_{\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p}}}{\sqrt{NbSol(\mathsf{DP}_{\boldsymbol{H},\boldsymbol{s},w}) \cdot PrGoodCut \cdot \frac{NbSolFound}{NbSol(\mathsf{DP}_{\boldsymbol{H}'',\boldsymbol{s}'',p})}}}\right\}.$$

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 $\pi$ 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 in quantum superposition a list $L$
   of $\boldsymbol{e}'' \in \mathbb{F}_3^{k+\ell}$ such that $|\boldsymbol{e}''| = p$ and $\boldsymbol{e}''\mathbf{H}''^\top = \mathbf{s}''$, i.e. construct the quantum state
   $$\frac{1}{\sqrt{|L|}} \sum_{(\boldsymbol{e}'',\boldsymbol{y})\in L} \left|\mathrm{ind}_L(\boldsymbol{e}'')\right\rangle \left|\boldsymbol{e}''\right\rangle,$$
   where $\mathrm{ind}_L(\boldsymbol{e}'')$ is the index of the tuple $(\boldsymbol{e}'', \boldsymbol{y} = \boldsymbol{e}''\mathbf{H}''^\top)$ in list $L$.
4: Test step. From a vector $\boldsymbol{e}''$ we can compute the complete candidate solution
   $\boldsymbol{e} \in \mathbb{F}_3^n$ such that $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$. We transform $|\psi_L\rangle$ into the state
   $$\frac{1}{\sqrt{|L|}} \sum_{\substack{(\boldsymbol{e}'',\boldsymbol{y})\in L \\ \boldsymbol{e}=(\mathbf{s}'-\boldsymbol{e}''\mathbf{H}'^\top\|\boldsymbol{e}'')}} \left|\mathrm{ind}_L(\boldsymbol{e}'')\right\rangle |\boldsymbol{e}\rangle.$$
   The vectors $\boldsymbol{e}$ in the second register then satisfy $\boldsymbol{e}\mathbf{H}^\top = \mathbf{s}$. Apply Grover to only
   keep the $\boldsymbol{e}$ in the superposition which are of weight $w$.
5: Apply a Amplitude Amplification to find a good permutation $\pi$ in Step 1 with
   high probability.
6: Measure $\boldsymbol{e}$. Return $\boldsymbol{e}_0 = \pi^{-1}(\boldsymbol{e})$. It satisfies $\boldsymbol{e}_0\mathbf{H}_0^\top = \mathbf{s}$ and $|\boldsymbol{e}_0| = w$
```

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

Step 3. This step takes time $T_{\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}}$ that depends on the choice of the subroutine. It constructs a quantum superposition $|\psi_L\rangle$ over $|L| = NbSolFound$ solutions to the $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$ subproblem, where

$$|\psi_L\rangle = \frac{1}{\sqrt{|L|}} \sum_{(\boldsymbol{e}'',\boldsymbol{y})\in L} \left|\mathrm{ind}_L(\boldsymbol{e}'')\right\rangle \left|\boldsymbol{e}''\right\rangle.$$

Actually, all the $\boldsymbol{y}$ in tuples in L in this state are equal to $\boldsymbol{y} = \mathbf{s}''$ as it is an output condition of the subroutine. So we can discard this last register that can now be considered classical.

Step 4. After discarding the classical register $|\mathbf{s}''\rangle$, we add a zero quantum register to $|\psi_L\rangle$ to get the state

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

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 $|0\rangle|\boldsymbol{e}''\rangle \mapsto |\mathbf{s}' - \mathbf{H}'\boldsymbol{e}''\rangle|\boldsymbol{e}''\rangle$ on its two last registers to get the state

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

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 on this state Grover's search [Gro96] with a check function that given $\boldsymbol{e}$ returns 1 if $|\boldsymbol{e}|=w$, and 0 otherwise. It transforms the state into the following quantum superposition over solutions to $\mathsf{DP}_{\mathbf{H},\mathbf{s},w}$

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

where Z is the number of such solutions $\boldsymbol{e}$. This requires at most $\sqrt{|L|}=\sqrt{NbSolFound}$ Grover iterations. Therefore the time of this step will be dominated by the time of Step 3.

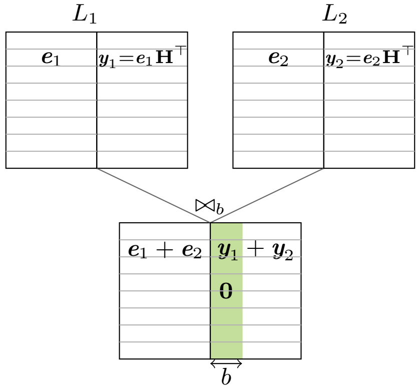
Step 5. Suppose there is a precise solution $\boldsymbol{e}$ that we want to find where $\boldsymbol{e}=(\boldsymbol{e}'\|\boldsymbol{e}'')$ with $\boldsymbol{e}'\in\mathbb{F}_3^{k+\ell}$ and $\boldsymbol{e}''\in\mathbb{F}_3^{n-k-\ell}$. Lemma 1 gives the probability that $\boldsymbol{e}$ is "well cut", *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 $\frac{NbSolFound}{NbSol(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p})}$ of the solutions to the $\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}$ subproblem. One iteration of Steps 1-4 returns a solution with probability given by Equation 2. To get a solution with a probability close to 1, we apply an amplitude amplification on this process, which takes $1/\sqrt{PrFindSol}$ 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 $\boldsymbol{e}_0\mathbf{H}_0=\mathbf{s}$. 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 Thorem 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 ISD algorithms will make great use of list merging. Merging two lists L_1 and L_2 on the b first coordinates means constructing the following list.

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



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

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

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 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 $\log |L_1|$ 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 $(|L_1| + |L_2|) \cdot \log |L_1|$. Hence the following lemma.

Lemma 2. *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 the state $|\psi_{L_2}\rangle$ of the uniform quantum superposition on the list L_2,

$$|\psi_{L_2}\rangle = \frac{1}{\sqrt{|L_2|}} \sum_{(e_2,\boldsymbol{y}_2)\in L_2} |\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\rangle\, |\boldsymbol{e}_2\rangle\, |\boldsymbol{y}_2\rangle\,. \tag{5}$$

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:

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

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 $|L_1|\le|L_2|$, then there will be on average at most one such tuple in L_1. 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:

$$\frac{1}{\sqrt{|L_2|}} \sum_{(e_2,\boldsymbol{y}_2)\in L_2} |\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\rangle\, |match_{L_1}(\boldsymbol{e}_2,\boldsymbol{y}_2)\rangle\, |\boldsymbol{e}_2\rangle\, |\boldsymbol{y}_2\rangle\,.$$

While the classical merging ran a `for` loop on L_2 to check, in the quantum model we replace it with Grover's search [Gro96]. We define the Grover check function as follows: Given $(\boldsymbol{e}_2,\boldsymbol{y}_2)$, it returns 1 if $match_{L_1}(\boldsymbol{e}_2,\boldsymbol{y}_2)\neq\perp$, and 0 else. Applying Grover on the previous state requires at most $\sqrt{|L_2|}$ iterations, and constructs the following state, where the non-$\perp$ elements are removed from the superposition.

$$\frac{1}{\sqrt{|L_1 \bowtie_b L_2|}} \sum_{\substack{(e_2,\boldsymbol{y}_2)\in L_2 \\ match_{L_1}(e_2,\boldsymbol{y}_2)=(e_1,\boldsymbol{y}_1)\neq\perp}} |\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\rangle\, |\boldsymbol{e}_1\rangle\, |\boldsymbol{y}_1\rangle\, |\boldsymbol{e}_2\rangle\, |\boldsymbol{y}_2\rangle\,.$$

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

$$\frac{1}{\sqrt{|L_1 \bowtie_b L_2|}} \sum_{\substack{(e_2,\boldsymbol{y}_2)\in L_2 \\ match_{L_1}(e_2,\boldsymbol{y}_2)=(e_1,\boldsymbol{y}_1)\neq\perp}} |\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\rangle\, |\boldsymbol{e}_1+\boldsymbol{e}_2\rangle\, |\boldsymbol{y}_1+\boldsymbol{y}_2\rangle\, |\boldsymbol{e}_2,\boldsymbol{y}_2\rangle\,,$$

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

[2] When we will apply quantum merging further in this work, we will manipulate random lists L_1, L_2 such that $|L_1|^2 = |L_2|$, 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.

state then can be rewritten

$$|\psi_{L_1 \bowtie_b L_2}\rangle\,|\mathrm{Aux}\rangle := \frac{1}{\sqrt{|L_1 \bowtie_b L_2|}} \sum_{\substack{(\boldsymbol{e},\boldsymbol{y})\in L_1\bowtie_b L_2\\ e=e_1+e_2,e_1\in L_1,e_2\in L_2}} |\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\rangle\,|\boldsymbol{e}\rangle\,|\boldsymbol{y}\rangle\,|\mathrm{Aux}\rangle\,. \tag{6}$$

This whole process takes time $(|L_1| + \sqrt{|L_2|})\cdot\log|L_1|$.

Lemma 3. *Given a list L_1 classically stored and quantumly accessible, and the quantum state $|\psi_{L_2}\rangle$ (Eq. 5) such that $|L_1| \leq |L_2|$, one can construct the quantum state $|\psi_{L_1\bowtie_b L_2}\rangle$ (Eq. 6) for an arbitrary b in time $\widetilde{\mathcal{O}}(|L_1|, \sqrt{|L_2|})$.*

3 Key attacks on DWK

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 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. [Sen23] pointed out 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. The parameter 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

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

So the time complexity of this key 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, with t and t' are freely chosen such that they respect Equation 7. In this section, we present how to solve $\mathsf{DP}_{\mathbf{H},\mathbf{0},t}$ and these algorithms can directly be adapted to the dual version.

3.1 Classical key 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 start by constructing the following lists.

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

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 $(\boldsymbol{e}'', \boldsymbol{e}''\mathbf{H}''^\top) = (\boldsymbol{e}'', \mathbf{0})$. We can see in Equation 8 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, which ensures the correctness of the algorithm. Using this process as a subroutine within the ISD framework leads to the following theorem.

Theorem 5. *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*

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

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: $NbSol(\mathsf{DP}_{\mathbf{H}'',\mathbf{0},p}) = \binom{k+\ell}{p} 2^p$. And Proposition 2 gives the number of solutions to the DP problem in the $(U, U+V)$-code: $NbSol(\mathsf{DP}_{\mathbf{H},\mathbf{0},t}) = \binom{n/2}{t/2} \frac{2^{t/2}}{3^{n/2-k_V}}$.

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 since 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 see 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 optimisation to minimize the time complexity of the attack. As a result, we obtain as optimal parameters $l \approx 0.01$, $t \approx 0.21$ and $p \approx 0.003$. 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) can be found in the following table.

Table 2. Sets of Wave parameters as selected in [Ban+23] (NIST submission, round 1) and the corresponding required security levels in the number of bits.

	Classic	Quantum	n	k	w	k_U
(I)	128	64	8576	4288	7668	2966
(III)	192	96	12544	6272	11226	4335
(V)	256	128	16512	8256	14784	5704

With this choice of parameters, the algorithm finds $|L|$ solutions in time $|L|$, so in amortized time $O(1)$ 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.

3.2 Quantum key attack

The quantum key attack algorithm has a very similar structure to the classical one. We use the quantum version of the ISD framework (Algorithm 2), which performs a quantum Amplitude Amplification (Theorem 2 [Bra+02]) instead of a classical while 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.

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: $|\psi_{L_2}\rangle = \frac{1}{\sqrt{|L_2|}} \sum_{(\boldsymbol{e}_2,\boldsymbol{y}_2)\in L_2} |\mathrm{ind}_{L_2}(\boldsymbol{e}_2)\rangle |\boldsymbol{e}_2\rangle |\boldsymbol{y}_2\rangle$, 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 $|\psi_{L_1 \bowtie_\ell L_2}\rangle$, 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 look at 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 $|\boldsymbol{e}''| = |\boldsymbol{e}''_1| + |\boldsymbol{e}''_2| = p/3 + 2/3 = p$. 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. *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 11 just above. 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 $|\boldsymbol{e}''_2\rangle |0\rangle \mapsto |\boldsymbol{e}''_2\rangle |\boldsymbol{e}''_2\mathbf{H}''^{\top}\rangle$.

The quantum merging takes time $|L_1| + \sqrt{|L_2|}$ by Lemma 3. On average we can expect $|L_1 \bowtie_\ell L_2| = \binom{k+\ell}{p} 2^p \cdot 3^{-\ell} := \mathit{NbSolFound}$ by Equation 4. This is also equal, up to a polynomial factor, to $\mathit{NbSol}(\mathsf{DP}_{\mathbf{H}'',\mathbf{0},p})$ 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 $\mathit{NbSol}(\mathsf{DP}_{\mathbf{H},\mathbf{0},t}) = \binom{n/2}{t/2} \frac{2^{t/2}}{3^{n/2-k_U}}$. Plugging these values into the Theorem 4 with the same notations provides the result. □

3.3 Numerical results

The time complexity is optimal when the list L_2 is of maximal size, so when p is fixed such that $\binom{k+\ell}{p}^{1/3} 2^{p/3} = 3^\ell$. Parameters ℓ and t are obtained by numerical optimisation 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 2.

The time complexity is optimal for $t \approx 0.21$, $l \approx 0.0052$ and $p \approx 0.0024$. 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}. The slight differences in the time exponents come from the fact that the dimensions k_U, k_V are not exactly linear in n. These results are summarized in the second column of Table 1.

4 Message attacks

This attack consists 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 we take Wagner's algorithm [SS81] as a subroutine instead of just Dumer's [Dum91].

4.1 Classical message attack

The best known classical message attack algorithm is the smoothed Wagner's algorithm from [Bri+20] 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 [SS81] can be seen as a generalisation 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

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

$$L_i^{(0)} := \left\{ ((\boldsymbol{e}'', \mathbf{0}), \boldsymbol{e}''\mathbf{H}''^\top) : \boldsymbol{e}'' \in E_i^{(0)} \right\} \tag{12}$$

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

$$L_{2^a}^{(0)} \subseteq \left\{ ((\mathbf{0}, \mathbf{s}''), -\mathbf{s}'') : \mathbf{s} = (\mathbf{s}' \| \mathbf{s}'') \in S, \mathbf{s}' \in \mathbb{F}_3^{n-k-\ell}, \mathbf{s}'' \in \mathbb{F}_3^\ell \right\}. \tag{13}$$

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 [Bri+20] has already shown and that we recover in our numerical optimisations, 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:

$$3^{\ell/a} \leq 2^{\frac{k+\ell}{2^a-1}}. \tag{14}$$

Merging tree. For Wagner's algorithm, we consider a binary merging tree with at the first level the lists $L_1^{(0)}, \dots, L_{2^a-1}^{(0)}$ and $L_{2^a}^{(0)}$ defined in Equations 12 and 13. To pass from the j-th level to the $(j+1)$-th we merge pairwise lists using Lemma 2, for odd i:

$$L_{(i+1)/2}^{(j+1)} := L_i^{(j)} \bowtie_{(j+1)\ell/a} L_{i+1}^{(j)} \tag{15}$$

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 much solutions $(\boldsymbol{e}'', \mathbf{s}'')$ to the $\mathsf{DP}_{\mathbf{H}'', \mathbf{s}'', p}$ subproblem.

Algorithm 3 Classical Wagner's algorithm for DOOM [Bri+20]

```
Input: $\mathbf{H}'' \in \mathbb{F}_3^{(k+\ell)\times\ell}$, a list $S$ of target syndromes $\mathbf{s}''_1, \dots, \mathbf{s}_{3^{\ell/a}} \in \mathbb{F}_3^{\ell}$
length $\ell$, 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 12.
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 $L_1^{(a)}$
```

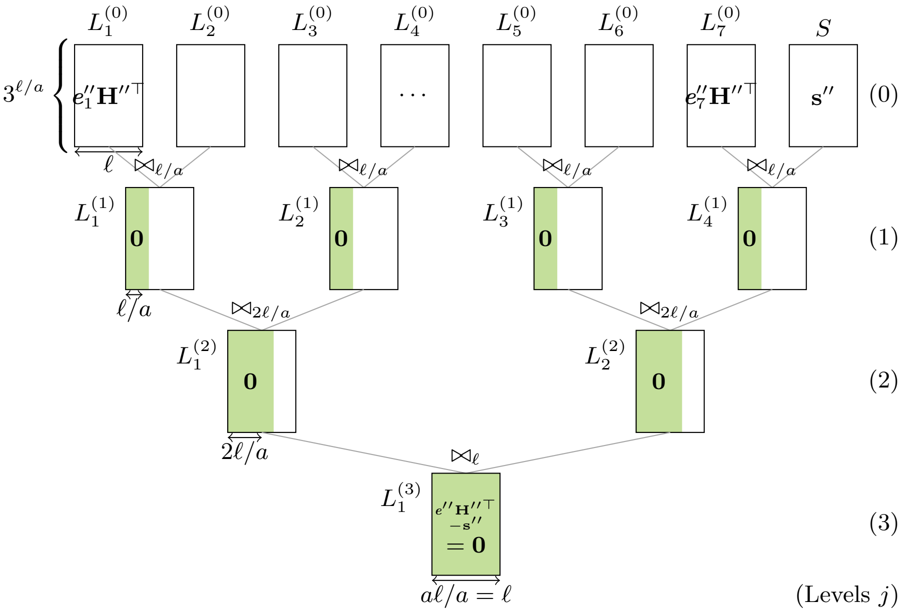


Fig. 2. Wagner subroutine for $a = 3$. There are $2^a - 1 = 7$ initial lists of $\boldsymbol{e}''\mathbf{H}''^\top$, plus the syndromes list S. At each level, the lists are merged on ℓ/a more coordinates, until the final list $L_1^{(a)}$ filled with elements in the form $((\boldsymbol{e}'', \mathbf{s}''),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

Proposition 3. *Let us fix parameters ℓ, p, a such that $3^{\ell/a} \leq 2^{\frac{k+\ell}{2^a-1}}$ (See Equation 14). 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 $3^{\ell/a}$, so Wagner's subroutine 3 takes time $T_{\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}} = 3^{\ell/a}$ to find $NbSolFound = 3^{\ell/a}$ solutions. By Proposition 1, the solutions to the DP problem are at number $NbSol(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) = \binom{n}{w}\frac{2^w}{3^{n-k}}$, and the solutions to the subproblem are at number $NbSol(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}) = \binom{k+\ell}{p}2^p$. 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. [Bri+20] introduced a smoothed Wagner algorithm, whose idea is to start with longer lists and a stricter first merging. 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 [Bri+20]

```
Input: $\mathbf{H}'' \in \mathbb{F}_3^{(k+\ell)\times\ell}$, target syndromes $\mathbf{s}''_1, \dots, \mathbf{s}_{3^{\ell/a}} \in \mathbb{F}_3^{\ell}$
length $\ell$, target weight $p$, tree depth $a$, smoothing parameters $m$, $\lambda$.
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 $2^a - 1$, and $L_{2^a}^{(0)}$ using Equation 12.
Compute $\lambda$ and $m$ using Equations 16 and 18.
for $i = 1$ to $2^a$ do
    Merge $L^{(1)}_{(i+1)/2} = L_i^{(0)} \bowtie_m L_{i+1}^{(0)}$
for $j = 1$ to $a - 1$ do
    for $i = 1$ to $2^{(a-j)}$ do
        Merge $L^{(j+1)}_{(i+1)/2} = L_i^{(j)} \bowtie_{m+j\frac{\lambda}{\log_2(3)}} L_{i+1}^{(j)}$
return $L_1^{(a)}$
```

Proposition 4. *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 $DP_{\boldsymbol{H}'',\boldsymbol{s}'',p}$in time $\mathcal{O}(2^\lambda)$ with*

$$\lambda = \frac{1}{a-2}\left(\ell\log(3) - 2\cdot\frac{k+\ell}{2^a-1}\right). \tag{16}$$

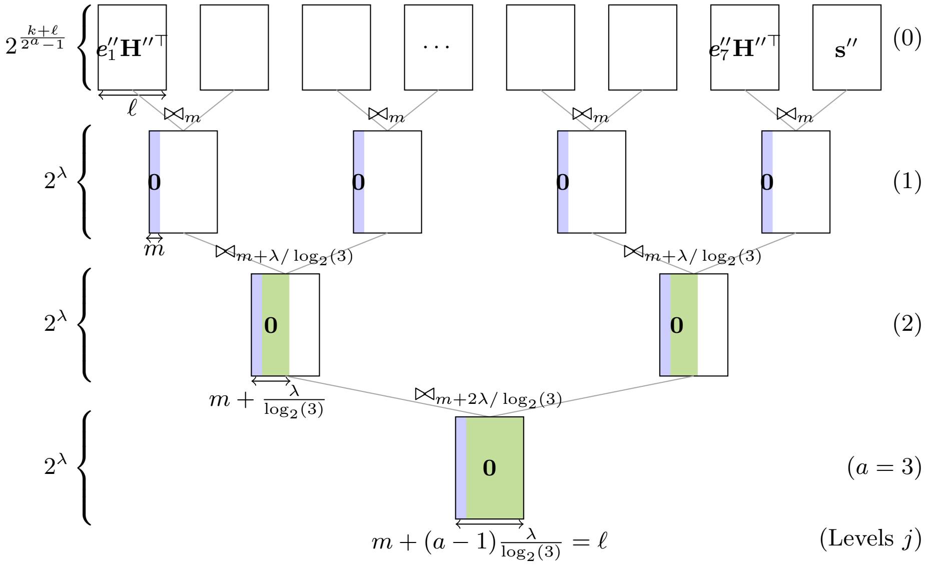


Fig. 3. Smoothed Wagner's algorithm for $a = 3$. The first merging is operated on a small number of coordinates m, and then we merge on $\lambda/\log_2 3$ more coordinates at each level.

Proof. We restate the proof from [Bri+20] 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 14, 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 firstly 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

$$\frac{\left(2^{\frac{k+\ell}{2^a-1}}\right)^2}{3^m} = 2^\lambda \quad i.e. \quad \lambda = \frac{2(k+\ell)}{2^a-1} - m\log_2 3 \tag{17}$$

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:

$$m + (a-1)\frac{\lambda}{\log_2 3} = \ell. \tag{18}$$

By combining Equations 17 and 18, 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 7. *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 $NbSol(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) = \binom{n}{w} \frac{2^{w}}{3^{n-k}}$ 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 optimisation. The optimal a in this setting is here $a = 5$ and $\ell \approx 0.05$. The optimal values of ℓ may slightly vary in function of Wave parameters due to the fact that 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. If 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 bits of security less in comparison with the analysis of the smoothed Wagner algorithm from [Bri+20]. 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

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 a 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 Definition 1).

For the quantum message attack, we combine DOOM approach from [Sen11], quantum Wagner's algorithm of [CDE21] and smoothing from [Bri+20]. 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.

Algorithm 5 Quantum 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 $\ell$, 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 $\boldsymbol{e}''\mathbf{H}''^{\top} = \mathbf{s}''$
Sample lists $L_i^{(0)}$ for $i = 1$ to $2^a - 1$ using Equations 12 and 13.
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 $j = 0$ to $a - 1$ do
    for $i = 1$ to $2^{(a-j)} - 1$ do
        Classically merge $L_{(i+1)/2}^{(j+1)} = L_i^{(j)} \bowtie_{(j+1)\ell/a} L_{i+1}^{(j)}$
    Quantumly merge $L_{(2^{a-j}+1)/2}^{(j+1)} = L_{2^{a-j}-1}^{(j)} \bowtie_{(j+1)\ell/a} L_{2^{a-j}}^{(j)}$
return $\left|\psi_{L_1^{(a)}}\right\rangle$
```

Proposition 5. *We are given n, k, w. Let 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*

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

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 $NbSol(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) = \binom{n}{w}\frac{2^w}{3^{n-k}}$, and to the DP subproblem $NbSol(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}) = \binom{k+\ell}{p} 2^p \cdot 3^{-\ell}$. Applying Theorem 3 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. □

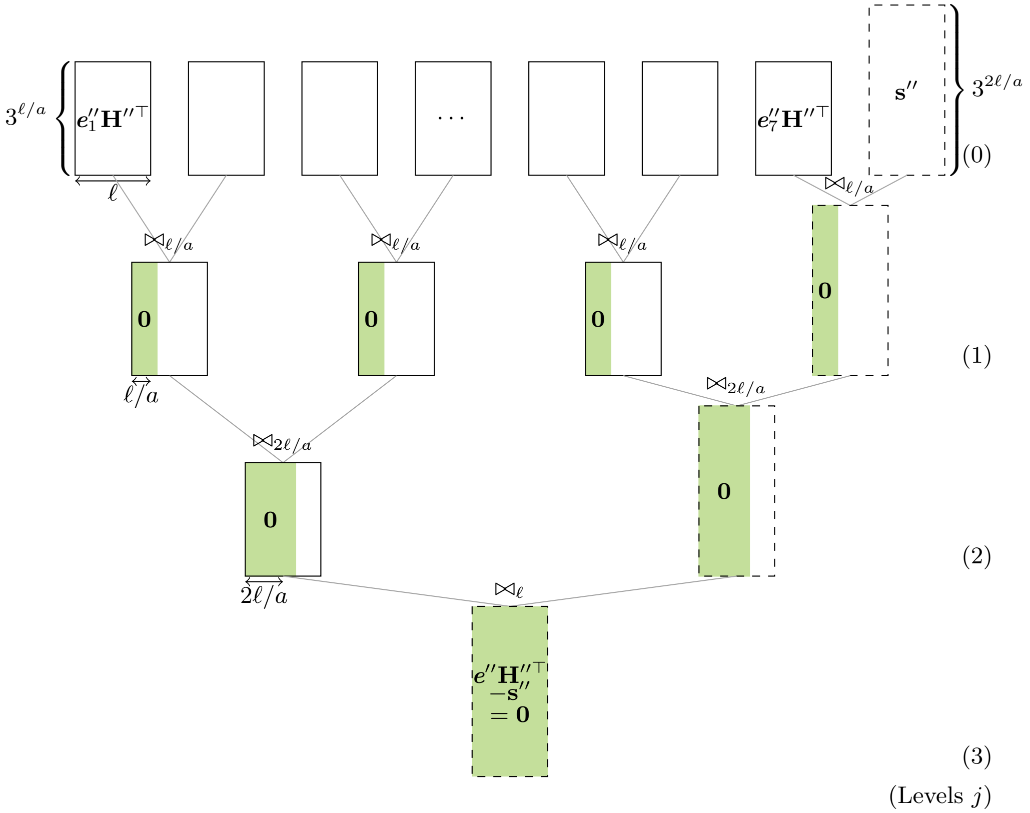


Fig. 4. Quantum Wagner's algorithm from Algorithm 5. Dashed-line boxes represent lists that are not classically constructed in hard but whose we have a quantum superposition of its elements.

Algorithm 6 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 $\ell$, 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
$\boldsymbol{e}''\mathbf{H}''^{\top} = \mathbf{s}''$
Sample lists $L_i^{(0)}$ for $i = 1$ to $2^a - 1$ using Equations 12 and 13.
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 8. *We are given n, k, w. Let 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 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 14. 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 16 and 18, where $\lambda = \frac{1}{a-2}\left(\ell\log(3) - \frac{2(k+\ell)}{2^a-1}\right)$.

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 $T_{\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}} = 2^{\lambda}$

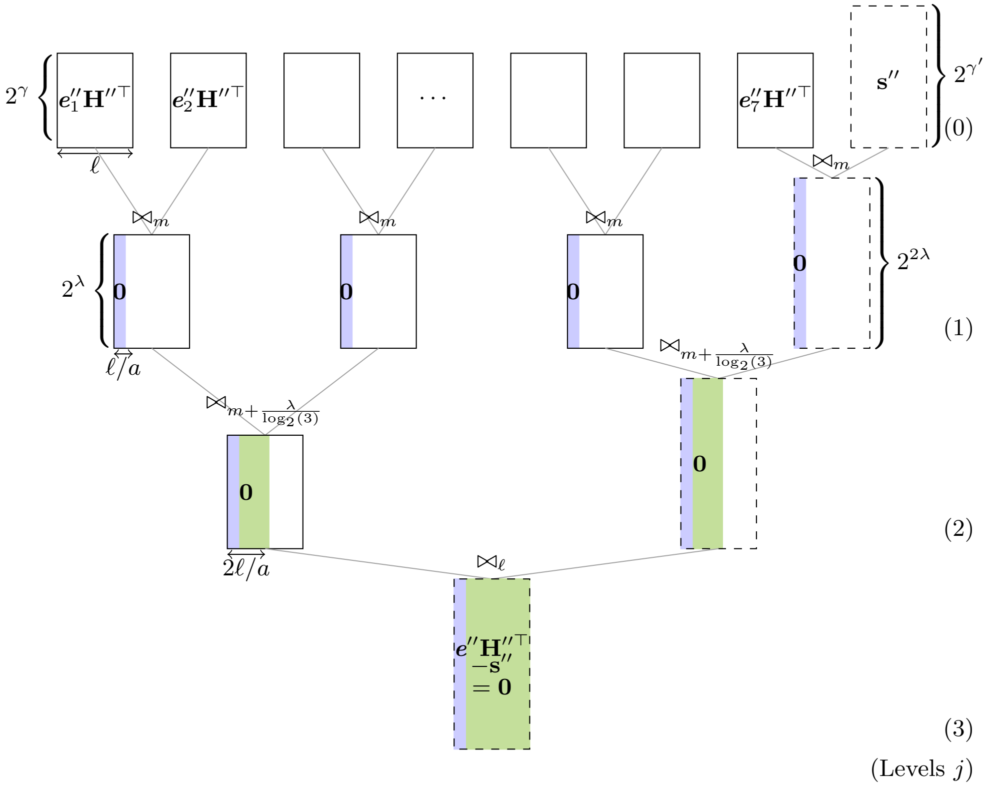


Fig. 5. Quantum smoothed Wagner's algorithm (Proposition 8 and Algorithm 6). Dashed-line boxes represent lists that are not classically constructed in hard but whose we have a quantum superposition of its elements.

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 $NbSol(\mathsf{DP}_{\mathbf{H},\mathbf{s},w}) = \binom{n}{w}\frac{2^w}{3^{n-k}}$, and to the DP subproblem $NbSol(\mathsf{DP}_{\mathbf{H}'',\mathbf{s}'',p}) = \binom{k+\ell}{p} 2^p \cdot 3^{-\ell}$. 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 optimisation.

Without smoothing. Taking $l \approx 0.032$ 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 $l \approx 0.034$ 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 3. Number of quantum security bits for message attacks.

Algorithm	(I)	(III)	(V)
ISD + Wagner [CDE21]	79	120	161
Estimation [Ban+23]	77	117	157
ISD + Smoothed Wagner (Thm. 8)	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 [Ban+23], whose estimation did not rely on analysing an explicitly described algorithm, which is now formalized by our theorem 8. 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 Conclusion

The process to deduce the optimal Wave parameters [Sen23] takes the trade-off between opposite constraints induced by each attack, key and message. Doing so, Wave parameter selection from [Ban+23] (recalled in Table 2) is chosen such that the time of both best known attacks is superior and the closest to the required security bits number. Table 1 reveals a visible gap between the minimum security threshold and the time of key attacks. This is a direct consequence of the discreteness of the Wave parameters, which prevents exactly reaching the minimum number of security bits for the two attacks at once.

For message attacks presented in Section 4, our analysis of the classical algorithm shows that it gets close time complexity to the lower bound from [FS09] which applies to the ISD class of algorithms. We correct the claimed security level given in [Ban+23] for quantum message attacks, finding a slight difference of one security bit with their estimation. We observe that key attacks benefit more from the quantum setting than message attacks. The reason is that Grover's algorithm has a stronger impact when there is a search on a large range, as in Dumer's algorithm, instead of on several fragmented small ones as what occurs in Wagner's algorithm.

Quantum security seems to be far from being a limiting factor for Wave. A quadratic gain is not even reached and there is a large margin of safety from the minimum threshold. Moreover, the time analysis of the quantum attacks was done without considering the extra time of QRAM operations. Future practical implementations of these attacks then can be way more demanding in running time. Therefore the classical attacks remain the ones to consider in priority for selecting Wave optimal parameters. It is now 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.

Code used for the numerical results. All our numerical optimisations and results have been obtained using Sage-Math. Please find the code file here: `https://github.com/johanna-loyer/WaveISDcryptanalysis.git`.

Acknowledgements. The author would like to thank André Chailloux and Nicolas Sendrier for helpful discussions.

References

[Ara+21] N Aragon, P. L. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Ghosh, S. Gueron, T. Güneysu, C. Aguilar Melchor, R. Misoczki, E. Persichetti, J. Richter-Brockmann, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. Zémor. "BIKE - Bit Flipping Key Encapsulation". In: NIST PQC (2021). URL: `https://bikesuite.org/files/v5.0/BIKE_Spec.2022.10.10.1.pdf`.

[Ban+21] Gustavo Banegas, Thomas Debris-Alazard, Milena Nedeljkovi, and Benjamin Smith. "Wavelet: Code-based postquantum signatures with fast verification on microcontrollers". In: Cryptology ePrint Archive (2021). URL: `https://ia.cr/2021/1432`.

[Ban+23] Gustavo Banegas, Pierre Karpman, Kévin Carrier, Johanna Loyer, André Chailloux, Ruben Niederhagen, Alain Couvreur, Nicolas Sendrier, Thomas Debris-Alazard, Benjamin Smith, Philippe Gaborit, and Jean-Pierre Tillich. "Wave Support Documentation". In: NIST PQC (2023). URL: https://wave-sign.org/wave_documentation.pdf.

[Bec+12] 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.

[Ber10] Daniel J. Bernstein. "Grover vs. McEliece". In: PQCrypto (2010). URL: https://cr.yp.to/codes/grovercode-20100303.pdf.

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. "Smaller decoding exponents: ball-collision decoding". In: Crypto. Vol. 6841. Lecture Notes in Computer Science. 2011, pp. 743–760.

[BM17] Leif Both and Alexander May. "Optimizing BJMM with Nearest Neighbors: Full Decoding in $2^{2/21n}$ and McEliece Security". In: WCC. Sept. 2017. URL: http://wcc2017.suai.ru/Proceedings%7B%5C_%7DWCC2017.zip.

[BM18] Leif Both and Alexander May. "Decoding Linear Codes with High Error Rate and Its Impact for LPN Security". In: PQCrypto. Ed. by Tanja Lange and Rainer Steinwandt. Vol. 10786. Lecture Notes in Computer Science. Fort Lauderdale, FL, USA: Springer, Apr. 2018, pp. 25–46. URL: https://doi.org/10.1007/978-3-319-79063-3%7B%5C_%7D2.

[BMT78] E. R. Berlekamp, R. J. McEliece, and H. C. van Tilborg. "On the inherent intractability of certain coding problems". In: IEEE 24, no.3 (1978). URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1055873.

[Bon+20] Xavier Bonnetain, Rémi Bricout, André Schrottenloher, and Yixin Shen. "Improved Classical and Quantum Algorithms for Subset-Sum". In: ASIACRYPT. 2020. ISBN: 978-3-030-64833-6. URL: https://doi.org/10.1007/978-3-030-64834-3_22.

[Bra+02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. "Quantum amplitude amplification and estimation". In: QCQI (2002), 305:53–74. URL: https://arxiv.org/abs/quant-ph/0005055.

[Bri+20] Rémi Bricout, André Chailloux, Thomas Debris-Alazard, and Matthieu Lequesne. "Ternary Syndrome Decoding with Large Weight". In: SAC (2020). URL: https://arxiv.org/pdf/1903.07464.pdf.

[Car+22] 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.

[CDE21] André Chailloux, Thomas Debris-Alazard, and Simona Etinski. "Classical and Quantum algorithms for generic Syndrome Decoding problems and applications to the Lee metric". In: PQCrypto (2021). URL: https://arxiv.org/pdf/2104.12810.pdf.

[DST19] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. "Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes". In: ASIACRYPT (2019). URL: https://arxiv.org/pdf/1810.07554.pdf.

[Dum91] Ilya Dumer. "On minimum distance decoding of linear codes". In: Joint Soviet-Swedish Int. Workshop Inform. Theory (1991), pp. 50–52.

[Fou+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". In: 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: ASIACRYPT. Ed. by M. Matsui. Vol. 5912. Lecture Notes in Computer Science. Springer, 2009, pp. 88–105. 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. ACM. 2008, pp. 197–206. URL: https://dl.acm.org/doi/pdf/10.1145/1374376.1374407.

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

[Jab01] Abdulrahman Al Jabri. "A statistical decoding algorithm for general linear block codes". In: 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". In: 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: PQCrypto. Ed. by Tanja Lange and Rainer Steinwandt. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 507–527. URL: https://doi.org/10.1007/978-3-319-79063-3%5C_24.

[KL22] Pierre Karpman and Charlotte Lefevre. "Time-Memory Tradeoffs for Large-Weight Syndrome Decoding in Ternary Codes". In: PKC. Ed. by Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe. Vol. 13177. Lecture Notes in Computer Science. Springer, 2022, pp. 82–111. URL: https://doi.org/10.1007/978-3-030-97121-2%5C_4.

[KT17] Ghazal Kachigar and Jean-Pierre Tillich. "Quantum Information Set Decoding Algorithms". In: PQCrypto. Ed. by Tanja Lange and

Tsuyoshi Takagi. Vol. 10346. Lecture Notes in Computer Science. Springer, 2017, pp. 69–89. URL: https://arxiv.org/pdf/1703.00263.pdf.

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

[Mel+21] 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)”. In: NIST PQC (2021). URL: https://pqc-hqc.org/download.php?file=hqc-specification_2023-04-30.pdf.

[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: EUROCRYPT. Ed. by E. Oswald and M. Fischlin. Vol. 9056. Lecture Notes in Computer Science. Springer, 2015, pp. 203–228. URL: https://www.crypto.ruhr-uni-bochum.de/imperia/md/content/may/paper/codes.pdf.

[NC00] Michael A. Nielsen and Isaac L. Chuang. QCQI. New York, NY, USA: Cambridge University Press, 2000. ISBN: 0-521-63503-9.

[Pra62] Eugene Prange. “The use of information sets in decoding cyclic codes”. In: IRE Transactions on Information Theory 8.5 (1962), pp. 5–9. URL: http://dx.doi.org/10.1109/TIT.1962.1057777.

[Sen11] Nicolas Sendrier. “Decoding one out of many”. In: PQCrypto (2011). URL: https://eprint.iacr.org/2011/367.pdf.

[Sen23] Nicolas Sendrier. “Wave Parameter Selection”. In: PQCrypto (2023). URL: https://eprint.iacr.org/2023/588.pdf.

[SS81] Richard Schroeppel and Adi Shamir. “A $T = O(2^{n/2})$, $S = O(2^{n/4})$ Algorithm for Certain NP-Complete Problems”. In: SIAM (1981). URL: https://epubs.siam.org/doi/epdf/10.1137/0210033.

[Ste88] Jacques Stern. “A method for finding codewords of small weight”. In: Coding Theory and Applications. Ed. by G. D. Cohen and J. Wolfmann. Vol. 388. Lecture Notes in Computer Science. Springer, 1988, pp. 106–113.

[Wag02] David Wagner. “A generalized birthday problem”. In: Crypto (2002). URL: https://www.enseignement.polytechnique.fr/informatique/profs/Francois.Morain/Master1/Crypto/projects/Wagner02.pdf.