A tight security reduction in the quantum random oracle model for code-based signature schemes

André Chailloux[2] and Thomas Debris-Alazard[1,2]

[1] Sorbonne Universités, UPMC Univ Paris 06
[2] Inria, Paris
{andre.chailloux,thomas.debris}@inria.fr

Abstract. Quantum secure signature schemes have a lot of attention recently, in particular because of the NIST call to standardize quantum safe cryptography. However, only few signature schemes can have concrete quantum security because of technical difficulties associated with the Quantum Random Oracle Model (QROM). In this paper, we show that code-based signature schemes based on the full domain hash paradigm can behave very well in the QROM *i.e.* that we can have tight security reductions. We also study quantum algorithms related to the underlying code-based assumption. Finally, we apply our reduction to a concrete example: the SURF signature scheme. We provide parameters for 128 bits of quantum security in the QROM and show that the obtained parameters are competitive compared to other similar quantum secure signature schemes.

Keywords: Quantum Random Oracle, Quantum-Safe Cryptography, Code-Based Cryptography, Signature Scheme, Quantum Decoding Algorithm

1 Introduction

Quantum computers are a potential big threat for many public-key cryptosystems. Every cryptographic application based on the hardness of factoring or the discrete logarithm (in a finite field or on elliptic curves) can be broken using Shor's quantum algorithm [Sho94]. Even if the future existence of such quantum hardware is still in question, there is a rapidly increasing interest in building cryptosystems which are secure against quantum computers, field sometimes referred to as *post-quantum* cryptography, or *quantum-safe* cryptography.

There are at least a couple of reasons why we should care and develop this line of research. First, sensitive data is now stored online and we want to guarantee long-term security. Indeed, we do not want current medical, political or other sensitive data to be stored today and decrypted in let's say 20 to 30 years from now, so cryptographic applications should try to find a way to prevent this kind of attacks. Quantum-safe cryptography would prevent a potential quantum computer to break today's schemes. Moreover, creating alternatives to RSA-based schemes could also be useful if another (non quantum) attack is found on factoring or the discrete log. While this doesn't seem to be the most probable, there has been for example big improvements on the discrete logarithm problem [BGGM15] so such a scenario is not totally out of the question. Those concerns are strong and initiated a NIST call to standardize quantum-safe cryptography [NIS16], and also lead to industrial interest [Goo16].

In this paper, we study (classical) signature schemes secure against quantum adversaries. Digital signature schemes allow to authenticate messages and documents and are a crucial element of many cryptographic applications such as software certification. There are several proposals for quantum-safe signatures based mostly on the hardness of lattice problems, such as BLISS [DDLL13], GPV [EBB14] or TESLA [ABB+17]. There are also other quantum-safe assumptions that can be used such as the hardness of code-based problems, multivariate polynomial problems [DS05] or the quantum security of hash functions [BHH+15]. All of those quantum-secure signature schemes have different merits and imperfections. Some have good time and size parameters but use a very structured lattice-based assumption. Others have large key and/or signature sizes and can have large running times.

Code based cryptography is among the oldest proposals for modern cryptography but suffered historically from the difficulty to construct a good signature scheme. The underlying computational assumption was actually one of the first proposed computational assumption [Mc 78] and still resists to known classical and quantum attacks. Until recently, there were very few proposals that were able to perform a code-based digital signature, the most notable being the CFS signature scheme [CFS01]. However, a very recent proposal, the SURF signature scheme (see [DST17]), presents competitive parameters (comparable to TESLA) but the security was shown only against classical adversaries.

Actually, most of signature schemes listed above - even though they use a quantum-safe computational assumption - can only prove security against classical adversaries. In fact, as of today only SPHINCS and TESLA-2 have full security reductions which claim 128 bits of quantum security. This small amount of signature schemes comes from the difficulties to deal with the quantum random oracle model (QROM). In most of the security proofs used, we are in the random oracle model meaning that we use a hash function that behaves as a random function. A quantum attacker could still perform superposition attacks on this hash function and this creates many difficulties in the security reductions. There has already been a extensive amount of work to provide security reduction in the QROM[BDF$^+$11, Zha12]. However, most of them are not tight and there are significant losses in the parameters that can be used. TESLA recently managed to overcome those problems in the QROM while SPHINCS does not require the random oracle all together.

One of the most standard constructions for signature schemes is the Full Domain Hash (FDH) paradigm. In its most basic form, the idea is to use a trapdoor one-way function f, informally a function that can be efficiently inverted only with some secret key but that can be computed with the public key available. The signature of a message $\mathbf{m}$ is a string $\mathbf{x}$ such that $f(\mathbf{x}) = \mathcal{H}(\mathbf{m})$ where $\mathcal{H}$ is a hash function, modeled in the ROM as a random function. Such a signature for $\mathbf{m}$ can be done only by a signer which has access to the secret key.

There are many constructions for signature schemes which use the FDH paradigm [BR96, CFS01, BLS04]. Some of them can be proved secure even against quantum adversaries in the QROM, for example when the security reduction is history free [BDF$^+$11]. However, those reductions are, in many case, not tight. Indeed, one usually needs to reprogram the random oracle in the security proof and this is usually costly - especially in the quantum setting. This is one of the reasons why there are so few signature schemes with concrete quantum security parameters with a quantum reduction. However, as the NIST competition arrives, it becomes increasingly important to develop signature schemes, and associated security proofs, in order to provide fully quantum-safe cryptography.

Contributions

In this work we study code-based signatures in the Full Domain Hash (FDH) paradigm. We will show under which conditions we can perform a quantum security reduction in the QROM for such schemes. While our work was strongly motivated by a recent construction of the SURF signature scheme, it can apply to different constructions, in particular to a different choice of codes and metrics like the rank metric.

We start from a family of error correcting codes $\mathcal{F}$ from which we can construct a trapdoor one way function f. The FDH paradigm then allows us to construct a signature scheme (for more details, see Section 3). We show the following results on this signature scheme:

1. We show conditions on the code family $\mathcal{F}$ used such that the resulting signature scheme is secure against quantum adversaries in the QROM. Under these conditions, we present tight security reductions to the DOOM_∞ problem [JJ02, Sen11] (the Decode One Out of Many problem), which is an already used and studied variant of the standard syndrome decoding problem, where we have the choice between many words to decode instead of a single word (the ∞ subscript indicates that we do not limit this number).
2. We perform a complete analysis of quantum algorithms for the DOOM_∞ problem, which can serve a reference for future work. The main idea here is to use the best known quantum algorithms for the 4-Sum problem and reduce DOOM_∞ to this problem.

3. We apply our security reduction to a specific signature scheme: SURF (see [DST17]). In this scheme, the family of codes $\mathcal{F}$ used there satisfies all the requirements of point 1. Thanks to our reduction, we can provide a full quantum security proof of this scheme. We get concrete parameters for which we have 128 bits of security in the QROM. We also compare the parameters of SURF with the ones of other schemes and show that it is competitive as a quantum-safe signature scheme.

In our first contribution, the security reduction, we actually manage to avoid most of the problems of the QROM. In particular, we do not reprogram the random oracle by injecting an instance of a hard problem. From a purely abstract way, this is done by considering in the FDH paradigm a one-way trapdoor function for which it is essentially as hard to find one out of many preimages. This already appeared implicitly in security reduction for FDH-like signature schemes but was handled with challenge injection and resulted in a non-tight security proof. More precisely, we will consider in this paper a one-way function f such that, for a set of random and independent elements $\{\mathbf{y}_1, \cdots, \mathbf{y}_q\}$ where we have the choice of the q we consider, it stays hard to find $(\mathbf{x}, i)$ such that $f(\mathbf{x}) = \mathbf{y}_i$. This *One Out of Many problem* is clearly easier that the problem of inversion ($q = 1$) but in this new paradigm we have seen a drastic advantage by considering it instead of performing instance injection in order to have a security proof in the QROM. Quantum security proofs seem more natural and flexible with this approach and could be used outside of code-based cryptography.

Why is it that in code-based signatures, we can afford to work on a 'One Out of Many' variant of Syndrome Decoding?

The most standard problem in code-based cryptography is the syndrome decoding (SD) problem:

Problem 1. [Syndrome Decoding - SD]
Instance: $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_2^{n-k}$, w integer
Output: $\mathbf{e} \in \mathbb{F}_2^n$ such that $|\mathbf{e}| = w$ and $\mathbf{H}\mathbf{e}^T = \mathbf{s}^T$

We instead rely on the DOOM_∞ problem:

Problem 2. [DOOM_∞]
Instance: $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$; $\mathcal{H}$ a hash function in the QROM which takes its values in $\mathbb{F}_2^{n-k}$
Output: $\mathbf{e} \in \mathbb{F}_2^n$ of Hamming weight w, $\mathbf{a} \in \mathbb{F}_2^*$ such that, $\mathbf{H}\mathbf{e}^T = \mathcal{H}(\mathbf{a})^T$

Here, we do not have a single input $\mathbf{s}$ but we can generate as many inputs $\mathcal{H}(\mathbf{a})$ as we want and we only need to solve the SD problem on one of the inputs. In the quantum setting, we even have access to a quantum oracle version of $\mathcal{H}$. It seems at first sight that this second problem is substantially easier than the first one. For example, when performing a brute force algorithm for SD, then this algorithm can be used to solve DOOM_∞ q times faster if we add q queries to $\mathcal{H}$. However, the best classical and quantum algorithms for SD are much better than the brute force algorithm. What actually happens is that the best known classical algorithms for DOOM_∞ are not that much faster than those for SD. This running time difference decreases even more when looking at parameters used in the SURF signature scheme. Moreover, the quantum setting does not offer in the state-of-the-art a fully quadratic advantage compared to classical case for solving DOOM_∞ as we will see below. This, combined with our tight security reduction, will allow us to give parameters for the SURF signature scheme for a quantum security of 128 which correspond to a classical security smaller than 256 bits.

In our second contribution, we make the above explicit for quantum algorithms as well. The SD problem has been widely studied both classically and quantumly [KT17]. There is - for most parameters - an algorithmic technique that does significantly better than others for this problem: the information set decoding technique first presented by Prange [Pra62] and then improved several times [Ste88, Dum91, MMT11, BJMM12, MO15]. Similarly, the best DOOM_∞ algorithms use the same method and the current state-of-the-art can be found in [Sen11].

The best asymptotic exponent among all those decoding techniques are [MO15, BJMM12] for SD. However, algorithm [MO15] is penalized by a big polynomial overhead which makes it more expensive that [BJMM12]. It is why in the following table we will consider asymptotic exponents given by [BJMM12]. We give in Table 1 classical exponents in base 2 of the Prange algorithm (which was the first algorithm proposed to solve syndrome decoding problem), [BJMM12] and the state-of-the-art to solve DOOM_∞ (see [Sen11]). We present the running times for $k = n/2$ and for two error weights w: namely $w \approx 0.11n$ which corresponds to the Gilbert-Varshamov weight and is the weight around which those problems are the hardest; and $w \approx 0.191n$ which corresponds to the weight used in the SURF signature scheme.

	Classical asymptotic exponent in base 2 (divided by n)		
w/n	SD (Prange)	SD ([BJMM12])	DOOM_∞ [Sen11]
0.11	0.1199	0.1000	0.0872
0.191	0.02029	0.01687	0.01654

Table 1: Asymptotic exponent for classically solving SD and DOOM_∞ for $k/n = 0.5$

The above table contains classical asymptotic exponent in base 2 (divided by n). This means for example that the Prange algorithm for SD with $w = 0.11n$ runs in time $2^{0.1199n}$.

We extend in this paper the best DOOM_∞ algorithms to the quantum setting. We first present an overview of existing algorithms and we then show that the best known quantum algorithms for DOOM_∞ are very close, in complexity to the best known quantum algorithms for SD. Table 2 compares our algorithm to the current quantum knowledge for the same range of parameters. We will come back to these tables in §6.

	Quantum asymptotic exponent in base 2 (divided by n)		
w/n	SD (Prange)	SD [KT17]	DOOM_∞(this work)
0.11	0.059958	0.058434	0.056683
0.191	0.010139	0.009218	0.009159

Table 2: Asymptotic exponent for quantumly solving SD and DOOM_∞ for $k/n = 0.5$

As we can see, the best asymptotic exponents between the SD problem and the DOOM_∞ problem are very close, especially for $w \approx 0,191$ which corresponds to the parameters of the SURF signature scheme. This allows us to greatly improve the security reduction in the QROM compared to the case where we would have used SD as a hard problem and performed challenge injection.

In our third contribution, we use the results presented above on the SURF signature scheme. As we said, there are very few signature schemes that claim quantum security. We present in table 3 security parameters for known quantum-safe (with a quantum security reduction) signature schemes. This data is taken from [ABB+17], where we added parameters for the SURF scheme obtained here.

Table 3: Security parameters for signature schemes with quantum security claims

Scheme	Quantum security (in bits)	Public key size (in kBytes)	Private key size (in kBytes)	Signature size (in kBytes)
SPHINCS	128	1	1	41
GPV-poly	59	55	26	32
GPV	59	27840	12064	30
TESLA-2	128	21799	7700	4
SURF	128	5960	3170	1.7

We only presented here signature schemes for which quantum security is provided. There are many other signature that rely on a quantum-secure computational assumption but the full parameter analysis is not provided. We refer to [ABB+17] for further details on this topic.

There is also another important metric that we do not discuss here: the running time of the different signature schemes. We did not add them here since both TESLA-2 and SURF do not have those available yet. Also the main contribution of our paper is to present an efficient security reduction, and not to compare in detail existing signature schemes.

Organisation of the paper

After presenting some notations, we provide in Section 2 a description of the quantum random oracle model. In Section 3, we present the general construction of code-based FDH signatures schemes and code-based problems. In Section 4, we present some general preliminaries as well as security notions for signature schemes. In Section 5, we present the quantum security proof in the QROM. In Section 6, we study quantum algorithms for the DOOM_∞ problem. In Section 7 we apply our security reduction to the SURF signature scheme and show concrete parameters that achieve 128 bits of quantum security. Finally in Section 8, we perform a small discussion about the obtained results, and present directions for future research.

Notations

We provide here some notation that will be used throughout the paper. Vectors will be written with bold letters (such as $\mathbf{e}$) and uppercase bold letters are used to denote matrices (such as $\mathbf{H}$). Vectors are in row notation. Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors, we will write $(\mathbf{x}|\mathbf{y})$ to denote their concatenation. The Hamming weight of $\mathbf{x}$ is denoted by $|\mathbf{x}|$. By some abuse of notation, we will use the same notation to denote the size of a finite set: $|S|$ stands for the size of the finite set S. It will be clear from the context whether $|\mathbf{x}|$ means the Hamming weight or the size of a finite set. The notation $x \stackrel{\triangle}{=} y$ means that x is defined to be equal to y. We denote by $\mathbb{F}_2^n$ the set of binary vectors of length n and S_w is its subset of words of weight w. Let S be a finite set, then $x \hookleftarrow S$ means that x is assigned to be a random element chosen uniformly at random in S. For a distribution $\mathcal{D}$ we write $\xi \sim \mathcal{D}$ to indicate that the random variable ξ is chosen according to $\mathcal{D}$.

2 The quantum random oracle model

2.1 The random oracle model - ROM.

In many signature schemes we need a function that behaves like a random function. We typically use hash functions to mimic such random functions. The random oracle model (or ROM) is an idealized model that assumes that the hash function used behaves exactly like a random function. This model is appealing as it allows simpler security proofs. There are some specific cases where the ROM is not adapted [CGH04, LN09]. Despite those examples, this model is fairly standard and accepted in the cryptographic community. Particularly, there have been no successful real-world attacks specifically because of the ROM. Additionally, schemes that are proven secure in the ROM are usually efficient.

More precisely, consider a hash function $\mathcal{H} : \{0,1\}^n \rightarrow \{0,1\}^m$ used in a cryptographic protocol. An adversary would perform an attack by applying $\mathcal{H}$ many times. Suppose the adversary makes q calls to $\mathcal{H}$ on inputs $\mathbf{x}_1, \ldots, \mathbf{x}_q$ and get answers $\mathcal{H}(\mathbf{x}_1), \ldots, \mathcal{H}(\mathbf{x}_q)$. In the ROM, this function $\mathcal{H}$ is replaced by a function f uniformly chosen from the set of functions from $\{0,1\}^n$ to $\{0,1\}^m$. This means that f outputs a random output $\mathbf{y}_i$ for every input $\mathbf{x}_i$.

Describing a random function from $\{0,1\}^n$ to $\{0,1\}^m$ requires $m2^n$ bits and cannot be hence realistically full described. Fortunately, one can emulate queries to a random function f without describing it entirely. We use the following procedure:

> On input $\mathbf{x}$, we distinguish 2 cases: if $\mathbf{x}$ was queried before then give the same answer, otherwise pick a random $\mathbf{y} \in \{0,1\}^m$ and output $\mathbf{y} = f(\mathbf{x})$.

We keep a table of the inputs that were already queried to perform the above procedure, which is efficient. This procedure is especially useful when we want to slightly modify the function f, for example by injecting the input of a computational problem as an output of f, or more generally to give a special property to f.

2.2 The quantum random oracle model - QROM.

Since we have hash functions that are believed to be secure against quantum adversaries, it is natural to extend the ROM to the quantum setting. Here again, we assume that we replace the hash function $\mathcal{H} : \{0,1\}^n \rightarrow \{0,1\}^m$ by a function f uniformly chosen from the set of functions from $\{0,1\}^n$ to $\{0,1\}^m$.

What will change compared to the classical setting is the way those functions are queried. Indeed, from the circuit $\mathcal{H}$, it is always possible to construct the unitary $O_{\mathcal{H}}$ acting on $n+m$ qubits satisfying

$$\forall \mathbf{x} \in \{0,1\}^n, \forall \mathbf{y} \in \{0,1\}^m,\ O_{\mathcal{H}}(|\mathbf{x}\rangle|\mathbf{y}\rangle) = |\mathbf{x}\rangle|\mathcal{H}(\mathbf{x}) + \mathbf{y}\rangle.$$

When replacing $\mathcal{H}$ with a random function f, queries to $O_{\mathcal{H}}$ are replaced with queries to O_f where

$$\forall \mathbf{x} \in \{0,1\}^n, \forall \mathbf{y} \in \{0,1\}^m,\ O_f(|\mathbf{x}\rangle|\mathbf{y}\rangle) = |\mathbf{x}\rangle|f(\mathbf{x}) + \mathbf{y}\rangle.$$

Again, a random function f, and the associated unitary O_f is fully determined by $m2^n$ bits corresponding to all the outcomes $f(\mathbf{x})$ for $\mathbf{x} \in \{0,1\}^n$. Unlike the classical case, there is no known procedure to efficiently produce answer to queries. Suppose for example that you want to emulate a query to O_f on input $\frac{1}{2^{n/2}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle|\mathbf{0}\rangle$. In order to emulate this, and generate $\frac{1}{2^{n/2}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle|f(\mathbf{x})\rangle$ we would need to generate some randomness $\mathbf{r}$ for each $\mathbf{x} \in \{0,1\}^n$. Another way of seeing this difficulty is that the procedure that generates a random number cannot be represented as a circuit and therefore cannot be quantized by the usual procedure.

2.3 Tweaking the QROM.

As we mentioned, it is often useful to modify the random function and to give it extra properties in order to prove the security of the underlying cryptographic scheme. The fact that we need to emulate O_f makes it much harder to include those changes in an efficient way. There are several known techniques, such as rewinding, reprogramming or challenge injection that can be done in some cases, often with a polynomial loss in the number of challenges.

Our goal was to limit as much as possible the use of those techniques in order to have the quantum security as tight as possible. The only result we will use is the following from [Zha12]:

Proposition 1. *Say $\mathcal{A}$ is a quantum algorithm that makes q quantum oracle queries. Suppose further that we draw the oracle O from two distributions. The first is the random oracle distribution. The second is the distribution of oracles where the value of the oracle at each input x is identically and independently distributed by some distribution D whose variational distance is within ε from uniform. Then the variational distance between the distributions of outputs of $\mathcal{A}$ with each oracle is at most $\frac{8\pi}{\sqrt{3}} q^{\frac{3}{2}} \sqrt{\varepsilon}$.*

3 Code-based Full Domain Hash signature schemes

We give in this section the code-based signatures schemes we will consider in our security proof in the QROM and in §3.2 code-based problems that will be involved.

3.1 Description of the scheme

Let us first recall the concept of signature schemes.

Definition 1 (Signature Scheme). *A signature scheme $\mathcal{S}$ is a triple of algorithms* `Gen`*,* `Sgn`*, and* `Vrfy` *which are defined as:*

- *The key generation algorithm* `Gen` *is a probabilistic algorithm which given 1^λ, where λ is the security parameter, outputs a pair of matching public and private keys (pk, sk);*
- *The signing algorithm is probabilistic and takes as input a message $\mathbf{m} \in \{0,1\}^*$ to be signed and returns a signature $\sigma = \texttt{Sgn}^{\mathrm{sk}}(\mathbf{m})$;*
- *The verification algorithm takes as input a message $\mathbf{m}$ and a signature σ. It returns $\texttt{Vrfy}^{\mathrm{pk}}(\mathbf{m},\sigma)$ which is 1 if the signature is accepted and 0 otherwise. It is required that $\texttt{Vrfy}^{\mathrm{pk}}(\mathbf{m},\sigma) = 1$ if $\sigma = \texttt{Sgn}^{\mathrm{sk}}(\mathbf{m})$.*

We briefly present now the code-based signatures scheme we consider. A binary linear code $\mathcal{C}$ of length n and dimension k (that we denote by $[n,k]$-code) is a subspace of $\mathbb{F}_2^n$ of dimension k and is usually defined by a parity-check matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$ of full rank as:

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{F}_2^n : \mathbf{H}\mathbf{x}^T = \mathbf{0}\}$$

Code-based signatures schemes we consider are FDH-like in which the following one way function is used:

$$\begin{aligned} f_{\mathbf{H},w} : S_w &\longrightarrow \mathbb{F}_2^{n-k} \\ \mathbf{e} &\longmapsto \mathbf{e}\mathbf{H}^T \end{aligned}$$

where $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$ is a parity-check matrix of a $[n,k]$-code. Inverting this function means on an input $\mathbf{s}$ (usually called a syndrome) to find an error $\mathbf{e}$ of Hamming weight w such that $\mathbf{H}\mathbf{e}^T = \mathbf{s}^T$. The general scheme is then defined as follows. We first suppose that we have a family of $[n,k]$-codes defined by a set of parity-check matrices $\mathcal{F}$ of size $(n-k)\times n$ such that for all $\mathbf{H} \in \mathcal{F}$ we have an algorithm $\mathcal{D}_{\mathbf{H},w}$ which on input $\mathbf{s}$ computes $\mathbf{e} \in f_{\mathbf{H},w}^{-1}(\mathbf{s})$. Then we pick uniformly at random $\mathbf{H}_{\mathrm{sec}} \in \mathcal{F}$, an $n\times n$ permutation matrix $\mathbf{P}$, a non-singular matrix $\mathbf{S} \in \mathbb{F}_2^{(n-k)\times(n-k)}$ which define the secret and public key as:

$$sk \leftarrow (\mathbf{H}_{\mathrm{sec}}, \mathbf{P}, \mathbf{S}) \ ; \ pk \leftarrow \mathbf{H}_{\mathrm{pub}} \text{ where } \mathbf{H}_{\mathrm{pub}} \stackrel{\triangle}{=} \mathbf{S}\mathbf{H}_{\mathrm{sec}}\mathbf{P}$$

This construction of $\mathbf{H}_{\mathrm{pub}}$ is the standard method to scramble a code and originates from the original work of McEliece [Mc 78].

Remark 1. Let $\mathcal{C}_{\mathrm{sec}}$ be the code defined by $\mathbf{H}_{\mathrm{sec}}$. Then the parity-check matrix $\mathbf{H}_{\mathrm{pub}}$ represents the code $\mathcal{C}_{\mathrm{pub}} \stackrel{\triangle}{=} \{\mathbf{c}\mathbf{P} : \mathbf{c} \in \mathcal{C}_{\mathrm{sec}}\}$ with a basis picked uniformly at random thanks to $\mathbf{S}$.

Then, we select a cryptographic hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_2^{n-k}$ and a parameter λ_0 which lead to define algorithms $\texttt{Sgn}^{\mathrm{sk}}$ and $\texttt{Vrfy}^{\mathrm{pk}}$ as follows

$\texttt{Sgn}^{\mathrm{sk}}(\mathbf{m})$:	$\texttt{Vrfy}^{\mathrm{pk}}(\mathbf{m},(\mathbf{e}',\mathbf{r}))$:
$\quad\mathbf{r} \hookleftarrow \{0,1\}^{\lambda_0}$	$\quad\mathbf{s} \leftarrow \mathcal{H}(\mathbf{m}\vert\mathbf{r})$
$\quad\mathbf{s} \leftarrow \mathcal{H}(\mathbf{m}\vert\mathbf{r})$	$\quad w_0 \leftarrow \vert\mathbf{e}'\vert$
$\quad\mathbf{e} \leftarrow \mathcal{D}_{\mathbf{H}_{\mathrm{sec}},w}(\mathbf{S}^{-1}\mathbf{s}^T)$	$\quad\texttt{if } \mathbf{H}_{\mathrm{pub}}\mathbf{e}'^T = \mathbf{s}^T \texttt{ and } w_0 = w \texttt{ return } 1$
$\quad\texttt{return}(\mathbf{e}\mathbf{P},\mathbf{r})$	$\quad\texttt{else return } 0$

To summarize, a signature of a message $\mathbf{m}$ with the public key $(\mathbf{H}_{\mathrm{pub}}, w)$ is a pair $(\mathbf{e},\mathbf{r})$ such that $\mathbf{H}_{\mathrm{pub}}\mathbf{e}^T = \mathcal{H}(\mathbf{m}|\mathbf{r})^T$ with $|\mathbf{e}| = w$

Remark 2. The use of a salt $\mathbf{r} \in \{0,1\}^{\lambda_0}$ in algorithm $\texttt{Sgn}^{\mathrm{sk}}$ is made in order to have a tight security proof. In particular, this allows two signatures of a message $\mathbf{m}$ to be different with high probability.

3.2 Code-Based Problems and computational assumtpions

We introduce in this subsection the code-based problems on which our security reduction in the QROM will stand. The first is Decoding One Out of Many (DOOM). Its classical version was first considered in [JJ02] and later analyzed in [Sen11]. We will come back to its analysis in the quantum case in §6. As we are going to see, the best known algorithms to solve this problem are functions of the distance w. Let us first consider the basic problem upon which all code-based cryptography relies.

Problem 1. [Syndrome Decoding - SD]
Instance: $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_2^{n-k}$, w integer
Output: $\mathbf{e} \in \mathbb{F}_2^n$ such that $|\mathbf{e}| = w$ and $\mathbf{He}^T = \mathbf{s}^T$

This problem has been studied for a long time and despite many efforts on this issue [Pra62, Ste88, Dum91, MMT11, BJMM12, MO15, DT17] the best known algorithms for solving this problem [BJMM12, MO15] are exponential in the weight w of $\mathbf{e}$ as long as $w = (1-\epsilon)(n-k)/2$ for any $\epsilon > 0$. Furthermore when w is sublinear in n, the exponent of the best known algorithms has not changed [CTS16] since the Prange algorithm [Pra62] dating back to the early sixties. Moreover, it seems very difficult to lower this exponent by a multiplicative factor smaller than $\frac{1}{2}$ in the quantum computation model as illustrated by [KT17].

However, in a context of code-based signatures an attacker may produce, say q, favorable messages and hash them to obtain $s_1, \cdots, s_q$ syndromes on which he tries to solve one of the q instances associated to Problem 1. This brings us to introduce a different version of the SD problem.

Problem 3 (DOOM_q *– Decoding One Out of Many).*
Instance: $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$; $\mathbf{s}_1, \cdots, \mathbf{s}_q \in \mathbb{F}_2^{n-k}$; $w \in \{0, \cdots, n\}$
Output: $(\mathbf{e}, i) \in \mathbb{F}_2^n \times [\![1, q]\!]$ of Hamming weight w such that $\mathbf{He}^T = \mathbf{s}_i^T$.

The above problem can be defined for any $q \in \mathbb{N}^*$. This problem is of course easier than SD but can not be solved at most q times faster than the SD problem. As it happens best algorithm gain much less than this q factor. Also using the hardness of DOOM_q is appealing when performing security proofs in the QROM as it allows to avoid instance injection.

Moreover, an interesting feature of the above problem is that known algorithms to solve it fail to take advantage of very large values of q. Actually, depending on the parameters, there is a limit after which increasing q does n'ot decrease the time. Therefore, it is natural to define a variant where we do not limit a priori q. We also require the inputs $\mathbf{s}_i$ to be the output of a random function instead of requiring to write them all. This allows to have a compact description of the inputs. This will also simplify the quantum security proof.

Problem 2. [DOOM_∞]
Instance: $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$; $\mathcal{H}$ a hash function in the QROM which takes its values in $\mathbb{F}_2^{n-k}$
Output: $\mathbf{e} \in \mathbb{F}_2^n$ of Hamming weight w, $\mathbf{a} \in \mathbb{F}_2^*$ such that, $\mathbf{He}^T = \mathcal{H}(\mathbf{a})^T$

We study those problems in §6. In the classical setting, we can easily see that those problems are equivalent for sufficiently large values of q. We also present there the best known quantum algorithms for DOOM_∞.

Definition 2 (One-Wayness of DOOM_∞**).** *We define the quantum success of an algorithm* $\mathcal{A}$ *against* DOOM_∞ *with the parameters* n, k, w *as:*

$$QSucc^{n,k,w}_{\mathrm{DOOM}_\infty}(\mathcal{A}) = \mathbb{P}\big(\mathcal{A}(\mathbf{H}, \mathcal{H}) \text{ solution}\big).$$

where $\mathbf{H}$ *is chosen uniformly at random in* $\mathbb{F}_2^{(n-k)\times n}$*,* $\mathcal{H}$ *a hash function in the* QROM *which takes its values in* $\mathbb{F}_2^{n-k}$ *and the probability is taken over these choices of* $\mathbf{H}$ *and internal coins of* $\mathcal{A}$*.*

The quantum computational success in time t of breaking DOOM_∞ *with the parameters n, k, w is then defined as:*

$$QSucc^{n,k,w}_{\mathrm{DOOM}_\infty}(t) = \max_{|\mathcal{A}|\leq t} \left\{ QSucc^{n,k,w}_{\mathrm{DOOM}_\infty}(\mathcal{A}) \right\}$$

As we discussed in the introduction, it is appealing to consider the DOOM_∞ problem as it will greatly improve our security reduction on the one side, but on the other side remains almost as hard as the SD problem.

4 Basic security definitions

4.1 Basic definitions

A function $f(n)$ is said to be negligible if for all polynomials $p(n)$, $|f(n)| < p(n)^{-1}$ for all sufficiently large n. We will denote $negl(n)$ the set of negligible functions. The statistical distance between two discrete probability distributions over a same space $\mathcal{E}$ is defined as:

$$\rho(\mathcal{D}^0, \mathcal{D}^1) \stackrel{\triangle}{=} \frac{1}{2} \sum_{x \in \mathcal{E}} |\mathcal{D}^0(x) - \mathcal{D}^1(x)|.$$

The following classical proposition on the statistical distance will be useful:

Proposition 2. *Let $(\mathcal{D}^0_1, \dots, \mathcal{D}^0_n)$ and $(\mathcal{D}^1_1, \dots, \mathcal{D}^1_n)$ be two n-tuples of discrete probability distributions where $\mathcal{D}^0_i$ and $\mathcal{D}^1_i$ are distributed over a same space $\mathcal{E}_i$. For $a \in \{0,1\}$, let us denote by $\mathcal{D}^a_1 \otimes \cdots \otimes \mathcal{D}^a_n$ the product probability distribution of $\mathcal{D}^a_1, \dots, \mathcal{D}^a_n$, that is $\mathcal{D}^a_1 \otimes \cdots \otimes \mathcal{D}^a_n(x_1, \dots, x_n) = \mathcal{D}^a_1(x_1) \dots \mathcal{D}^a_n(x_n)$ with $x_i \in \mathcal{E}_i$ for $i \in \{1, \dots, n\}$. In such a case we have*

$$\rho\left(\mathcal{D}^0_1 \otimes \cdots \otimes \mathcal{D}^0_n, \mathcal{D}^1_1 \otimes \cdots \otimes \mathcal{D}^1_n\right) \leq \sum_{i=1}^{n} \rho(\mathcal{D}^0_i, \mathcal{D}^1_i).$$

A distinguisher between two distributions $\mathcal{D}^0$ and $\mathcal{D}^1$ over the same space $\mathcal{E}$ is a randomized algorithm $\mathcal{A}$ which takes as input an element of $\mathcal{E}$ that follows the distribution $\mathcal{D}^0$ or $\mathcal{D}^1$ outputs $b \in \{0,1\}$. Such an $\mathcal{A}$ is characterized by its advantage:

$$Adv^{\mathcal{D}^0,\mathcal{D}^1}(\mathcal{A}) \stackrel{\triangle}{=} \mathbb{P}_{\xi \sim \mathcal{D}^0}\left(\mathcal{A}(\xi) \text{ outputs } 1\right) - \mathbb{P}_{\xi \sim \mathcal{D}^1}\left(\mathcal{A}(\xi) \text{ outputs } 1\right)$$

where $\mathbb{P}_{\xi \sim \mathcal{D}^i}\left(\mathcal{A}(\xi) \text{ outputs } 1\right)$ is the probability that $\mathcal{A}(\xi)$ outputs 1 when its inputs are picked according to the distribution $\mathcal{D}^i$ and for each executions its internal coins are picked uniformly at random. We call this quantity the advantage of $\mathcal{A}$ against $\mathcal{D}^0$ and $\mathcal{D}^1$.

Definition 3 (Quantum Computational Distance and Indistinguishability). *The quantum computational distance between two distributions $\mathcal{D}^0$ and $\mathcal{D}^1$ in time t is:*

$$\rho_{Qc}\left(\mathcal{D}^0, \mathcal{D}^1\right)(t) \stackrel{\triangle}{=} \max_{|\mathcal{A}|\leq t} \left\{ Adv^{\mathcal{D}^0,\mathcal{D}^1}(\mathcal{A}) \right\}$$

where $|\mathcal{A}|$ denotes the running time of $\mathcal{A}$ on its inputs.

The ensembles $\mathcal{D}^0 = (\mathcal{D}^0_n)$ and $\mathcal{D}^1 = (\mathcal{D}^1_n)$ are computationally indistinguishable in time (t_n) if their computational distance in time (t_n) is negligible in n.

4.2 Digital signature security and games.

For signature schemes one of the strongest security notion is *Quantum Existential Unforgeability under an adaptive Chosen Message Attack* (QEUF-CMA). In other words, a quantum adversary has access to any signatures of its choice and its goal is to produce a valid forgery. A valid forgery is a message/signature pair $(\mathbf{m}, \sigma)$ such that $\mathtt{Vrfy}^{\mathrm{pk}}(\mathbf{m}, \sigma) = 1$ whereas the signature of $\mathbf{m}$ has never been requested by the forger. Moreover the forger has access to quantum hash queries. By quantum hash queries we mean that adversaries can make a superposition of queries. In other words, a quantum access to a hash function $\mathcal{H}$ is an access to the following oracle:

$$O_{\mathcal{H}} : |\mathbf{m}, \mathbf{z}\rangle \mapsto |\mathbf{m}, \mathbf{z} \oplus \mathcal{H}(\mathbf{m})\rangle$$

Let us now define the QEUF-CMA security of a signature scheme:

Definition 4 (QEUF-CMA Security). *Let $\mathcal{S}$ be a signature scheme.*
A forger $\mathcal{A}$ is a $(t, q_{\text{hash}}, q_{\text{sign}}, \varepsilon)$-adversary in QEUF-CMA *against $\mathcal{S}$ if after at most q_{hash} quantum-queries to the hash oracle, q_{sign} classical-queries to signing oracle and t working time, it outputs a valid forgery with probability at least ε. We define the* QEUF-CMA *success probability against $\mathcal{S}$ as:*

$$QSucc_{\mathcal{S}}^{\text{QEUF-CMA}}(t, q_{\text{hash}}, q_{\text{sign}}) \stackrel{\triangle}{=} \max\left(\varepsilon | \textit{it exists a } (t, q_{\text{hash}}, q_{\text{sign}}, \varepsilon)\textit{-adversary}\right).$$

The signature scheme $\mathcal{S}$ is said to be $(t, q_{\text{hash}}, q_{\text{sign}})$-secure in QEUF-CMA *if the above success probability is a negligible function of the security parameter λ.*

In order to prove that a signature scheme is QEUF-CMA under some assumptions we will use the paradigm of games. A good reference of this topic can be found in [Sho04]. The following game gives the QEUF-CMA security:

Definition 5 (challenger procedures in the QEUF-CMA Game). *Challenger procedures for the* QEUF-CMA *Game corresponding to a signature scheme $\mathcal{S}$ are defined as:*

proc Initialize(λ)	proc Hash($\mathbf{m}, \mathbf{r}$)	proc Sign($\mathbf{m}$)	proc Finalize($\mathbf{m}, \sigma$)
$(pk, sk) \leftarrow$ Gen(1^λ) return pk	return $\mathcal{H}(\mathbf{m})$	return Sgn$^{\text{sk}}(\mathbf{m})$	return (Vrfy$^{\text{pk}}(\mathbf{m}, \sigma) = 1$)

5 Quantum security of FDH-like code-base signature schemes

In this section, we show that code-based signature schemes we defined in §3 are QEUF-CMA in the QROM against quantum adversaries. We redescribe the most important aspects of the scheme S_{code} defined in §3 so that the proof is easier to follow.

We have a family of $[n, k]$-codes defined by a set of parity-check matrices $\mathcal{F}$ of size $(n-k) \times n$ such that for all $\mathbf{H} \in \mathcal{F}$ we have an algorithm $\mathcal{D}_{\mathbf{H},w}$ which on input $\mathbf{s}$ computes $\mathbf{e} \in f_{\mathbf{H},w}^{-1}(\mathbf{s})$ where $f_{\mathbf{H},w}$ is the function such that $f_{\mathbf{H},w}(\mathbf{e}) = \mathbf{e}\mathbf{H}^T$. Then we pick uniformly at random $\mathbf{H}_{\text{sec}} \in \mathcal{F}$, an $n \times n$ permutation matrix $\mathbf{P}$, a non-singular matrix $\mathbf{S} \in \mathbb{F}_2^{(n-k)\times(n-k)}$. The secret and public key are:

$$sk \leftarrow (\mathbf{H}_{\text{sec}}, \mathbf{P}, \mathbf{S}) \ ; \ pk \leftarrow \mathbf{H}_{\text{pub}} \text{ where } \mathbf{H}_{\text{pub}} \stackrel{\triangle}{=} \mathbf{S}\mathbf{H}_{\text{sec}}\mathbf{P}$$

The signing and verification procedures are then the following

Sgn$^{\text{sk}}(\mathbf{m})$: $\mathbf{r} \hookleftarrow \{0,1\}^{\lambda_0}$ $\mathbf{s} \leftarrow \mathcal{H}(\mathbf{m}\vert\mathbf{r})$ $\mathbf{e} \leftarrow \mathcal{D}_{\mathbf{H}_{\text{sec}},w}(\mathbf{S}^{-1}\mathbf{s}^T)$ return($\mathbf{e}\mathbf{P}, \mathbf{r}$)	Vrfy$^{\text{pk}}(\mathbf{m}, (\mathbf{e}', \mathbf{r}))$: $\mathbf{s} \leftarrow \mathcal{H}(\mathbf{m}\vert\mathbf{r})$ $w_0 \leftarrow \vert\mathbf{e}'\vert$ if $\mathbf{H}_{\text{pub}}\mathbf{e}'^T = \mathbf{s}^T$ and $w_0 = w$ return 1 else return 0

Let us first recall and give definitions of distributions that will be used:

- $\mathcal{U}_w$ is the uniform distribution over S_w (words of weight w).
- $\mathcal{U}_{n-k}$ is the uniform distribution over $\mathbb{F}_2^{n-k}$.
- $\mathcal{D}_w$ is the distribution of $\mathcal{D}_{\mathbf{H}_{\text{sec}},w}(\mathbf{S}^{-1}\mathbf{s}^T)$ when $\mathbf{s} \hookleftarrow \mathbb{F}_2^{n-k}$ where $\mathcal{D}_{\mathbf{H}_{\text{sec}},w}(\cdot)$ is the algorithm used in $\mathcal{S}_{\text{code}}$ to invert $\mathbf{e} \in S_w \mapsto \mathbf{e}\mathbf{H}_{\text{sec}}^T$.

- $\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}$ is the distribution of the syndrome $\mathbf{H}_{\text{pub}}\mathbf{e}^T$ where $\mathbf{e}$ is drawn uniformly at random in S_w
- $\mathcal{D}_{\text{pub}}$ is the distribution of public keys $\mathbf{H}_{\text{pub}}$.
- $\mathcal{D}_{\text{rand}}$ is the uniform distribution over parity-check matrices of size $(n-k) \times n$.

Our main security statement is the following

Theorem 1 (Security Reduction). *Let $\mathcal{S}_{code}$ be the signature scheme defined in §3 with security parameter λ. Let q_{hash} (resp. q_{sign}) be the number of queries to the hash (resp. signing) oracle. We also take $\lambda_0 = \lambda + 2\log_2(q_{\text{sign}})$. For any running time t we have*

$$QSucc_{\mathcal{S}_{code}}^{\text{QEUF-CMA}}(t, q_{\text{hash}}, q_{\text{sign}}) \leq 2 \cdot QSucc_{\text{DOOM}_\infty}^{n,k,w}(2t) + \rho_{Qc}\left(\mathcal{D}_{\text{pub}}, \mathcal{D}_{\text{rand}}\right)(2t) + \frac{8\pi}{\sqrt{3}} q_{\text{hash}}^{\frac{3}{2}} \sqrt{\mathbb{E}_{\mathbf{H}_{\text{pub}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_{n-k})\right)} + q_{\text{sign}}\rho\left(\mathcal{U}_w, \mathcal{D}_w\right) + \frac{1}{2^\lambda}$$

In other words, signature schemes we introduced in §3 can be reduced to the hardness of DOOM_∞ in the QROM if the family $\mathcal{F}$ and the signature scheme satisfy the following conditions:

Condition 1

1. $\frac{8\pi}{\sqrt{3}} q_{\text{hash}}^{\frac{3}{2}} \sqrt{\mathbb{E}_{\mathbf{H}_{\text{pub}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_{n-k})\right)}) \in negl(\lambda)$
2. $q_{\text{sign}}\rho\left(\mathcal{U}_w, \mathcal{D}_w\right) \in negl(\lambda)$
3. $\rho_{Qc}\left(\mathcal{D}_{\text{pub}}, \mathcal{D}_{\text{rand}}\right)(t) = o(\frac{t}{2^\lambda})$.

The two first properties are properties of the code family $\mathcal{F}$ used while the third property is a property on the signing algorithms used: we require that signatures which are produced are indistinguishable from words uniformly and independently picked in S_w.

Notice that our security reduction is almost tight if the above holds. Indeed, we double the running and lose a factor 2 in front of $QSucc_{\text{DOOM}_\infty}^{\text{QEUF-CMA}}(t, q_{\text{hash}}, q_{\text{sign}})$. This makes us lose 2 bits of security. Actually, we could have a really tight reduction but it would involve a huge amount of quantum memory and access to quantum RAM. We wanted to construct an algorithm in our reduction in the most efficient way so we avoided this solution. We discuss this more at the end of the section.

The goal of what follows is to prove Theorem 1. Our security reduction will go as follows: let $\mathcal{A}$ be a $(t, q_{\text{sign}}, q_{\text{hash}}, \varepsilon)$-quantum adversary in the QEUF-CMA model against $\mathcal{S}_{\text{code}}$. Recall that in the QEUF-CMA model, we have a benign challenger and the following procedures

proc Initialize(λ)	proc Hash($\mathbf{m}, \mathbf{r}$)	proc Sign($\mathbf{m}$)	proc Finalize($\mathbf{m}, \mathbf{e}, \mathbf{r}$)
$(pk, sk, \lambda_0) \leftarrow \texttt{Gen}(1^\lambda)$	return $\mathcal{H}(\mathbf{m}\vert\mathbf{r})$	$\mathbf{r} \hookleftarrow \{0,1\}^{\lambda_0}$	$\mathbf{s} \leftarrow \texttt{Hash}(\mathbf{m}, \mathbf{r})$
$sk \leftarrow (\mathbf{P}, \mathbf{S}, \mathbf{H}_{\text{sec}})$		$\mathbf{s} \leftarrow \texttt{Hash}(\mathbf{m}, \mathbf{r})$	return
$pk \leftarrow (\mathbf{H}_{\text{pub}} \triangleq \mathbf{S}\mathbf{H}_{\text{sec}}\mathbf{P}$		$\mathbf{e} \leftarrow \mathcal{D}_{\mathbf{H}_{\text{sec}},w}(\mathbf{S}^{-1}\mathbf{s}^T)$	$\mathbf{H}_{\text{pub}}\mathbf{e}^T = \mathbf{s}^T \wedge \vert\mathbf{e}\vert = w$
return $(\mathbf{H}_{\text{pub}}, w)$		return $(\mathbf{e}\mathbf{P}, \mathbf{r})$	

In this model, $\mathcal{A}$ performs the following actions, that we model by a game:

Game 0

1. $\mathcal{A}$ makes a call to Initialize(λ) and receives $\mathbf{H}_{\text{pub}}$.
2. $\mathcal{A}$ performs q_{sign} calls to the Sign procedure. Let $\mathbf{m}_i$ the message that $\mathcal{A}$ wants to sign at query i and let σ_i the corresponding signature answered by the challenger.
3. $\mathcal{A}$ performs an algorithm that makes q_{hash} calls to Hash and outputs $\mathbf{m}', \mathbf{e}', \mathbf{r}'$
4. $\mathcal{A}$ wins if $\forall i,\ \mathbf{m}_i \neq \mathbf{m}'$ and Finalize($\mathbf{m}', \mathbf{e}', \mathbf{r}'$) $= 1$. This happens with probability ε and the whole running time is t.

Recall that procedure Sign is done by the challenger and $\mathcal{A}$ queries the challenger. $\mathcal{A}$ does not have access to the secret key and cannot run Sign by himself. Procedure Hash is public, efficient and is used both by the challenger and the adversary $\mathcal{A}$.

Our security reduction will go as follows: from the adversary $\mathcal{A}$, we will construct an algorithm $\mathcal{B}$ to solve the DOOM$_\infty$ problem. The main part of the proof will be to replace the hash function $\mathcal{H}$ (modeled by a random function from the QROM) by another hash function that we call Z. In Subsection 5.1 we show how to construct this function and in Subsection 5.2 we prove our main security statement.

5.1 Constructing the hash function Z

Informally, we want the following properties for Z:

1. Z is statistically close to a random function in the QROM.
2. Z and O_Z can be computed efficiently
3. For any message $\mathbf{m}$, there is an efficient algorithm to construct $\mathbf{r} \in \mathbb{F}_2^{\lambda_0}$ and $\mathbf{e} \in S_w$ such that $Z(\mathbf{m}, \mathbf{r}) = \mathbf{H}_{\text{pub}}\mathbf{e}^T$ *without knowing the secret key* $\mathbf{S}, \mathbf{P}, \mathbf{H}_{\text{sec}}$.
4. With constant probability, $Z(\mathbf{m}, \mathbf{r}) = \mathcal{H}(\mathbf{m}, \mathbf{r})$.

The first two properties will allow us to replace calls to $O_{\mathcal{H}}$ with calls to O_Z in $\mathcal{A}$ without changing much the statistical distance of the output. The third property will then allow to change the signing oracle into one that can be done locally without knowing the secret key. The final property will still enforce that the algorithm $\mathcal{B}$ we construct indeed solves the DOOM$_\infty$ problem.

Construction of Z. Let J be a cryptographic hash function that takes its values in $\mathbb{F}_2 \times S_w$. In particular, the first bit of $J(\mathbf{m}, \mathbf{r})$ is a random element of $\mathbb{F}_2$. From the functions J and $\mathcal{H}$ we can build the function $Z : \mathbb{F}_2^* \to \mathbb{F}_2^n$ as follows: fix an input $(\mathbf{m}, \mathbf{r})$ and let $(b, \mathbf{e}) = J(\mathbf{m}, \mathbf{r})$. If $b = 0$ then $Z(\mathbf{m}, \mathbf{r}) = \mathcal{H}(\mathbf{m}, \mathbf{r})$ else $Z(\mathbf{m}, \mathbf{r}) = \mathbf{H}_{\text{pub}}\mathbf{e}^T$. We can easily construct an efficient quantum circuit for O_Z using $O_{\mathcal{H}}$ and O_J. For the running time of O_Z, we assume that the running time of $\mathcal{H}$ is roughly equivalent to the computing time of $(\mathbf{H}_{\text{pub}}\mathbf{e}^T)$ (if this is not the case, we can use a slower hash function $\mathcal{H}$ to match those 2 times).

Proposition 3. *For any* $\mathbf{H}_{\text{pub}}$*, outputs of* Z *are at most at statistical distance* $\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_{n-k})$ *to outputs of a random function in the QROM.*

Proof. It directly follows from the definition of Z and $\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}$ given above. Indeed, for any input $(\mathbf{m}, \mathbf{r})$, if $J(\mathbf{m}, \mathbf{r}) = (0, \mathbf{e})$ then the output distribution is totally random and equal to $\mathcal{U}_{n-k}$. Otherwise, it follows the distribution of $\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}$. Each of these events happens with probability $\frac{1}{2}$ which concludes the proof.

Moreover, for any message $\mathbf{m}$, we can find $\mathbf{r} \in \mathbb{F}_2^{\lambda_0}$ and $\mathbf{e} \in S_w$ such that $Z(\mathbf{m}, \mathbf{r}) = \mathbf{H}_{\text{pub}}\mathbf{e}^T$ with the following procedure: find $\mathbf{r}_0$ such that $J(\mathbf{m}, \mathbf{r}_0) = (b, \mathbf{e}_0)$ with $b = 1$. This means that the running time of O_Z is twice the running time of $O_{\mathcal{H}}$. This can be done with 2 calls to J on average. Output $\mathbf{r}_0, \mathbf{e}_0$ and notice that $\mathbf{e}_0 = Z(\mathbf{m}, \mathbf{r}_0)$.

5.2 Proof of Theorem 1

Proof. We present a sequence of games which initiates with Game 0 presented at the beginning of this section and ends with an quantum algorithm solving the DOOM$_\infty$ problem. Let $(\mathbf{H}_0, \mathcal{H})$ be an instance of the DOOM$_\infty$-problem for parameters n, k, w given by $\mathcal{S}_{\text{code}}$. We will denote by $\mathbb{P}(S_i)$ the probability of success of the game i.

Game 1 is identical to Game 0 except that we change the winning condition. Let F be the following failing event: there is a collision in a signature query (*i.e.* two signatures queries for a same message $\mathbf{m}$ lead to the same salt $\mathbf{r}$). The adversary wins Game 1 only if F does not occur additionally to the other requirements. A direct application of the birthday paradox gives $\mathbb{P}(F) \leq \frac{1}{2^\lambda}$ and

$$\mathbb{P}(S_0) \leq \mathbb{P}(S_1) - \mathbb{P}(F) \leq \mathbb{P}(S_1) - \frac{1}{2^\lambda}.$$

Game 2. Here, we consider Game 1 but both the adversary and the challenger use a different procedure `Hash`. The $\mathsf{Hash}(\mathbf{m}, \mathbf{r})$ procedure hence becomes: return $Z(\mathbf{m}, \mathbf{r})$. A call to $O_{\mathsf{Hash}}(|\psi\rangle)$ returns similarly $O_Z(|\psi\rangle)$ for all $|\psi\rangle$. We can relate this game to the previous one through the following lemma.

Lemma 1.

$$\mathbb{P}(S_1) \leq \mathbb{P}(S_2) + \frac{8\pi}{\sqrt{3}} q_{\text{hash}}^{\frac{3}{2}} \sqrt{\mathbb{E}_{\mathbf{H}_{\text{pub}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_{n-k})\right)}$$

Proof. It is clear that $\mathbb{P}(S_1) - \mathbb{P}(S_2) = \mathbb{E}_{\mathbf{H}_{\text{pub}}}\left(\mathbb{P}(S_1|\mathbf{H}_{\text{pub}}) - \mathbb{P}(S_2|\mathbf{H}_{\text{pub}})\right)$. Moreover if we fix $\mathbf{H}_{\text{pub}}$, we know from Proposition 3 that in the *QROM*, outputs of Z are at most at distance $\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_{n-k})$ from uniform. Game 2 differs from game 1 by replacing each call to Hash (resp. O_{Hash}) by a call to Z (resp. O_Z). Using Proposition 1, the output state after game 2 differs (in statistical distance) from the output state after game 1 by at most $\frac{8\pi}{\sqrt{3}} q_{\text{hash}}^{\frac{3}{2}} \sqrt{\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_w)}$ which leads to:

$$\mathbb{P}(S_1|\mathbf{H}_{\text{pub}}) - \mathbb{P}(S_2|\mathbf{H}_{\text{pub}}) \leq \frac{8\pi}{\sqrt{3}} q_{\text{hash}}^{\frac{3}{2}} \sqrt{\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_w)}$$

Then by concavity of the root function and Jensen's inequality we get:

$$\mathbb{P}(S_1) - \mathbb{P}(S_2) \leq \frac{8\pi}{\sqrt{3}} q_{\text{hash}}^{\frac{3}{2}} \sqrt{\mathbb{E}_{\mathbf{H}_{\text{pub}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_w)\right)}$$

Game 3 differs from Game 2 by changing in `proc Sign`. When it is queried $\mathbf{m}$, the procedure "$\mathbf{e} \leftarrow \mathcal{D}_{\mathbf{H}_{\text{sec}}, w}(\mathbf{S}^{-1}\mathbf{s}^T)$, return $(\mathbf{eP}, \mathbf{r})$" is replaced by "find $(\mathbf{e}, \mathbf{r})$ such that $J(\mathbf{m}, \mathbf{r}) = (1, \mathbf{e})$, return $(\mathbf{e}, \mathbf{r})$".

Any signature $(\mathbf{e}, \mathbf{r})$ produced by `proc Sign` is valid. J is modeled as a random function so the error $\mathbf{e}$ is drawn according to the uniform distribution $\mathcal{U}_w$ while previously it was drawn according to the output distribution of $\mathcal{D}_{\mathbf{H}_{\text{sec}}, w}$. We therefore have thanks to Proposition 2

$$\mathbb{P}(S_2) - \mathbb{P}(S_3) \leq q_{\text{sign}} \rho(\mathcal{U}_w, \mathcal{D}_w)$$

Moreover, to find $\mathbf{r}$ such that $J(\mathbf{m}, \mathbf{r}) = (1, \cdot)$ we pick uniformly at random $\mathbf{r}$ until finding it. As outputs of J are uniformly distributed, we find such a $\mathbf{r}$ in a constant time.

Game 4 is the game where in the initialize procedure, we replace the public matrix $\mathbf{H}_{\text{pub}}$ by $\mathbf{H}_0$, which is a totally random matrix in $\mathbb{F}_2^{(n-k)\times n}$. In this way we will force the adversary to build a solution of the DOOM_∞ problem. Here if a difference is detected between games it gives a distinguisher between the distribution $\mathcal{D}_{\text{rand}}$ and $\mathcal{D}_{\text{pub}}$:

$$\mathbb{P}(S_3) \leq \mathbb{P}(S_4) + \rho_{Qc}(\mathcal{D}_{\text{pub}}, \mathcal{D}_{\text{rand}})(2t).$$

Game 5 differs in the finalize procedure as follows:

`proc Finalize`$(\mathbf{m}, \mathbf{e}, \mathbf{r})$
$\mathbf{s} \leftarrow \mathtt{Hash}(\mathbf{m}, \mathbf{r})$
$b \leftarrow \mathbf{H}_{\text{pub}}\mathbf{e}^T = \mathbf{s}^T = 0 \wedge \lvert\mathbf{e}\rvert = w$
$(b', \mathbf{e}) = J(\mathbf{m}, \mathbf{r})$
return $b \wedge (b' == 0)$

We assume the forger outputs a valid signature $(\mathbf{e}, \mathbf{r})$ for the message $\mathbf{m}$. The probability of success of Game 5 is the probability of the event "$S_4 \wedge (J(\mathbf{m}, \mathbf{r}) = (0, \mathbf{e}))$".

If the forgery is valid, the message $\mathbf{m}$ has never been queried by `Sign`, and the adversary never had access to any output of $J(\mathbf{m}, \cdot)$. This way, the two events are independent and we get:

$$\mathbb{P}(S_5) = \mathbb{P}_{\mathbf{m},\mathbf{r}}(J(\mathbf{m}, \mathbf{r}) = (0, \mathbf{e})) \cdot \mathbb{P}(S_4) = \frac{1}{2}\mathbb{P}(S_4).$$

The probability $\mathbb{P}(S_5)$ is then exactly the probability for $\mathcal{A}$ to output $\mathbf{m}, \mathbf{r}$ and $\mathbf{e} \in S_w$ such that $\mathbf{H}_0\mathbf{e}^T = \mathcal{H}(\mathbf{m}, \mathbf{r})^T$ which gives

$$\mathbb{P}(S_5) \leq QSucc^{n,k,w}_{\mathrm{DOOM}_\infty}(2t).$$

as we know thanks to the output a preimage $(\mathbf{m}, \mathbf{r})$ of the solution of the decoding problem. This concludes the proof of Theorem 1 by combining this together with all the bounds obtained for each of the previous games.

Why do use the random function Z to reprogram our random oracle?

We just want to briefly mention why we use an extra function J to reprogram our (quantum) random oracle. We could have just, for the q values we use, reprogram the function $\mathcal{H}$ accordingly, as it is done for example in [ABB+17]. However, this actually requires q extra quantum bits of memory (recall that $q = 2^\lambda$ and can be very large) as well as an efficient quantum data structure that would act as a quantum RAM. However, we do not have yet efficient models of quantum RAM, as shown in [AGJO+15]. We do not want to go to deep in the discussion whether such data structures in the quantum model should be allowed or not, this is work for future research. However, we want to be in the safe side of things by not allowing here this kind of data structures. This means in particular that our reduction from the adversary $\mathcal{A}$ that breaks the signature scheme to the algorithm $\mathcal{B}$ that solves the DOOM_∞ problem not only preserves essentially the quantum time but also more generally the quantum resources used, in particular quantum memory.

6 DOOM_∞ Study

We study here the best known quantum algorithms to solve DOOM_∞. They all come from an old algorithm due to Prange [Pra62] and are known as Information Set Decoding (ISD). These kind of algorithms were first thought to solve the SD problem. The current state-of-the-art to solve the DOOM_q and DOOM_∞ slightly adapt them. In this way we are first going to describe general a skeleton of ISDs and quantum algorithms in this setting. Moreover, during our discussion we will give several reasons on why we think it is difficult to improve significantly quantum algorithms using ISDs.

Notations We provide here some notations that will be used throughout this section. Let $\mathbf{H}$ be a matrix of size $(n-k) \times n$ in $\mathbb{F}_2$ and $I = \{i_1, \cdots, i_p\} \subseteq \{1, \cdots, n\}$. We define the permutation π_I as:

$$\pi_I(i_j) = j \text{ for } 1 \leq j \leq p \text{ and } \pi_I(j) = j \text{ otherwise}$$

and $\mathbf{P}_{\pi_I}$ its associated matrix. Then $\mathbf{H}_{\pi_I}$ will denote $\mathbf{H}\mathbf{P}_{\pi_I}$. All quantities we are interested in are functions of the code-length n and we will write $f(n) = \tilde{O}(g(n))$ when there exists a constant C such that $f(n) = O\left(\log_2^C(g(n)) \cdot g(n)\right)$ and $f(n) = \Theta(g(n))$ when there exists two constants m, M such that $mg(n) \leq f(n) \leq Mg(n)$.

6.1 Information Set Decoding - ISD

Let us first recall that algorithms we will study were thought to solve the following problem:

Problem 1. [Syndrome Decoding - SD]
Instance: $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_2^{n-k}$, w integer
Output: $\mathbf{e} \in \mathbb{F}_2^n$ such that $|\mathbf{e}| = w$ and $\mathbf{H}\mathbf{e}^T = \mathbf{s}^T$

Existing literature in the study of algorithms solving SD usually assumes that there is a unique solution as for instance in a context of encryption the ciphertext of $\mathbf{e}$ is $\mathbf{H}\mathbf{e}^T$ (see [Nie86]) which imposes to have an injective construction. In the case of code-based signature schemes we introduced in §3.1, the weight w is chosen greater than the Gilbert-Varshamov bound, namely $d_{\mathrm{GV}}(n,k) \stackrel{\triangle}{=} nh^{-1}(1-k/n)$ where $h(x) \stackrel{\triangle}{=} -x\log_2(x) - (1-x)\log_2(1-x)$ and $h^{-1}(x)$ is the inverse function defined for x in $[0, \frac{1}{2}]$ and ranging over $[0,1]$. It represents the weight w for which we can typically expect that SD admits one solution, beyond it there typically exits an exponential number of solutions and below it no solution. We need to choose w greater than this bound in order to be able to invert the function $\mathbf{e} \in S_w \mapsto \mathbf{e}\mathbf{H}^T$ on all words of $\mathbb{F}_2^{n-k}$. More precisely, the following proposition gives the number of solutions which are expected:

Proposition 4. *Let w be an integer and $\mathbf{s} \in \mathbb{F}_2^{n-k}$, then there exists in average $M_{n,k,w} \stackrel{\triangle}{=} \frac{\binom{n}{w}}{2^{n-k}}$ solutions to* SD *where probabilities are taken by picking matrices $\mathbf{H}$ uniformly at random in $\mathbb{F}_2^{(n-k)\times n}$.*

Remark 3. Asymptotically $\binom{n}{w} = \tilde{O}\left(2^{n\cdot h(w/n)}\right)$, then Gilbert-Varshamov's bound easily gives the weight for which we expect in average one solution to SD.

In the following we will consider weights w greater than $d_{\mathrm{GV}}(n,k)$ and we will have to take into account $M_{n,k,w}$ in our study.

The Prange Algorithm. Let us first consider a $[n,k]$-code $\mathcal{C}$ with parity-check matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$ and a syndrome $\mathbf{s} \in \mathbb{F}_2^{n-k}$. The matrix $\mathbf{H}$ is a full-rank, therefore we can choose uniformly at random a set $I \subseteq \{1,\cdots,n\}$ of size $n-k$, usually called an information set, such that, with a high probability, $\mathbf{H}$ restricted to these positions is an invertible matrix. In other words we have $\mathbf{H}_{\pi_I} = [\mathbf{A}|\mathbf{B}]$ where $\mathbf{A} \in \mathbb{F}_2^{(n-k)\times(n-k)}$ is non-singular. We look now for $\mathbf{e}$ of the form $\mathbf{e}_{\pi_I} = (\mathbf{e}'|\mathbf{0}_k)$. We should therefore have $\mathbf{s}^T = \mathbf{H}\mathbf{e}^T = \mathbf{A}\mathbf{e}'^T$. Then thanks to Gaussian elimination, which is done in polynomial time, we compute $\mathbf{e}'^T = \mathbf{A}^{-1}\mathbf{s}^T$. In this way, if the weight of $\mathbf{e} = (\mathbf{e}', \mathbf{0}_k)_{\pi_I^{-1}}$ is w, we just found a solution, otherwise we pick an other set I of $n-k$ positions. Thus, the hard part of this algorithm consists of finding the good set of positions. It can be shown that the probability to find a fixed error of weight w during an iteration is given by $p_{prange} \stackrel{\triangle}{=} \frac{\binom{n-k}{w}}{\binom{n}{w}}$ (it relies among other things on a counting argument over information sets). As it is explained above there is an exponential number $M_{k,n,w}$ (see Proposition 4) errors $\mathbf{e}$ of weight w such that $\mathbf{H}\mathbf{e}^T = \mathbf{s}^T$. In this way, under the assumption (which is a classical one in the study of ISDs) that solutions to SD behave independently of the set I we pick, the average probability (on matrices $\mathbf{H}$) to not find any solution during an iteration is $(1-p_{prange})^{M_{k,n,w}}$ which implies a probability of succeed during one iteration:

$$P_{prange} \stackrel{\triangle}{=} 1 - (1-p_{prange})^{M_{k,n,w}} = \Theta\big(\min\left(1, M_{n,k,w}\cdot p_{prange}\right)\big)$$

where $M_{n,k,w} \cdot p_{prange} = \frac{\binom{n-k}{w}}{2^{n-k}}$. Thus, Prange's algorithm will make on average $\tilde{O}\left(1/P_{prange}\right)$ samples which gives its complexity as the Gaussian elimination is polynomial and it is easily verified that for all w such that $d_{\mathrm{GV}}(n,k) \leq w < (n-k)/2$, $1/P_{prange}$ is exponential in the code length.

Quantum quadratic speedup of the Prange algorithm. There is a direct quantum quadratic speedup which consists to apply Grover's algorithm to find the right information set. It leads to a quantum complexity of $\tilde{O}\left(1/\sqrt{P_{prange}}\right)$.

Generalized information set decoding. The Prange algorithm has been improved in [Ste88, Dum91] by relaxing a little bit the constraint on the set of columns we pick: it allows to have a little bit more than 0 errors in the complementary of the information set I. To perform this task, the algorithm introduces two new parameters p, l and looks for an error of the form $(\mathbf{e}'|\mathbf{e}'')$ where the right side has size $k+l$, $|\mathbf{e}'| = w-p$, $|\mathbf{e}''| = p$ with $\mathbf{e}'$ uniquely determined by $\mathbf{e}''$. More precisely, the improved algorithm first picks a set $I \subseteq \{1, \cdots, n\}$ of size $n-k-l$, then performs a Gaussian elimination on lines of $\mathbf{H}_{\pi_I}$ which gives a non-singular matrix $\mathbf{U}$, as well as matrices $\mathbf{H}'_I \in \mathbb{F}_2^{(n-k-l)\times(k+l)}$ and $\mathbf{H}''_I \in \mathbb{F}_2^{l\times(k+l)}$ such that

$$\mathbf{U}\mathbf{H}_{\pi_I} = \begin{pmatrix} \mathbf{Id}_{n-k-l} & \mathbf{H}'_I \\ \mathbf{0} & \mathbf{H}''_I \end{pmatrix} \tag{1}$$

and

$$\mathbf{U}\mathbf{s}^T = (\mathbf{s}'_I|\mathbf{s}''_I)^T \text{ where } \mathbf{s}'_I \in \mathbb{F}_2^{n-k-l} \text{ and } \mathbf{s}''_I \in \mathbb{F}_2^l. \tag{2}$$

Then if $\mathbf{e}$ is a vector such that $\mathbf{e}_{\pi_I} = (\mathbf{e}'|\mathbf{e}'')$ we have:

$$\begin{aligned} \mathbf{H}\mathbf{e}^T = \mathbf{s}^T &\iff \mathbf{U}\mathbf{H}\mathbf{e}^T = \mathbf{U}\mathbf{s}^T \\ &\iff \begin{pmatrix} \mathbf{Id}_{n-k-l} & \mathbf{H}' \\ \mathbf{0} & \mathbf{H}'' \end{pmatrix} \mathbf{e}_{\pi_I}^T = \begin{pmatrix} \mathbf{s}'^T \\ \mathbf{s}''^T \end{pmatrix} \\ &\iff \begin{pmatrix} \mathbf{e}'^T + \mathbf{H}'\mathbf{e}''^T \\ \mathbf{H}''\mathbf{e}''^T \end{pmatrix} = \begin{pmatrix} \mathbf{s}'^T \\ \mathbf{s}''^T \end{pmatrix} \\ &\iff \mathbf{e}'^T = \mathbf{H}'\mathbf{e}''^T + \mathbf{s}'^T \text{ and } \mathbf{H}''\mathbf{e}''^T = \mathbf{s}''^T \end{aligned}$$

In this way, we compute all errors $\mathbf{e}''$ of weight p such that $\mathbf{H}''\mathbf{e}'^T = \mathbf{s}''^T$, for all vectors we get, we consider $\mathbf{e}_s \triangleq (\mathbf{e}''\mathbf{H}'^T + \mathbf{s}'|\mathbf{e}'')_{\pi_I^{-1}}$ and if one of them has a Hamming weight of w then it is a solution, otherwise we pick another set of size $n-k-l$. Let us introduce now, for each subset I we picked and syndrome $\mathbf{s}$ we look to decode, the set:

$$\mathcal{S}_I = \{\mathbf{e}'' \in \mathbb{F}_2^{k+l} \text{ of Hamming weight } p \;:\; \mathbf{H}''_I\mathbf{e}''^T = \mathbf{s}''^T_I\} \tag{3}$$

$$f_I : \mathbf{e}'' \in \mathbb{F}_2^{k+l} \mapsto \mathbf{e}''\mathbf{H}''^T_I \in \mathbb{F}_2^l \tag{4}$$

$$z_I^{\mathbf{s}} : \mathbf{e}'' \in \mathbb{F}_2^{k+l} \mapsto (\mathbf{e}''\mathbf{H}'^T_I + \mathbf{s}'_I|\mathbf{e}'')_{\pi^{-1}} \in \mathbb{F}_2^n \tag{5}$$

Thanks to equations (1),(2),(3),(4),(5) we are able to formalize generalizes ISDs in Algorithm 1.

Algorithm 1 (generalized) ISD

- **input:** $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}, \mathbf{s} \in \mathbb{F}_2^{(n-k)}, l, p, w$ integers
- **loop**
  - pick a set $I \subseteq \{1, \cdots, n\}$ of size $n-k-l$
  - compute $\mathbf{H}'_I, \mathbf{H}''_I, \mathbf{s}'_I, \mathbf{s}''_I$
  - compute $\mathcal{S}_I$
  - **for all** $\mathbf{e}'' \in \mathcal{S}_I$ **do**
    - $\mathbf{e} \leftarrow h_I(\mathbf{e}'')$
    - **if** $|\mathbf{e}| = w$ **then output** $\mathbf{e}$

Remark 4. From each information set I we can build matrices $\mathbf{H}'_I$, $\mathbf{H}''_I$, $\mathbf{s}_I$ and $\mathbf{s}''_I$ in polynomial time thanks to Gaussian elimination.

This new algorithm leads to a probability $p_{p,l} \triangleq \frac{\binom{k+l}{p}\binom{n-k-l}{w-p}}{\binom{n}{w}}$ ($\geq p_{prange}$ for a set of parameters p, l) of finding a fixed solution.

Remark 5. We stress that to have this probability the algorithm has to consider all errors $\mathbf{e}''$ of weight p such that $\mathbf{H}''\mathbf{e}''^T = \mathbf{s}''^T$.

In a same fashion as before this algorithm will succeed with probability:

$$P_{p,l} \stackrel{\triangle}{=} 1 - (1 - p_{p,l})^{M_{k,n,w}} = \theta\big(\min(1, M_{n,k,w} \cdot p_{p,l})\big)$$

and if we denote by T_{class} the time complexity to compute S_I, which is exponential as the size of S_I is exponential, Algorithm 1 has a complexity given by:

$$\tilde{O}\left(\frac{T_{class}}{P_{p,l}}\right)$$

Many classical algorithms have been proposed to solve Instruction 5 (see [Ste88, Dum91, MMT11, BJMM12, MO15]). They all rely on splitting the matrices even more and finding elements S_I via multi-collision algorithms. In the case of DOOM$_\infty$, similar ideas are applied. We generate several syndromes $\mathbf{s}_1, \ldots, \mathbf{s}_q$. When performing the generalized ISD algorithm, we now have one set S_I for each syndrome $\mathbf{s}_q$. The multi-collision algorithms used in the ISD can take advantage of this in order to find all elements of all the S_I (for different syndromes) in a reduced amortized cost. In this case, as we consider more good events, we obtain

$$P_{p,l} = 1 - (1 - p_{p,l})^{M_{k,n,w}} = \theta\big(\min(1, q \cdot M_{n,k,w} \cdot p_{p,l})\big)$$

Of course, in this case, the computing T_{class} changes and new optimizations have to be done. We will not go into the details of these algorithms and optimizations (see [Sen11] for more details).

The best asymptotic exponent among all those decoding techniques are [MO15, BJMM12] for SD. However, algorithm [MO15] is penalized by a big polynomial overhead which makes it more expensive that [BJMM12]. It is why in the following table we will consider asymptotic exponents given by [BJMM12]. We give in Table 1 classical exponents in base 2 of the Prange algorithm (which was the first algorithm proposed to solve syndrome decoding problem), [BJMM12] and the state-of-the-art to solve DOOM$_\infty$ (see [Sen11]). We present the running times for $k = n/2$ and for two error weights w: namely $w \approx 0.11n$ which corresponds to the Gilbert-Varshamov weight and is the weight around which those problems are the hardest; and $w \approx 0.191n$ which corresponds to the weight used in the SURF signature scheme.

	Classical asymptotic exponent in base 2 (divided by n)		
w/n	SD (Prange)	SD ([BJMM12])	DOOM$_\infty$ [Sen11]
0.11	0.1199	0.1000	0.0872
0.191	0.02029	0.01687	0.01654

Table 1: Asymptotic exponent for classically solving SD and DOOM$_\infty$ for $k/n = 0.5$

The above table contains classical asymptotic exponent in base 2 (divided by n). This means for example that the Prange algorithm for SD with $w = 0.11n$ runs in time $2^{0.1199n}$.

In the quantum setting, things become trickier. While Instruction 3 can be Groverized, it seems hard to get a full quadratic speedup for Instruction 5, because multi-collision problems have a less than quadratic speedup in the quantum setting. If T_{quant} is the quantum running time of Instruction 5 then the total running time becomes $\tilde{O}\left(\frac{T_{quant}}{\sqrt{P_{p,l}}}\right)$. Moreover, any improvement we do in Instruction 5 seems to augment $P_{p,l}$ and therefore reduce the Grover advantage we have from Instruction 3. There seems to be very little place for improvement. In [KT17], authors still managed to find a quantum improvement over the simple quantum Prange algorithm using quantum random walks, even though the advantage is small.

6.2 Quantum Algorithm for solving DOOM$_\infty$

We will focus on Instruction 5 and find the best tradeoffs for our quantum algorithm for DOOM$_\infty$. Similarly as in classical algorithms for SD, we will reduce our problem to a k-sum problem (actually a 4-sum problem). Then by considering known results on quantum walks developed in [KT17], we will be able to give a running time for our quantum algorithm. Let us first introduce the following classical problem.

Problem 4 (Generalized k-sum Problem).
Let $\mathcal{G}$ be an Abelian group, $\mathcal{E}$ be an arbitrary set, k subsets $\mathcal{V}_1, \cdots, \mathcal{V}_k$ of $\mathcal{E}$, $k+1$ arbitrary maps:

$$\forall i \in [\![1,k]\!],\ f_i : \mathcal{E} \to \mathcal{G} \quad ; \quad g : \mathcal{E}^k \to \{0,1\}$$

and an arbitrary $S \in \mathcal{G}$. A solution is a tuple $(v_1, \cdots, v_k) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_k$ such that:

- $f_1(v_1) + \cdots + f_k(v_k) = S$ (subset-sum condition).
- $g(v_1, \cdots, v_k) = 1$.

We now show this reduction. Let $\mathbf{H}, \mathcal{H}$ be an instance of DOOM$_\infty$ and $\mathcal{H}_l$ will denote the projection of $\mathcal{H}$'s outputs onto their last l coordinates. We first pick an information set $I \subseteq \{1, \cdots, n\}$ of size $n-k-l$, then we build matrices $\mathbf{H}'_I$ and $\mathbf{H}''_I$ as in (1).

Associated 4-sum problem. We introduce the following sets and functions (see Equations (3),(4) and (5)):

$$\mathcal{G} = \mathbb{F}_2^{l/2} \times \mathbb{F}_2^{l/2} \quad ; \quad \mathcal{E} = \mathbb{F}_2^{k+l}; S = 0$$
$$\forall i \in [\![1,3]\!],\ f_i : \mathbf{e}'' \in \mathbb{F}_2^{k+l} \mapsto \mathbf{H}''_I \mathbf{e}''^T; f_4 = \mathcal{H}_l$$

with

$$\begin{aligned}
\mathcal{V}_1 &\stackrel{\triangle}{=} \{(\mathbf{e}_1, \mathbf{0}_{2(k+l)/3}) \in \mathbb{F}_2^{k+l} \ :\ \mathbf{e}_1 \in \mathbb{F}_2^{(k+l)/3},\ |\mathbf{e}_1| = p/3\} \\
\mathcal{V}_2 &\stackrel{\triangle}{=} \{(\mathbf{0}_{(k+l)/3}, \mathbf{e}_2, \mathbf{0}_{(k+l)/3}) \in \mathbb{F}_2^{k+l} \ :\ \mathbf{e}_2 \in \mathbb{F}_2^{(k+l)/3},\ |\mathbf{e}_2| = p/3\} \\
\mathcal{V}_3 &\stackrel{\triangle}{=} \{(\mathbf{0}_{2(k+l)/3}, \mathbf{e}_3) \in \mathbb{F}_2^{k+l} \ :\ \mathbf{e}_3 \in \mathbb{F}_2^{(k+l)/3},\ |\mathbf{e}_3| = p/3\} \\
\mathcal{V}_4 &\text{ be an arbitrary set of size } \binom{(k+l)/3}{p/3}
\end{aligned}$$

and

$$g(v_1, v_2, v_3, v_4) = 1 \iff |z_I^{\mathcal{H}(v_4)}(v_1 + v_2 + v_3)| = w$$

Proposition 5. *If (v_1, v_2, v_3, v_4) is a solution of the above problem then $(v_1 + v_2 + v_3, v_4)$ is a solution of the* DOOM$_\infty$ *problem on inputs* $(\mathbf{H}, \mathcal{H})$.

Proof. Let (v_1, v_2, v_3, v_4) a solution of the associated 4-sum problem. We have

$$\begin{aligned}
f_1(v_1) + f_2(v_2) + f_3(v_3) + f_4(v_4) = 0 &\iff \mathbf{H}''_I (v_1 + v_2 + v_3)^T = \mathcal{H}_l(v_4) \\
&\iff v_1 + v_2 + v_3 \in \mathcal{S}_I \ \text{ for the syndrome } \mathcal{H}(v_4)
\end{aligned}$$

This means that $|v_1+v_2+v_3| = p$. We also $g(v_1, v_2, v_3, v_4) = 1$ which implies $|z_I^{\mathcal{H}(v_4)}(v_1+v_2+v_3)| = w$. By definition of z_I, this shows that

$$\mathbf{H}(z_I^{\mathcal{H}(v_4)}(v_1 + v_2 + v_3)) = \mathcal{H}(v_4)$$

which concludes the proof.

All the above discussion was for a fixed information set I so our goal is to use a quantum algorithm for the 4-sum problem to solve instruction 5. Fortunately, there already exists a quantum study of this problem using quantum walks [KT17, Proposition 2].

Proposition 6. *Consider the generalized* 4*-sum problem defined in Problem 4 with sets* $\mathcal{V}_i$ *of the same size* V*. Assume that* $\mathcal{G}$ *can be decomposed as* $\mathcal{G} = \mathcal{G}_0 \times \mathcal{G}_1$ *with* $|\mathcal{G}_0|, |\mathcal{G}_1| = \Theta(V^{4/5})$*. There is a quantum algorithm (using a random walk) for solving the* 4*-sum problem in running time* $\tilde{O}\left(V^{6/5}\right)$*.*

We now put everything together and present the running time of this quantum algorithm for DOOM_∞.

Theorem 2. *We can solve* DOOM_∞ *for parameters* n, k *and* $w \geq d_{GV}(n,k)$ *in time:*

$$\tilde{O}\left(\min_{0 \leq l \leq n-k}\left(\frac{T_{quant}(p,l)}{\sqrt{P_{p,l}}}\right)\right)$$

where:

$$P_{p,l} = \Theta\left(\min\left(1, \frac{\binom{k+l}{p}\binom{n-k-l}{w-p}\binom{(k+l)/3}{p/3}}{2^{n-k}}\right)\right)$$

and

$$T_{quant}(p,l) = \binom{(k+l)/3}{p/3}^{6/5}$$

with p *chosen such that:*

$$2^{l/2} = \Theta\binom{(k+l)/3}{p/3}^{4/5}$$

The value of T_{quant} is obtained from Proposition 6. The other parameters are obtained from the classical analysis in the case where we consider $\binom{(k+l)/3}{p/3}$ syndromes. We present below quantum asymptotic exponents for SD and for DOOM_∞. Again, we consider $k = n/2$ and for error weights $w \approx 0.11n$ and $w \approx 0.191n$ which corresponds to the weight used in the SURF signature scheme.

	Quantum asymptotic exponent in base 2 (divided by n)		
w/n	SD (Prange)	SD [KT17]	DOOM_∞(this work)
0.11	0.059958	0.058434	0.056683
0.191	0.010139	0.009218	0.009159

Table 2: Asymptotic exponent for quantumly solving SD and DOOM_∞ for $k/n = 0.5$

7 Quantum security of the SURF signature scheme

We apply in this section our results to the SURF signature scheme presented in [DST17]. Let us recall the condition upon which stands our security reduction in the QROM:

Condition 1

1. $\frac{8\pi}{\sqrt{3}} q_{\text{hash}}^{\frac{3}{2}} \sqrt{\mathbb{E}_{\mathbf{H}_{\text{pub}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_{n-k})\right)}) \in negl(\lambda)$
2. $q_{\text{sign}} \rho\left(\mathcal{U}_w, \mathcal{D}_w\right) \in negl(\lambda)$
3. $\rho_{Qc}\left(\mathcal{D}_{\text{pub}}, \mathcal{D}_{\text{rand}}\right)(t) = o(\frac{t}{2^\lambda})$.

where λ is the security parameters. Authors of [DST17] proposed to use the family of $(U|U+V)$-codes as the secret key, namely:

Definition 6 **($(U, U+V)$-Codes).** *Let U, V be linear binary codes of length $n/2$ and dimension k_U, k_V. We define the subset of $\mathbb{F}_2^n$:*

$$(U, U+V) \stackrel{\triangle}{=} \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \text{ such that } \mathbf{u} \in U \text{ and } \mathbf{v} \in V\}$$

which is a linear code of length n and dimension $k = k_U + k_V$.

We choose parameters of public keys as:

$$n = 13976 \ ; \ k = 6988 \ ; \ k_U = 4320 \ ; k_V = 2668 \ ; \ w = 2668.$$

The value n was chosen to get 128 bits of security for the DOOM_∞ problem and the other parameters were already constrained (given n) from the specifications of SURF. We can now check the 3 conditions.

1. Using the results of [DST17, Proposition 4], we get for our parameters $\mathbb{E}_{\mathbf{H}_{\text{pub}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_{n-k})\right) = 2^{-0.06n}$ which gives if we choose a conservative $q_{\text{hash}} = 2^{128}$:
$$q_{\text{hash}}^{\frac{3}{2}} \sqrt{\mathbb{E}_{\mathbf{H}_{\text{pub}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pub}}}, \mathcal{U}_{n-k})\right)} = \frac{1}{2^{235}}.$$
2. SURF performs a rejection sampling (see [DST17, Section 5]) algorithm that achieves $\rho(\mathcal{U}_w, \mathcal{D}_w) = 0$.
3. While the authors of [DST17] do not formally study quantum distinguishers for their code family, the best known classical algorithms not only also use multi-collision techniques and are hard even to Groverize. Also, for our parameters the classical advantage (see [DST17, Section 7]) is of the order of 2^{-500}. Any quantum distinguisher for those codes would have to find radically new quantum algorithmic techniques way beyond the state of the art.

Finally, with parameters and using the analysis of [DST17], we obtain the following parameters (we also include the parameters of the other quantum-safe signature schemes)

Table 3: Security parameters for signature schemes with quantum security claims

Scheme	Quantum security (in bits)	Public key size (in kBytes)	Private key size (in kBytes)	Signature size (in kBytes)
SPHINCS	128	1	1	41
GPV-poly	59	55	26	32
GPV	59	27840	12064	30
TESLA-2	128	21799	7700	4
SURF	128	5960	3170	1.7

Moreover, for this choice of parameters the SURF signature scheme achieves a classical security of 231 bits.

8 Conclusion

In this paper, we presented a method to perform tight security reductions for FDH-like signature schemes using code-based computational assumptions, more precisely on the DOOM_∞ problem. We also analyzed the best known quantum algorithm for this problem. Finally, we applied our security reduction to the SURF signature scheme, presenting parameters for 128 bits of concrete quantum security and think this scheme will play an important role in the future standardization attempts from NIST. We finally list several open questions and perspectives that come out of this work:

– Our security reduction can be applied to only one signature scheme now. Are there other constructions that could benefit from this reduction? The SURF signature scheme uses a code family which has very little structure. This strengthens the security but increases the key sizes. Can we use another code family that would stay secure with smaller key sizes?
– More generally, our techniques show that it is much better in the code-based setting to consider a computational assumption which starts from many instances of a problem and where we need to solve one of them. This One Out of Many approach appears implicitly when performing instance injection but doesn't appear explicitly in other signature schemes. For example, it would be very interesting to consider a One Out of Many equivalent for lattice schemes, and could be a way to reduce losses resulting from the quantum security reduction.
– Finally, since the security rely on the quantum hardness of the DOOM_∞ problem, it is important to continue to study it - similarly as other quantum-safe computational assumptions - in order to increase our trust in quantum secure schemes.

References

[ABB+17] Erdem Alkim, Nina Bindel, Johannes Buchmann, Özgür Dagdelen, Edward Eaton, Gus Gutoski, Juliane Krämer, and Filip Pawlega. Revisiting TESLA in the quantum random oracle model. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography : 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, pages 143–162, Cham, 2017. Springer International Publishing.

[AGJO+15] Srinivasan Arunachalam, Vlad Gheorghiu, Tomas Jochym-O'Connor, Michele Mosca, and Priyaa Varshinee Srinivasan. On the robustness of bucket brigade quantum ram. *New Journal of Physics*, 17(12), 2015.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Proceedings of the 17th International Conference on The Theory and Application of Cryptology and Information Security*, ASIACRYPT'11, pages 41–69, Berlin, Heidelberg, 2011. Springer-Verlag.

[BGGM15] Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improving nfs for the discrete logarithm problem in non-prime finite fields. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 129–155, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[BHH+15] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. Sphincs: Practical stateless hash-based signatures. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 368–397, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[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.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *Journal of Cryptology*, 17(4):297–319, 2004.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures-how to sign with rsa and rabin. In Ueli Maurer, editor, *Advances in Cryptology — EUROCRYPT '96: International Conference on the Theory and Application of Cryptographic Techniques Saragossa, Spain, May 12–16, 1996 Proceedings*, pages 399–416, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[CFS01] Nicolas T. Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings*, pages 157–174, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, July 2004.

[CTS16] Rodolfo Canto-Torres and Nicolas Sendrier. Analysis of information set decoding for a sub-linear error weight. In *PQCRYPTO*, LNCS, pages 144–161, Fukuoka, Japan, 2016.

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 40–56, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security: Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005. Proceedings*, pages 164–175, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[DST17] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. SURF: A new code-based signature scheme. *CoRR*, abs/1706.08065, 2017.

[DT17] Thomas Debris-Alazard and Jean-Pierre Tillich. Statistical decoding. In *ISIT*, Aachen, Germany, 2017.

[Dum91] Ilya Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, pages 50–52, Moscow, 1991.

[EBB14] Rachid El Bansarkhani and Johannes Buchmann. Improvement and efficient implementation of a lattice-based signature scheme. In Tanja Lange, Kristin Lauter, and Petr Lisoněk, editors, *Selected Areas in Cryptography – SAC 2013: 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, pages 48–67, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[Goo16] Google. Experimenting with post-quantum cryptography. Matt Braithwaite, Google Security Blog: https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html, 2016.

[JJ02] Thomas Johansson and Fredrik Jönsson. On the complexity of some cryptographic problems based on the general decoding problem. *IEEE Transactions on Information Theory*, 48(10):2669–2678, 2002.

[KT17] Ghazal Kachigar and Jean-Pierre Tillich. Quantum information set decoding algorithms. preprint, arXiv:1703.00263 [cs.CR], February 2017.

[LN09] Gaëtan Leurent and Phong Q. Nguyen. How risky is the random-oracle model? In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 445–464, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[Mc 78] R.J Mc Eliece. A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116, 1978.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $O(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2011.

[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.

[Nie86] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.

[NIS16] NIST. Post-quantum crypto project. http://csrc.nist.gov/groups/ST/post-quantum-crypto/, February 2016.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[Sen11] Nicolas Sendrier. Decoding one out of many. In *PQCRYPTO*, volume 7071 of *Lecture Notes in Computer Science*, pages 51–67, 2011.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332, 2004.

[Ste88] Jacques Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, pages 679–687, Washington, DC, USA, 2012. IEEE Computer Society.