Efficient error-correcting codes for the HQC post-quantum cryptosystem

Carlos Aguilar-Melchor, Nicolas Aragon, Jean-Christophe Deneuville, Philippe Gaborit, Jérôme Lacan, Gilles Zémor

Designs, Codes and Cryptography (2024) 92:4511–4530
https://doi.org/10.1007/s10623-024-01507-6

Efficient error-correcting codes for the HQC post-quantum cryptosystem

Carlos Aguilar-Melchor[1] · Nicolas Aragon[2] · Jean-Christophe Deneuville[3] · Philippe Gaborit[2] · Jérôme Lacan[4] · Gilles Zémor[5]

Received: 21 June 2023 / Revised: 20 August 2024 / Accepted: 19 September 2024 /
Published online: 9 October 2024

Abstract
The HQC post-quantum cryptosystem enables two parties to share noisy versions of a common secret binary string, and an error-correcting code is required to deal with the mismatch between both versions. This code is required to deal with binary symmetric channels with as large a transition parameter as possible, while guaranteeing, for cryptographic reasons, a decoding error probability of provably not more than 2^{-128}. This requirement is non-standard for digital communications, and modern coding techniques are not amenable to this setting. This paper explains how this issue is addressed in the last version of HQC: precisely, we introduce a coding scheme that consists of concatenating a Reed–Solomon code with the tensor product of a Reed–Muller code and a repetition code. We analyze its behavior in detail and show that it significantly improves upon the previous proposition for HQC, which consisted of tensoring a BCH and a repetition code. As additional results, we also provide a better approximation of the weight distribution for HQC error vectors, and we remark that

Communicated by O. Ahmadi.

✉ Jean-Christophe Deneuville
jean-christophe.deneuville@enac.fr

Carlos Aguilar-Melchor
carlos@sandboxquantum.com

Nicolas Aragon
nicolas.aragon@unilim.fr

Philippe Gaborit
gaborit@unilim.fr

Jérôme Lacan
jerome.lacan@isae-supaero.fr

Gilles Zémor
zemor@math.u-bordeaux.fr

1 SandboxAQ, Palo Alto, USA

2 University of Limoges, Limoges, France

3 Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, Toulouse, France

4 ISAE-Supaero, University of Toulouse, Toulouse, France

5 IMB, University of Bordeaux, Bordeaux, France

the size of the exchanged secret in HQC can be reduced to match the protocol security which also significantly improves performance.

Keywords Post-quantum cryptography · Code-based cryptography · Public-key encryption · HQC · NIST

Mathematics Subject Classification 94A60 · 11T71 · 14G50

1 Introduction

Post-quantum cryptography, which aims at providing schemes resistant against quantum computers, has been a growing topic of study for the last decade and has been given an enormous boost by the NIST standardization competitive process [13]. One of the approaches to devising post-quantum cryptosystems is code-based cryptography. Code-based cryptography is arguably the oldest post-quantum approach, since it includes the McEliece cryptosystem [11], originally proposed in 1978 (though its post-quantum appeal was of course not apparent at the time) and still considered by NIST for standardization at the time of writing [13].

The McEliece scheme provides a general public-key framework that can in principle be instantiated with any family of error-correcting codes having an efficient decoding algorithm: indeed, the scheme's simple and attractive main idea is to encrypt messages as noisy codewords and hide the decoding algorithm from anyone but the legitimate receiver. However, the security of the cryptosystem is highly dependent on the choice of the family of codes being used. Even though the original instantiation, based on binary Goppa codes, has time-tested security, many others (for example using reducible rank codes [5] or Reed–Solomon codes [12]) have been broken by recovering the hidden structure of the code from the public key. The very nature of this framework makes it difficult to reduce the security of the scheme to the difficulty of a well-established problem such as decoding random linear codes.

In [2], the authors propose a framework that can be instantiated in different metrics to derive code-based cryptosystems with a built-in security reduction to a decision version of the decoding problem for random quasi-cyclic codes. The instantiation of this framework in the Hamming metric is HQC (Hamming Quasi Cyclic), which was submitted to the NIST Post-Quantum standardization process and is one of the three remaining candidates in the fourth round of the competition. Thanks to the quasi-cyclic structure, the scheme features compact key sizes (about 3kB for a security of 128 bits) as well as fast key generation, encryption, and decryption operations.

At the heart of the HQC scheme is a key-exchange mechanism which enables the sender and the receiver to share a random binary string: this binary string could then in principle be one-time padded to a plaintext to create a ciphertext. However, the communicating parties do not have exactly the same random binary string: the receiver has a noisy version of the sender's. Therefore, a structured public error-correcting code $\mathcal{C}$ is needed to encode the plaintext and remove the noise inherent to the decryption process. To ensure the underlying problems are hard enough, the error weight is counted in thousands of bits in the HQC scheme, but the message to be transported is generally small (between 128 and 256 bits). As the most communication-efficient solution is to be obtained, this leads to a setting in which low transmission-rate codes in a high error-rate channel are needed. Moreover, it is required that the resulting cryptographic scheme successfully decrypts with overwhelming probability. This additional constraint is inherited from two considerations: first, as exhibited

by Guo et al. [6], decryption failures can yield secret key recovery attacks and second, it is a pre-requisite to the generic HHK transform of [7] for resisting against active adversaries (this transform is described in more details in Sect. 3, and the resulting KEM is pictured in Appendix A). In the rest of this paper, this notion is referred to as the Decryption Failure Rate (DFR), which is defined as the decoding error probability for the code $\mathcal{C}$. Choosing the most appropriate code is therefore a purely coding-theoretic problem, since its role is solely to correct errors and it has no cryptographic requirement, apart from implementation concerns. However, figuring out what is the best suited error-correcting code is an unusual challenge from a coding theory point of view. In particular, the required decoding error probability rules out decoder simulations, and it also requires large minimum distances together with the absence of small-weight or average-weight pseudo-codewords, which leaves out of the picture modern decoding techniques for LDPC codes or polar codes, for example.

In [2], the authors proposed tensor products of BCH and repetition codes: this was a natural suggestion, because intuition suggests that repetition is hard to beat in a very high noise setting, and its simplicity allows for fast encoding and precise DFR analysis. The analysis consists of two steps: first, the weight distribution of the error vector is studied, and then the DFR of the chosen codes for given weights is analyzed.

The contribution of the present paper is the proposal and analysis of an alternative coding scheme aimed at the high-noise low-rate scenario of HQC, and which is now integrated in the current upgraded version of the NIST submission. It consists of the concatenation of a Reed–Solomon code with an inner code that is the tensor product of a Reed–Muller code and a small repetition code. We will call the inner code a duplicated Reed–Muller code for short. We provide a fine-grained analysis of the DFR of duplicated Reed–Muller codes; we prove that the DFR of these concatenated codes is cryptographically small for codes that are shorter and faster to decode than the original BCH/repetition construction. This leads to shorter keys and ciphertexts and faster encryption/decryption operations in HQC. Concatenated coding schemes have been extensively studied and applied, and first-order Reed–Muller codes are one of the oldest, simplest, and most studied codes: however, using duplicated Reed–Muller codes in a concatenated scheme is non-standard, and we do not know of any previous experiments with these codes. Not every code has the same decoding complexity, and it is also important to consider their effective performance. In our case, replacing BCH and repetition codes by Reed–Solomon and Reed–Muller codes turns out to be more efficient, as noted in the round 2 submission of HQC to the NIST standardization process.

Beside these contributions, we also derive formal lower bounds for the HQC framework using a sphere packing argument: the lower bounds show that there is not very much room for significant improvements to the error-correcting scheme. These bounds also provide an implicit metric between different codes for this construction (the ratio between the code size and the lower bound), which we use to measure the impact of our contributions over HQC. We also provide a better analysis of the distribution of the weight of the error vector in HQC, which leads to a better DFR analysis regardless of which public code is used to decode it. Finally, we note that it is possible to reduce the number of bits of the exchanged message without lowering security, which provides a significant improvement on the code size when using the encryption scheme for key exchange.

Paper organization. In Sect. 2 we introduce notation and describe the HQC scheme. Section 3 provides lower bounds on the length of the public code for HQC in order to reach a negligible DFR. We then provide an improvement of the previous analysis of the distribution of the error weight in HQC in Sect. 4 and then apply this analysis to the original BCH/repetition construction and to the new Reed–Muller/Reed–Solomon one in Sect. 5.

2 Preliminaries

In this section, we introduce the basic notation and the description of the HQC scheme. For more details on the protocol and the security proof, we refer the reader to [2].

Throughout this document, $\mathbb{Z}$ denotes the ring of integers and $\mathbb{F}_2$ the binary field. Additionally, we denote by $\omega(\cdot)$ the Hamming weight of a vector *i.e.* the number of non-zero coordinates, and by $\mathcal{S}_w^n(\mathbb{F}_2)$ the set of words in $\mathbb{F}_2^n$ of weight w. Formally:

$$\mathcal{S}_w^n(\mathbb{F}_2) = \left\{\mathbf{v} \in \mathbb{F}_2^n, \text{ such that } \omega(\mathbf{v}) = w\right\}.$$

Elements of $\mathbb{F}_2^n$ can be interchangeably considered as row vectors or polynomials in the ring $\mathbb{F}_2[X]/(X^n-1)$. Vectors/Polynomials (resp. matrices) will be represented by lower-case (resp. upper-case) bold letters. For a vector $\mathbf{v}$, v_k denotes its k-th coordinate. For the sake of conciseness, we will say that a prime integer n is primitive if 2 is a primitive n-th root of unity or equivalently, if the polynomial $(X^n-1)/(X-1)$ is irreducible in $\mathbb{F}_2[X]$.

For $\mathbf{u}, \mathbf{v} \in \mathbb{F}_2^n$, we define their product similarly as in $\mathbb{F}_2[X]/(X^n-1)$, *i.e.* $\mathbf{uv} = \mathbf{w} \in \mathbb{F}_2^n$ with

$$w_k = \sum_{i+j \equiv k \mod n} u_i v_j, \text{ for } k \in \{0, 1, \ldots, n-1\}. \tag{1}$$

HQC takes great advantage of matrices with a cyclic structure. Following [2], $\mathbf{rot}(\mathbf{v})$ for $\mathbf{v} \in \mathbb{F}_2[X]/(X^n-1)$ denotes the circulant matrix whose i-th column is the vector corresponding to $\mathbf{v}X^i$. This is captured by the following definition.

Definition 1 (Circulant Matrix) Let $\mathbf{v} = (v_0, \ldots, v_{n-1}) \in \mathbb{F}_2^n$. The *circulant matrix* induced by $\mathbf{v}$ is defined and denoted as follows:

$$\mathbf{rot}(\mathbf{v}) = \begin{pmatrix} v_0 & v_{n-1} & \ldots & v_1 \\ v_1 & v_0 & \ldots & v_2 \\ \vdots & \vdots & \ddots & \vdots \\ v_{n-1} & v_{n-2} & \ldots & v_0 \end{pmatrix} \in \mathbb{F}_2^{n \times n}. \tag{2}$$

As a consequence, it is easy to see that the product of any two elements $\mathbf{u}, \mathbf{v} \in \mathbb{F}_2[X]/(X^n-1)$ can be expressed as a usual vector–matrix (or matrix–vector) product using the $\mathbf{rot}(\cdot)$ operator as

$$\begin{aligned} \mathbf{u} \cdot \mathbf{v} &= \mathbf{u} \times \mathbf{rot}(\mathbf{v})^\top \\ &= \left(\mathbf{rot}(\mathbf{u}) \times \mathbf{v}^\top\right)^\top \\ &= \mathbf{v} \times \mathbf{rot}(\mathbf{u})^\top \\ &= \mathbf{v} \cdot \mathbf{u}. \end{aligned} \tag{3}$$

We now recall the HQC scheme in Fig. 1. In [2], the code $\mathcal{C}$ used for decoding is a tensor product of BCH and repetition codes. But since this code is public, its structure has no incidence on security as long as its DFR is cryptographically low, and one can choose any code family, influencing only the DFR and the parameter sizes.

We have $\mathbf{v} - \mathbf{uy} = \mathbf{mG} + \mathbf{xr}_2 - \mathbf{r}_1\mathbf{y} + \mathbf{e} = \mathbf{mG} + \mathbf{e}'$ for some $\mathbf{e}'$. Therefore, for the correctness property to hold, the error vector $\mathbf{e}'$ inside $\mathbf{v} - \mathbf{uy}$ must be small enough to successfully decode with overwhelming probability. The next section gives a binary symmetric channel model and establishes bounds for a decoding error event so as to obtain lower bounds on the code length.

- $\mathsf{Setup}(1^\lambda)$: generates and outputs the global parameters $\mathsf{param} = (n, k, \delta, w, w_{\mathbf{r}}, w_{\mathbf{e}})$.
- $\mathsf{KeyGen}(\mathsf{param})$: samples $\mathbf{h} \xleftarrow{\$} \mathcal{R}$, the generator matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ of the public code $\mathcal{C}$, $\mathsf{sk} = (\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{R}^2$ such that $\omega(\mathbf{x}) = \omega(\mathbf{y}) = w$, sets $\mathsf{pk} = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{hy})$, and returns $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m})$: generates $\mathbf{e} \xleftarrow{\$} \mathcal{R}$, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \xleftarrow{\$} \mathcal{R}^2$ such that $\omega(\mathbf{e}) = w_{\mathbf{e}}$ and $\omega(\mathbf{r}_1) = \omega(\mathbf{r}_2) = w_{\mathbf{r}}$, sets $\mathbf{u} = \mathbf{r}_1 + \mathbf{h}\cdot\mathbf{r}_2$ and $\mathbf{v} = \mathbf{mG} + \mathbf{sr}_2 + \mathbf{e}$, returns $\mathbf{c} = (\mathbf{u}, \mathbf{v})$.
- $\mathsf{Decrypt}(\mathsf{sk}, \mathbf{c})$: returns $\mathcal{C}.\mathsf{Decode}(\mathbf{v} - \mathbf{uy})$.

Fig. 1 Description of HQC

3 Lower bounds on the code length to achieve negligible DFR

In this section, we provide lower bounds on the lengths of the public codes that are needed for HQC and compare these lower bounds to the code lengths obtained with the BCH/repetition proposal and with the present work. In order to define the lower bounds, we consider the fixed error weights $w, w_{\mathbf{e}}, w_{\mathbf{r}}$ (defined in Fig. 1) that are needed to achieve respectively $\lambda = 128, \lambda = 192$ and $\lambda = 256$ bits of security in HQC, ensuring in particular that the best-known attacks (see [2, Sects. II.D & VII.A]) will have a complexity at least 2^λ. What is then required is an error-correcting code $\mathcal{C}$ that can deal with a Binary Symmetric Channel (BSC) of a certain parameter $p^\star$ that is a function of $n, w, w_{\mathbf{e}}, w_{\mathbf{r}}$ and that we will compute explicitly in Sect. 4. The assumption that codewords of $\mathcal{C}$ are submitted to a BSC will also be discussed at length in Sect. 4. Below, we derive a lower bound on the code length for any code $\mathcal{C}$ to achieve a DFR of $2^{-\lambda}$ when submitted to the above BSC: this bound enables us to give a measure of the performance of codes $\mathcal{C}$ and associated decoding schemes for HQC.

To this end, we provide two figures: one in which the code is required to transport λ bits and the other one in which the code is always required to transport 256 bits, irrespective of the security level, for comparison. These requirements must be met to use HQC for key exchange, which is the main application of public key encryption schemes. Indeed, a key exchange scheme must be able to transport keys of the same size as its security parameter and in [2], the authors propose to use the HHK transformation [7] to obtain IND-CCA security[1]. This transformation extends the Fujisaki-Okamoto approach, and allows to turn any One-Way Public Key Encryption scheme secure against Chosen Plaintext Attacks (OW-CPA PKE for short) into an IND-CCA Key Encapsulation Mechanism in the Random Oracle Model. This transformation is detailed in Appendix A. In order to apply this transform, the PKE scheme has to satisfy the correctness property, with overwhelming probability in the security parameter. Hence, as stated before, the DFR constraint is a pre-requisite that stems from that transformation, but also from the fact that decryption failures could yield key recovery attacks [6].

Figure 2 presents the lower bounds and compares these bounds with the lengths achieved for the BCH/Repetition approach and with the present work. These results are also presented numerically for our work together with the lower bounds in Table 1. The last column of this table presents the ratio between the bounds and our work which shows that we are relatively close to an optimal solution.

[1] IND-CCA and IND-CCA2 used to refer to different security properties, the difference being that in the latter, the adversary could make additional queries to the decryption oracle *after* having received the challenge

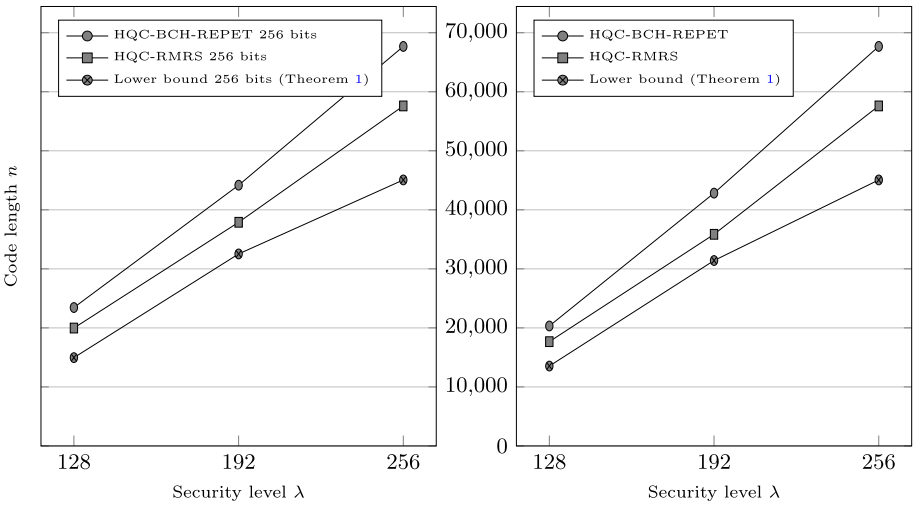

Fig. 2 Values of n needed to achieve a DFR of $2^{-\lambda}$ and transport keys of 256 (left figure) or λ (right figure) bits. Note that for $\lambda = 256$ the results provided by both figures are the same. For lower values of λ, sending λ bits instead of 256 results in a noticeable reduction of n. The ratio n/n_b, where n_b is the lower bound on n provided in the figures, is a performance measure defining the distance to an optimal solution. Replacing HQC-BCH-REPET with our proposal improves significantly this measure. Basically we reduce the gap between what is practically feasible and theoretically possible by a factor close to 2

Table 1 Theoretical code length to obtain a DFR $< 2^{-\lambda}$, where λ is the security parameter

Security parameter	Minimal n	Current n	Minimal/current
128	13,534	17,669	0.766
192	31,411	35,851	0.876
256	45,064	57,601	0.782

Proof of the lower bounds. As mentioned above, the noisy codeword $\mathbf{v} - \mathbf{uy}$ of HQC (see Fig. 1) must be decoded on a memoryless BSC with a Bit Error Rate (BER) $p^\star$ that depends on the weights w, $w_\mathbf{e}$, $w_\mathbf{r}$ and on n, the length[2] of the code. It is relatively straightforward to justify the existence of lower bounds. For a given security parameter λ the noise weights w, $w_\mathbf{e}$ and $w_\mathbf{r}$ are fixed. When n decreases, the BER $p^\star$ increases according to Eq. (6). Below a given value of n the error rate will be too high to be corrected with probability $1 - 2^{-\lambda}$ when transporting λ bits, thus a lower bound exists. Since the values w, $w_\mathbf{e}$, $w_\mathbf{r}$ are determined by the security parameter λ, let us denote by $\mathsf{ber}(n, \lambda)$ the function that calculates the BER $p^\star$ from n and the security parameter λ. We follow a standard packing argument consisting of saying that if the error vector has a sufficiently large weight t, then the spheres of radius t centered on the codewords must intersect on many points of the ambient space, and we have a decoding error event with large probability, see for example [3].

ciphertext. IND-CCA2 being the most commonly required security feature, it is now commonly referred to as IND-CCA, and we adopt this convention in this paper.

[2] Note that, as n must be primitive, for some code parameters we will extend the code by padding it with a few zeros so that the length is a primitive number. In practice, the length of the initial code and n will be very close and for simplicity we will consider them equal.

Theorem 1 *Let $\mathcal{C}[n, k]$ denote a linear code over $\mathbb{F}_2$. Let λ denote the security parameter, and $p = \mathsf{ber}(n, \lambda)$. Let $t_0 = t_0(n, \lambda)$ denote the smallest integer t such that $|\mathcal{S}_t^n(\mathbb{F}_2)| = \binom{n}{t} \geq 2^{n+1-k}$. Also denote the probability of having an error $\mathbf{e}$ of weight greater than or equal to t in a memoryless BSC by $p_{\geq t}(n, p) = \Pr[\omega(\mathbf{e}) \geq t]$:*

$$p_{\geq t}(n, p) = \Pr[\omega(\mathbf{e}) \geq t] = \sum_{i=t}^{n} \binom{n}{i} p^i (1-p)^{n-i}.$$

Finally, define n_λ as the smallest n such that

$$\frac{1}{2} \cdot p_{\geq t_0}(n, \mathsf{ber}(n, \lambda)) < 2^{-\lambda}.$$

Then n_λ is a lower bound on the code length n for the code $\mathcal{C}$ to have negligible DFR $2^{-\lambda}$.

Proof $|\mathcal{S}_t^n(\mathbb{F}_2)| = \binom{n}{t}$ is the number of vectors in $\mathbb{F}_2^n$ on the surface of the sphere of radius t centered in 0, and thus $t_0 = t_0(n, \lambda)$ is the smallest t such that $|\mathcal{C}| \cdot \binom{n}{t} \geq 2 \cdot |\mathbb{F}_2^n|$. If we consider all the spheres of radius $t_0(n, \lambda)$ around the codewords of $\mathcal{C}$, on average each vector in $\mathbb{F}_2^n$ is expected to be on the surface of at least two spheres. Given the code linearity, which implies that a decoding error event depends only on the error vector and not the transmitted codeword, we have that a uniformly random error of weight $t_0(n, \lambda)$ (or above) decoded by maximum likelihood has a probability at least $1/2$ of leading to a decoding error.

The probability of having an error $\mathbf{e}$ of weight greater than or equal to t in a memoryless BSC with error rate p is

$$p_{\geq t}(n, p) = \sum_{i=t}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

and therefore $\mathrm{LB}_{\mathrm{DFR}}(n, \lambda) = (1/2) \cdot p_{\geq t_0}(n, \mathsf{ber}(n, \lambda))$ is a lower bound on the DFR of $\mathcal{C}$. To find the best lower bound on n for a given λ, it is enough to start at $n = 0$ and iterate until the first value n_λ such that $\mathrm{LB}_{\mathrm{DFR}}(n_\lambda, \lambda) < 2^{-\lambda}$ is found. (notice that in practice, for cryptographic reasons, n will have to be primitive: since $(X^n - 1)/(X - 1)$ is irreducible in $\mathbb{F}_2[X]$, it removes all potential structure that could be exploited by attacks such as [9]). □

In the following section, we study the distribution of the resulting error vector when trying to decrypt the ciphertext. This will be helpful to set the parameters so that, with overwhelming probability, the error will have a weight below the error correction capability of the code.

4 Analysis of the error vector distribution for Hamming distance

From the description of the HQC framework (see Fig. 1), decryption corresponds to decoding the received vector: $\mathbf{v} - \mathbf{uy} = \mathbf{mG} + \mathbf{e}'$ for the error vector $\mathbf{e}' = \mathbf{xr}_2 - \mathbf{r}_1\mathbf{y} + \mathbf{e}$. In this section, we provide a more precise analysis of the error distribution approximation compared to [2]. We first compute exactly the probability distribution of each fixed coordinate e'_k of the error vector

$$\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e} = (e'_0, \ldots e'_{n-1}).$$

We obtain that every coordinate e'_k is Bernoulli distributed with parameter $p^\star = P[e'_k = 1]$ given by Proposition 3.

To compute the probabilities of decoding errors, we will then need the probability distribution of the weight of the error vector $\mathbf{e}'$ restricted to given sets of coordinates. We will make the simplifying assumption that the coordinates e'_k of $\mathbf{e}'$ are independent variables, which will let us work with the binomial distribution of parameter $p^\star$ for the weight distributions of $\mathbf{e}'$. This working assumption is justified by remarking that, in the high weight regime relevant to us, since the component vectors $\mathbf{x}, \mathbf{y}, \mathbf{e}$ have fixed weights, the probability that a given coordinate e'_k takes the value 1 conditioned on abnormally many others equaling 1 can realistically only be $\leq p^\star$. We support this modeling of the otherwise intractable weight distribution of $\mathbf{e}'$ by extensive simulations (see Sect. 4.3, and in particular Table 3 and Fig. 3). These back up our assumption that our computations of decoding error probabilities and DFRs can only be upper bounds on their real values.

4.1 Analysis of the distribution of the product of two vectors

The vectors $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2, \mathbf{e}$ have been taken uniformly random and independently chosen among vectors of weight w, $w_\mathbf{r}$ and $w_\mathbf{e}$. We first evaluate the distributions of the products $\mathbf{x} \cdot \mathbf{r}_2$ and $\mathbf{r}_1 \cdot \mathbf{y}$.

Proposition 2 *Let $\mathbf{x} = (x_0, \ldots x_{n-1})$ and $\mathbf{r} = (r_0, \ldots, r_{n-1})$ be independent random vectors chosen uniformly among all binary vectors of weight w and w_r respectively. Then, denoting $\mathbf{z} = \mathbf{x} \cdot \mathbf{r}$, we have that, for every $k \in \{0, \ldots n-1\}$, the k-th coordinate z_k of $\mathbf{z}$ is Bernoulli distributed with parameter $\tilde{p} = P[z_k = 1]$ equal to:*

$$\tilde{p} = \frac{1}{\binom{n}{w}\binom{n}{w_\mathbf{r}}} \sum_{\substack{1 \leqslant \ell \leqslant \min(w, w_\mathbf{r}) \\ \ell \text{ odd}}} C_\ell$$

where $C_\ell = \binom{n}{\ell}\binom{n-\ell}{w-\ell}\binom{n-w}{w_\mathbf{r}-\ell}$.

Proof In order to evaluate $\tilde{p}$, we have to determine how many pairs of vectors are such that their product yields 1 on a given coordinate. The total number of ordered pairs $(\mathbf{x}, \mathbf{r})$ is $\binom{n}{w}\binom{n}{w_\mathbf{r}}$. Among those, we need to count how many are such that $z_k = 1$. We note that

$$z_k = \sum_{\substack{i+j=k \bmod n \\ 0 \leq i,j \leq n-1}} x_i r_j .$$

We need therefore to count the number of couples $(\mathbf{x}, \mathbf{r})$ such that we have $x_i r_{k-i} = 1$ an odd number of times when i ranges over $\{0, \ldots, n-1\}$ (and $k - i$ is understood modulo n). Let us count the number C_ℓ of couples $(\mathbf{x}, \mathbf{r})$ such that $x_i r_{k-i} = 1$ exactly ℓ times. For $\ell > \min(w, w_\mathbf{r})$ we must have $C_\ell = 0$. For $\ell \leq \min(w, w_\mathbf{r})$ we have $\binom{n}{\ell}$ choices for the set of coordinates i such that $x_i = r_{k-i} = 1$, then $\binom{n-\ell}{w-\ell}$ remaining choices for the set of coordinates i such that $x_i = 1$ and $r_{k-i} = 0$, and finally $\binom{n-w}{w_\mathbf{r}-\ell}$ remaining choices for the set of coordinates i such that $x_i = 0$ and $r_{k-i} = 1$. Hence $C_\ell = \binom{n}{\ell}\binom{n-\ell}{w-\ell}\binom{n-w}{w_\mathbf{r}-\ell}$. By summing all possible odd values for ℓ, $1 \leqslant \ell \leqslant \min(w, w_\mathbf{r})$, we obtain the number of ordered pairs $(\mathbf{x}, \mathbf{r})$ such that $z_k = 1$ and divide this value by the total number of choices for $(\mathbf{x}, \mathbf{r})$ to compute $\tilde{p}$. □

4.2 Analysis of the error vector

Let $\mathbf{x}, \mathbf{y}$ (resp. $\mathbf{r}_1, \mathbf{r}_2$) be independent random vectors chosen uniformly among all binary vectors of weight w (resp. $w_{\mathbf{r}}$).

By independence of $(\mathbf{x}, \mathbf{r}_2)$ with $(\mathbf{y}, \mathbf{r}_1)$, the k-th coordinates of $\mathbf{x} \cdot \mathbf{r}_2$ and of $\mathbf{r}_1 \cdot \mathbf{y}$ are independent, and they are Bernoulli distributed with parameter $\tilde{p}$ given by Proposition 2. Therefore their modulo 2 sum $\mathbf{t} = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y}$ is Bernoulli distributed with

$$\begin{cases} \Pr[t_k = 1] = \Pr[\mathbf{x}\mathbf{r}_2 = 0 \ \& \ \mathbf{r}_1\mathbf{y} = 1] + \Pr[\mathbf{x}\mathbf{r}_2 = 1 \ \& \ \mathbf{r}_1\mathbf{y} = 0] = 2\tilde{p}(1-\tilde{p}), \\ \Pr[t_k = 0] = \Pr[\mathbf{x}\mathbf{r}_2 = 0 \ \& \ \mathbf{r}_1\mathbf{y} = 0] + \Pr[\mathbf{x}\mathbf{r}_2 = 1 \ \& \ \mathbf{r}_1\mathbf{y} = 1] = (1-\tilde{p})^2 + \tilde{p}^2. \end{cases} \tag{4}$$

Finally, by adding modulo 2 coordinate-wise the two independent vectors $\mathbf{e}$ and $\mathbf{t}$, we obtain the distribution of the coordinates of the error vector $\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$ given by the following proposition:

Proposition 3 *Let* $\mathbf{x}, \mathbf{y}$ *(resp.* $\mathbf{r}_1, \mathbf{r}_2$*, resp.* $\mathbf{e}$*) be uniformly random vectors of weight* w *(resp.* $w_{\mathbf{r}}$*, resp.* $w_{\mathbf{e}}$*). We suppose furthermore that the random vectors* $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2, \mathbf{e}$ *are independent. Let* $\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e} = (e'_0, \ldots, e'_{n-1})$*. Then, for every* $k \in \{0, \ldots, n-1\}$ *we have:*

$$\begin{cases} \Pr[e'_k = 1] = 2\tilde{p}(1-\tilde{p})(1 - \frac{w_{\mathbf{e}}}{n}) + \left((1-\tilde{p})^2 + \tilde{p}^2\right)\frac{w_{\mathbf{e}}}{n}, \\ \Pr[e'_k = 0] = \left((1-\tilde{p})^2 + \tilde{p}^2\right)(1 - \frac{w_{\mathbf{e}}}{n}) + 2\tilde{p}(1-\tilde{p})\frac{w_{\mathbf{e}}}{n}. \end{cases} \tag{5}$$

Proof The vectors $\mathbf{x} \cdot \mathbf{r}_2$, $\mathbf{r}_1 \cdot \mathbf{y}$ and $\mathbf{e}$ are clearly independent. The k-th coordinate of $\mathbf{e}$ is Bernoulli distributed with parameter $w_{\mathbf{e}}/n$. The random Bernoulli variable e'_k is therefore the sum modulo 2 of three independent Bernoulli variables of parameters $\tilde{p}$ for the first two and of parameter $w_{\mathbf{e}}/n$ for the third one. Equation (5) is obtained in a similar way as Eq. (4). □

Proposition 3 gives us the probability that a coordinate of the error vector $\mathbf{e}'$ is 1. In our simulations, which occur in the regime $w = \alpha\sqrt{n}$ with constant α, we make the simplifying assumption that the coordinates of $\mathbf{e}'$ are independent, meaning that the weight of $\mathbf{e}'$ follows a binomial distribution of parameter $p^\star$, where $p^\star$ is defined as $\Pr[e'_k = 1]$ in Eq. (5):

$$p^\star = 2\tilde{p}\,(1-\tilde{p})\left(1 - \frac{w_{\mathbf{e}}}{n}\right) + \left((1-\tilde{p})^2 + \tilde{p}^2\right)\frac{w_{\mathbf{e}}}{n}. \tag{6}$$

This approximation will give us, for $0 \le d \le \min\left(2 \times w \times w_{\mathbf{r}} + w_{\mathbf{e}}, n\right)$,

$$\Pr[\omega\left(\mathbf{e}'\right) = d] = \binom{n}{d}\left(p^\star\right)^d\left(1-p^\star\right)^{(n-d)}. \tag{7}$$

4.3 Supporting elements for our modelization

Figure 3 shows the results of the simulations on the distribution of the weight of the error vector, together with the distribution of the associated binomial law of parameter $p^\star$. It is important to notice that error vectors are more likely to have a weight closer to the mean than predicted by the binomial distribution. This in turn, implies that the error is *less likely* to be of larger weight than if it was binomially distributed. This is for instance illustrated on the parameter set corresponding to real parameters used for 128 bits security. For cryptographic purposes we are mainly interested in large weight occurrences which are the ones that may induce decoding errors. These results show that the probability of obtaining a large weight is closer but smaller for the error weight distribution of $\mathbf{e}'$ rather than for the binomial

Table 2 Parameters used for simulation, see Fig. 3

Parameter set	w	$w_{\mathbf{e}} = w_{\mathbf{r}}$	n	$n_1 n_2$	$p^\star$
hqc-128	66	75	17,669	17,664	0.3398

Table 3 Simulated probabilities of large weights for hqc-128 for the distributions of the error vector and the binomial approximation

	0.1%	0.01%	0.001%	0.0001%
Error vectors	6169	6203	6232	6257
Binomial approximation	6197	6237	6272	6301

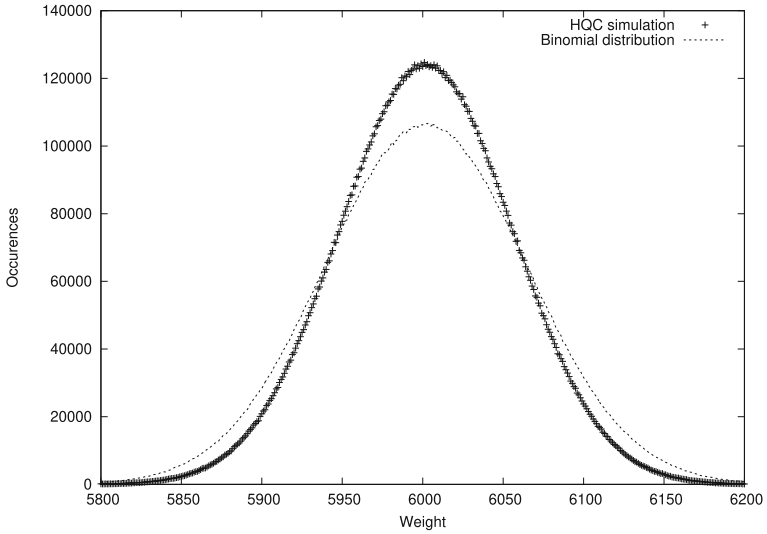

Fig. 3 Comparison of the weight of an error vector $\mathbf{e}'$ generated using hqc-128 parameters (Table 2) and the binomial distribution of parameter $p^\star$ (Eq. (6))

approximation. This supports our modelization and the fact that computing the decoding failure probability with this binomial approximation permits to obtain an upper bound on the real DFR. This will be confirmed in the next sections by simulations with real weight parameters (but smaller lengths).

Examples of simulations. We consider a parameter set that corresponds to cryptographic parameters and for which we simulate the error distribution versus the binomial approximation together with the probability of obtaining large error weights. We computed vectors of length n, where n is the smallest primitive prime greater than $n_1 n_2$, and then truncated the last $l = n - n_1 n_2$ bits before measuring the Hamming weight of the vectors.

Simulation results. Simulation results are shown Fig. 3. We computed the weights such that 0.1%, 0.01%, 0.001% and 0.0001% of the vectors are of weight greater than this value, to study how often extreme weight values occur. Results are presented in Table 3.

As we can see from these, extreme weight values happen more often in the case of the binomial approximation. Since these cases are the ones that may lead to decoding failure, this approximation leads to conservative DFR estimations.

Comparison with the previous analysis in [2]: in the case of decoding with BCH and repetition codes for security parameter 128 bits, the present analysis is sharper, and leads to

a DFR in 2^{-154} when the previous one lead to 2^{-128}. In practice, this allows to reduce by 3% the key size in the case of the BCH-repetition code decoder of [2].

The distribution of the error vector being established, next section studies different possibilities for the auxiliary error-correcting code in order to obtain better overall parameters.

5 Analysis of different auxiliary error-correcting codes

In this section, we discuss different possibilities for the public code used in HQC encryption, starting with the original proposition of BCH with repetition codes, then switching to concatenated Reed–Muller and Reed–Solomon codes. We analyze their respective DFR, and refine the original analysis. Finally, we propose new sets of parameters, improving on the original proposal.

5.1 Tensor product of BCH and repetition codes

Section 4 allowed us to determine the distribution of the weight of the error vector $\mathbf{e}'$. Now, in the decryption part, we need to decode this error vector $\mathbf{e}'$. At this point any decodable code can be used. In this section we are going to study tensor product codes of BCH and repetition codes, as in [2].

First we recall the definition of tensor product codes.

Definition 2 (Tensor Product Code) Let $\mathcal{C}_1$ (resp. $\mathcal{C}_2$) be an $[n_1, k_1, d_1]$ (resp. $[n_2, k_2, d_2]$) linear code over $\mathbb{F}_2$. The *Tensor Product Code* of $\mathcal{C}_1$ and $\mathcal{C}_2$ denoted $\mathcal{C}_1 \otimes \mathcal{C}_2$ is defined as the set of all $n_2 \times n_1$ matrices whose rows are codewords of $\mathcal{C}_1$ and whose columns are codewords of $\mathcal{C}_2$.

More formally, if $\mathcal{C}_1$ (resp. $\mathcal{C}_2$) is generated by $\mathbf{G}_1$ (resp. $\mathbf{G}_2$), then

$$\mathcal{C}_1 \otimes \mathcal{C}_2 = \left\{ \mathbf{G}_2^\top \mathbf{X} \mathbf{G}_1 \text{ for } \mathbf{X} \in \mathbb{F}_2^{k_2 \times k_1} \right\} \tag{8}$$

Remark 1 Using the notation of the above definition, the tensor product of two linear codes is an $[n_1 n_2, k_1 k_2, d_1 d_2]$ linear code.

Specifying the tensor product code In order to provide strong guarantees on the decryption failure probability of the HQC cryptosystem, the authors of [2] choose to restrict to a tensor product code $\mathcal{C} = \mathcal{C}_1 \otimes \mathcal{C}_2$, where $\mathcal{C}_1$ is a BCH(n_1, k_1, δ_1) code of length n_1, dimension k_1, and correcting capability δ_1 (i.e. it can correct any pattern of δ_1 errors), and $\mathcal{C}_2$ is the repetition code of length n_2 and dimension 1, denoted $\mathbb{1}_{n_2}$ (notice that $\mathbb{1}_{n_2}$ can decode up to $\delta_2 = \lfloor \frac{n_2 - 1}{2} \rfloor$).

In HQC, a message $\mathbf{m} \in \mathbb{F}_2^{k_1}$ is first encoded into $\mathbf{m}_1 \in \mathbb{F}_2^{n_1}$ with a BCH$(n_1, k_1 = k, \delta_1)$ code, then each coordinate $\mathbf{m}_{1,i}$ of $\mathbf{m}_1$ is re-encoded into $\tilde{\mathbf{m}}_{1,i} \in \mathbb{F}_2^{n_2}$ with a repetition code $\mathbb{1}_{n_2}$. This encoding method is known as a tensor product code in the literature [10, Chap. 18]. We denote $n_1 n_2$ the length of the tensor product code[3] (its dimension is $k = k_1 \times 1$), and by $\tilde{\mathbf{m}}$ the resulting encoded vector, i.e. $\tilde{\mathbf{m}} = \left(\tilde{\mathbf{m}}_{1,1}, \ldots, \tilde{\mathbf{m}}_{1,n_1}\right) \in \mathbb{F}_2^{n_1 n_2}$.

In the original HQC scheme, we have $\mathcal{C} = \mathcal{C}_1 \otimes \mathcal{C}_2$ where $\mathcal{C}_1$ is a BCH$(n_1, k_1 = k, d_1)$ code and $\mathcal{C}_2 = \mathbb{1}_{n_2}$ the $[n_2, k_2 = 1, d_2 = n_2]$ repetition code, that can decode up to $\delta_2 = \lfloor \frac{n_2 - 1}{2} \rfloor$.

[3] In practice, the length is the smallest primitive prime n greater than $n_1 n_2$ to avoid algebraic attacks.

The efficient algorithm used for the repetition code is the majority decoding. Formally:

$$\mathbb{1}_{n_2}.\mathsf{Decode}(\tilde{\mathbf{m}}_{1,j}) = \begin{cases} 1 \text{ if } \sum_{i=0}^{n_2-1} \tilde{\mathbf{m}}_{1,j,i} \geq \lceil \frac{n_2+1}{2} \rceil, \\ 0 \text{ otherwise.} \end{cases} \tag{9}$$

The decoding of BCH codes is discussed in the next section.

5.1.1 BCH codes

For any positive integers $m \geq 3$ and $t \leq 2^{m-1}$, there exists a binary BCH code with the following parameters [8]:

- block length $n = 2^m - 1$;
- number of parity-check digits $n - k \leq m\delta$, with δ, the correcting capacity of the code and k the number of information bits;
- and minimum (designed) distance $d_{min} \geq 2\delta + 1$.

We denote this code by BCH(n, k, δ). Let α be a primitive element in $\mathbb{F}_{2^m}$, the generator polynomial $g(x)$ of the BCH(n, k, δ) code is given by:

$$g(x) = \text{LCM}\{\phi_1(x), \phi_2(x), \cdots, \phi_{2\delta}(x)\}$$

with $\phi_i(x)$ being the minimal polynomial of α^i (refer to [8] for more details on generator polynomial) and LCM being the least common multiple.

Depending on the security level, we construct shortened BCH codes[4] from the following BCH codes:

- a [511, 241, 36] BCH code shortened to [398, 128, 36] for $\lambda = 128$;
- a [1023, 513, 57] BCH code shortened to [702, 192, 57] for $\lambda = 192$;
- a [1023, 483, 60] BCH code shortened to [796, 256, 60] for $\lambda = 256$.

5.1.2 Decryption failure probability

With a tensor product code $\mathcal{C} = \text{BCH}(n_1, k_1, \delta) \otimes \mathbb{1}_{n_2}$ as defined above, a decryption failure occurs whenever the decoding algorithm of the BCH code does not succeed in correcting errors that would have arisen after wrong decodings by the repetition code. Therefore, the analysis of the decryption failure probability is again split into three steps: evaluating the probability that the repetition code does not decode correctly, the conditional probability of a wrong decoding for the BCH code given an error weight and finally, the decryption failure probability using the law of total probability.

We first focus on the probability that an error occurs while decoding the repetition code. As shown in Sect. 4, the probability for a coordinate of $\mathbf{e}' = \mathbf{x}\mathbf{r}_2 - \mathbf{r}_1\mathbf{y} + \mathbf{e}$ to be 1 is $p^\star$ (see Eq. (5)). As mentioned above, $\mathbb{1}_{n_2}$ can decode up to $\delta_2 = \lfloor \frac{n_2-1}{2} \rfloor$ errors. Therefore, the probability that $\mathbb{1}_{n_2}$ does not decode correctly for odd n_2 is given by:

$$p_i = \sum_{i=\lceil n_2/2 \rceil}^{n_2} \binom{n_2}{i} p^{\star i} (1 - p^\star)^{n_2 - i}. \tag{10}$$

[4] Shortening a code can be seen as expurgating (removing codewords) then puncturing it (removing columns from the generator matrix). This operation keeps the $n - k$ redundancy bits and decreases k and n at the same pace. In this case, shortening BCH codes does not affect the designed distance.

For n_2 even, when the error has weight $\frac{n_2}{2}$, the probability of not decoding correctly is 1/2. We choose the arbitrary convention to decode the word as $\mathbf{0}$ in such a case. It follows that the probability of not decoding correctly becomes:

$$p_i = \frac{1}{2}\binom{n_2}{\frac{n_2}{2}} p^{\star\frac{n_2}{2}}(1-p^\star)^{\frac{n_2}{2}} + \sum_{i=\frac{n_2}{2}+1}^{n_2}\binom{n_2}{i} p^{\star i}(1-p^\star)^{n_2-j}. \tag{11}$$

Notice that in practice (except for simulations) we only consider odd n_2 in our parameters.

We now focus on the BCH(n_1, k_1, δ_1) code, and recall that it can correct any pattern of δ_1 errors. Now, the probability p_e that the BCH(n_1, k_1, δ_1) code fails to decode correctly the encoded message $\mathbf{m}_1$ back to $\mathbf{m}$ is given by the probability that an error occurred on at least $\delta_1 + 1$ blocks of the repetition code. Therefore, we have the following theorem:

Theorem 4 *The probability p_e that the BCH(n_1, k_1, δ_1) code does not decode correctly is given by:*

$$p_e = \sum_{l=\delta_1+1}^{n_1}\binom{n_1}{l} p_i^l (1-p_i)^{n_1-l}. \tag{12}$$

5.2 Concatenated Reed–Muller and Reed–Solomon codes

In this section we study the impact of using a new family of auxiliary error-correcting codes: instead of the tensor product codes used in the HQC framework [2] we propose to consider the concatenation of Reed–Solomon and duplicated first order Reed–Muller codes. We denote this instantiation of the HQC framework by HQC-RMRS.

5.2.1 Construction

Definition 3 (Concatenated codes) A concatenated code consists of an external code $[n_e, k_e, d_e]$ over $\mathbb{F}_q$ and an internal code $[n_i, k_i, d_i]$ over $\mathbb{F}_2$, with $q = 2^{k_i}$. We use a bijection between elements of $\mathbb{F}_q$ and the words of the internal code, this way we obtain a transformation:

$$\mathbb{F}_q^{n_e} \to \mathbb{F}_2^N$$

where $N = n_e n_i$. The external code is thus transformed into a binary code of parameters $[N = n_e n_i, K = k_e k_i, D \geqslant d_e d_i]$.

For the external code, we chose Reed–Solomon codes of dimensions 16, 24 or 32 over $\mathbb{F}_q = \mathbb{F}_{256}$, depending on whether we want to achieve a total dimension of 128, 192, or 256. For the internal code, we chose the Reed–Muller code [128, 8, 64] that we are going to duplicate between 2 and 6 times (*i.e.* duplicating each bit to obtain codes of parameters [256, 8, 128], [512, 8, 256], [786, 8, 384]).

Decoding: We perform maximum likelihood decoding (MLD) on the internal code. This yields a vector of $\mathbb{F}_q^{n_e}$ that we then decode using an algebraic decoder for the Reed–Solomon code.

Decoding the internal Reed–Muller code: The Reed–Muller code of order 1 and length 2^m can be decoded using a fast Hadamard transform [10, Chap. 14]. Recall that for any function $F : \mathbb{F}_2^m \to \mathbb{Z}$ the Hadamard (or Walsh-Hadamard) transform computes the function

$\hat{F} : \mathbb{F}_2^m \to \mathbb{Z}$ defined by $\hat{F}(\mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{F}_2^m} F(\mathbf{y})(-1)^{\mathbf{x}\cdot\mathbf{y}}$. Recall also that, for $n = 2^m$, vectors of $\mathbb{F}_2^n$ can be viewed as functions $f : \mathbb{F}_2^m \to \mathbb{F}_2$, and that decoding the Reed–Muller code involves applying the fast Hadamard transform to $F = (-1)^f$ and finding $\mathbf{x}$ that maximizes $|\hat{F}(\mathbf{x})|$.

The decoding algorithm needs to be slightly adapted when decoding duplicated codes. For example, if the Reed–Muller of length 2^m is duplicated three times, we create the function $F : \mathbb{F}_2^m \to \{3, 1, -1, -3\}$ (which can be thought of as a 2^m-tuple of symbols from $\{3, 1, -1, -3\}$) by transforming every block of three bits $y_1 y_2 y_3$ of the received vector of length 3.2^m to

$$(-1)^{y_1} + (-1)^{y_2} + (-1)^{y_3}.$$

More generally, when duplicating t times we transform the received function F into a function taking its values in the set of all integer even values between $-t$ and t when t is even and all odd values between $-t$ and t when t is odd.

We then apply the Hadamard transform to the function F, yielding $\hat{F}$. We take the maximum value of $\hat{F}$ and $\mathbf{x} \in \mathbb{F}_2^m$ that maximizes the value of $|\hat{F}(\mathbf{x})|$. If $\hat{F}(\mathbf{x})$ is positive, then the closest codeword is $\mathbf{x}\mathbf{G}_H$ where $\mathbf{G}_H$ is the generator matrix of the duplicated Hadamard code (without the all-one-vector). If $\hat{F}(\mathbf{x})$ is negative, then we need to add the all-one-vector to it.

5.2.2 Decoding failure rate analysis

We now consider the decoding failure rate of the concatenated code. We first provide two bounds on the MLD error probability of the duplicated Reed–Muller code: a first simple union bound and a second more accurate one. These bounds can then be plugged into the decoding error probability for the bounded distance decoder of the Reed–Solomon code.

Proposition 5 (Simple Upper Bound for the DFR of the internal code) *Over a BSC of transition probability p the DFR of a duplicated Reed–Muller code of dimension* 8 *and minimal distance d_i can be upper bounded by:*

$$p_i = 255 \sum_{j=d_i/2}^{d_i} \binom{d_i}{j} p^j (1-p)^{d_i - j}.$$

Proof For any linear code $\mathcal{C}$ of length n, when transmitting a codeword $\mathbf{c}$, the probability that the channel makes the received word $\mathbf{y}$ at least as close to a word $\mathbf{c}' = \mathbf{c} + \mathbf{x}$ as $\mathbf{c}$ (for $\mathbf{x}$ a non-zero word of $\mathcal{C}$ and $w = \omega(\mathbf{x})$ the weight of $\mathbf{x}$) is:

$$\sum_{j \geqslant w/2} \binom{w}{j} p^j (1-p)^{n-j}.$$

By the union bound applied on the different non-zero codewords $\mathbf{x}$ of $\mathcal{C}$, we obtain that the probability of a decoding failure can thus be upper bounded by:

$$\sum_{\mathbf{x} \in \mathcal{C}, \mathbf{x} \neq \mathbf{0}} \sum_{j \geqslant w/2} \binom{w}{j} p^j (1-p)^{n-j}.$$

There are 255 non-zero words in a [128,8,64] Reed–Muller code, 254 of weight 64 and one of weight 128. The contribution of the weight 128 vector is smaller than the weight 64 vectors, hence by applying the previous bound to duplicated Reed–Muller codes we obtain the result.

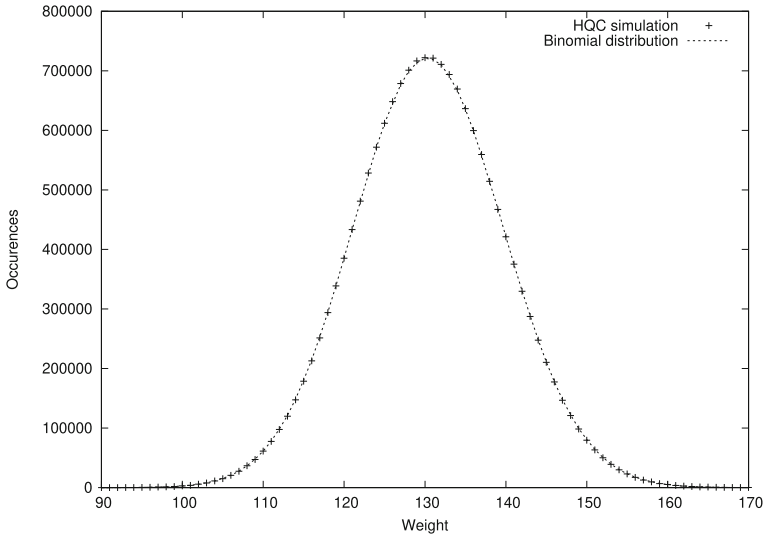

Fig. 4 The binomial distribution vs the actual weight distribution of the HQC error vector restricted to the support of a Reed–Muller code. Parameters correspond to 128 bits of security, thus the support length is 384

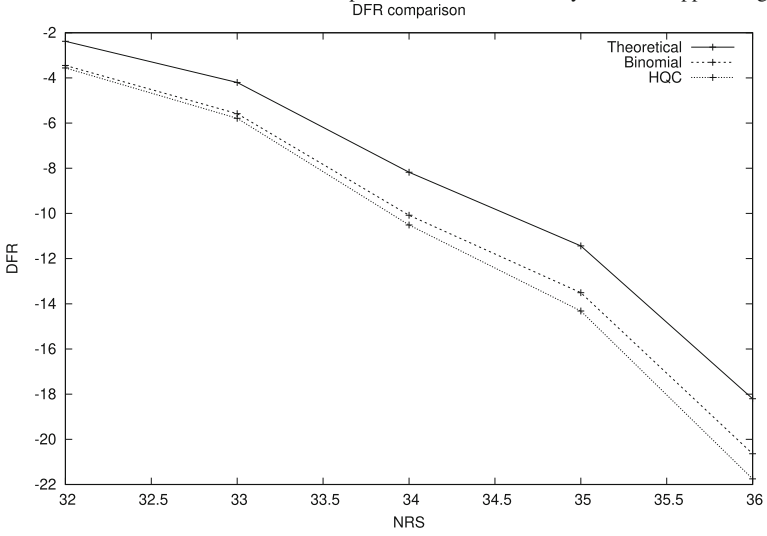

Fig. 5 Comparison between the DFR from 7 (Theoretical) and the actual DFR of concatenated codes against approximation by a BSC (binomial) and against HQC error vectors (HQC). Parameters simulated are derived from those of HQC for 128 security bits: $w = 66$, $w_{\mathbf{r}} = w_{\mathbf{e}} = 75$, a [384, 8, 192] duplicated Reed–Muller code for internal code and a [NRS, 16] Reed–Solomon code for external code

Better upper bound on the decoding error probability for the internal code. The previous simple bound pessimistically assumes that decoding fails when more than one codeword minimizes the distance to the received vector. The following bound improves the previous one by taking into account the fact that decoding can still succeed with probability $1/2$ when exactly two codewords minimize the distance to the received vector. □

Proposition 6 (Improved Upper Bound for the DFR of the internal code) *Over a BSC of transition probability p, the DFR of a duplicated first-order Reed–Muller code of dimension* 8 *and minimal distance* d_i *can be upper bounded by:*

$$p_i = \sum_{w=d_i/2}^{n} \mathfrak{A}_w p^w (1-p)^{n-w}$$

where

$$\begin{aligned}\mathfrak{A}_w = \min\Bigg[& \binom{n}{w}, \frac{1}{2}255\binom{d_i}{d_i/2}\binom{d_i}{w-d_i/2} \\ & + 255\sum_{j=d_i/2+1}^{d_i}\binom{d_i}{j}\binom{d_i}{w-j} \\ & + \frac{1}{2}\binom{255}{2}\sum_{j=0}^{d_i/2}\binom{d_i/2}{j}^3\binom{d_i/2}{w-d_i+j}\Bigg].\end{aligned}$$

Proof Let E be the decoding error event. Let $\mathbf{e}$ be the error vector.

- Let A be the event where the closest non-zero codeword $\mathbf{c}$ to the error is such that $d(\mathbf{e}, \mathbf{c}) = d(\mathbf{e}, \mathbf{0}) = \omega(\mathbf{e})$.
- Let B be the event where the closest non-zero codeword $\mathbf{c}$ to the error vector is such that $d(\mathbf{e}, \mathbf{c}) < \omega(\mathbf{e})$.
- Let $A' \subset A$ be the event where the closest non-zero codeword $\mathbf{c}$ to the error vector is such that $d(\mathbf{e}, \mathbf{c}) = \omega(\mathbf{e})$ and such a vector is unique, meaning that for every $\mathbf{c}' \in \mathcal{C}$, $\mathbf{c}' \neq \mathbf{c}$, $\mathbf{c}' \neq \mathbf{0}$, we have $d(\mathbf{e}, \mathbf{c}') > \omega(\mathbf{e})$.
- Finally, let A'' be the event that is the complement of A' in A, meaning the event where the closest non-zero codeword $\mathbf{c}$ to the error is at distance $\omega(\mathbf{e})$ from $\mathbf{e}$, and there exists at least one codeword $\mathbf{c}'$, $\mathbf{c}' \neq \mathbf{c}$, $\mathbf{c}' \neq \mathbf{0}$, such that $d(\mathbf{e}, \mathbf{c}') = d(\mathbf{e}, \mathbf{c}) = \omega(\mathbf{e})$.

The probability space is partitioned as $\Omega = A \cup B \cup C = A' \cup A'' \cup B \cup C$, where C is the complement of $A \cup B$. When C occurs, the decoder always decodes correctly, *i.e.* $P(E|C) = 0$. We therefore write:

$$P(E) = P(E|A')P(A') + P(E|A'')P(A'') + P(E|B)P(B).$$

When the event A' occurs, the decoder chooses at random between the two closest codewords and is correct with probability 1/2, *i.e.* $P(E|A') = 1/2$. We have $P(E|B) = 1$ and writing $P(E|A'') \leqslant 1$, we have:

$$\begin{aligned}P(E) &\leqslant \frac{1}{2}P(A') + P(A'') + P(B) \\ &= \frac{1}{2}(P(A') + P(A'')) + \frac{1}{2}P(A'') + P(B) \\ P(E) &\leqslant \frac{1}{2}P(A) + \frac{1}{2}P(A'') + P(B). \end{aligned} \tag{13}$$

Now for $X = A, A', A'', E$, let us denote by X_w the intersection of event X with the event "$\omega(\mathbf{e}) = w$".

We shall write

$$P(E) = \sum_{w=d_i/2}^{n} P(E_w). \tag{14}$$

Similarly to (13) we have

$$P(E_w) \leqslant \frac{1}{2}P(A_w) + \frac{1}{2}P(A''_w) + P(B_w). \tag{15}$$

It now remains to evaluate $P(A_w)$, $P(B_w)$ and $P(A''_w)$. For $P(B_w)$ and $P(A_w)$ we have the straightforward union bounds:

$$P(B_w) \leqslant 255 \sum_{j=d_i/2+1}^{d_i} \binom{d_i}{j}\binom{d_i}{w-j} p^w (1-p)^{n-w} \tag{16}$$

with $n = 2d_i$ the length of the inner code, and where we use the convention that a binomial coefficient $\binom{\ell}{k} = 0$ whenever $k < 0$ or $k > \ell$.

$$P(A_w) \leqslant 255\binom{d_i}{d_i/2}\binom{d_i}{w-d_i/2} p^w (1-p)^{n-w} \tag{17}$$

and it remains to find an upper bound on $P(A''_w)$.

We have:

$$P(A''_w) \leqslant \sum_{\mathbf{c},\mathbf{c}'} P(A^w_{\mathbf{c},\mathbf{c}'})$$

where the sum is over pairs of distinct non-zero codewords and where:

$$A^w_{\mathbf{c},\mathbf{c}'} = \{d(\mathbf{e},\mathbf{c}) = d(\mathbf{e},\mathbf{c}') = \omega(\mathbf{e}) = w\}.$$

This event is equivalent to the error meeting the supports of $\mathbf{c}$ and $\mathbf{c}'$ on exactly half of their coordinates. All codewords except the all-one vector have weight d_i, and any two codewords of weight d_i either have non-intersecting supports or intersect in exactly $d/2$ positions. $P(A^w_{\mathbf{c},\mathbf{c}'})$ is largest when $\mathbf{c}$ and $\mathbf{c}'$ have weight d and non-zero intersection. In this case we have:

$$P(A^w_{\mathbf{c},\mathbf{c}'}) = \sum_{j=0}^{d_i/2} \binom{d_i/2}{j}^3 \binom{d_i/2}{w-d_i+j} p^w (1-p)^{n-w}. \tag{18}$$

Hence

$$\begin{aligned} P(A''_w) &\leqslant \sum_{\mathbf{c},\mathbf{c}'} P(A_{\mathbf{c},\mathbf{c}'}) \\ &\leqslant \binom{255}{2} \sum_{j=0}^{d_i/2} \binom{d_i/2}{j}^3 \binom{d_i/2}{w-d_i+j} p^w (1-p)^{n-w}. \end{aligned} \tag{19}$$

Plugging Eqs. (17), (16) and (19) into (15), and then applying (14) we obtain the result. □

Remark 2 The previous formula permits to obtain a lower bound on the correct decoding probability of the duplicated Reed–Muller code; when the error rate gets smaller the bound becomes closer to the real value of the decoding probability. For cryptographic parameters the approximation is less precise, which means that the DFR obtained will be conservative compared to what happens in practice. We performed simulations to compare the real DFR with the theoretical one from Proposition 6 for [384, 8, 192] and [640, 8, 320] duplicated Reed–Muller codes using $p^\star$ values from actual parameters. Simulation results are presented in Table 4.

Table 4 Comparison between the observed DFR and the formula from Proposition 6

Security level	$p^\star$	Reed–Muller code	DFR from Proposition (6)	Observed DFR
128	0.3398	[384, 8, 192]	− 10.79	− 10.96
192	0.3618	[640, 8, 320]	− 14.14	− 14.39
256	0.3725	[640, 8, 320]	− 11.30	− 11.48

Results are presented as $\log_2(DFR)$

Remark 3 Propositions 5 and 6 have been derived with a BSC model for the distribution of the HQC error vector restricted to the support of a (duplicated) Reed–Muller code. Figure 4 compares the actual weight distribution of the error vector to the binomial distribution when restricted to this relatively small number of bits. We observe that they are virtually identical, meaning that a small proportion of HQC bits do behave as independent and identically distributed Bernoulli variables.

Theorem 7 *(Decoding failure rate of the concatenated code) Using a Reed–Solomon code* $[n_e, k_e, d_e]_{\mathbb{F}_{256}}$ *as the external code, the DFR of the concatenated code can be upper bounded by:*

$$\sum_{l=\delta_e+1}^{n_e} \binom{n_e}{l} p_i^l (1-p_i)^{n_e-l}$$

Where $d_e = 2\delta_e + 1$ *and* p_i *is defined as in Proposition 5.*

5.2.3 Simulation results

In Fig. 5, we tested the DFR of the concatenated codes against both symmetric binary channels and HQC vectors, and compared the results with the theoretical value obtained using propositions 6 and 7.

5.3 Proposed parameters

From the DFR analysis we derive new parameters for the HQC cryptosystem and for the HQC-RMRS variant introduced in section 5.2. These are described on Table 5 and Table 6.

The values of w, $w_{\mathbf{r}}$ and $w_{\mathbf{e}}$ were chosen such that the best known attacks against the HQC cryptosystem have a complexity $> 2^\lambda$. The best algorithms to solve the syndrome decoding problem are the information set decoding (ISD) algorithms, which have been studied in [4]. Moreover, because of the quasi-cyclic structure in HQC, we also need to consider the decoding one out of many (DOOM) attack [14]. More details on how the weight values were computed can be found in the HQC submission to the NIST standardization process [1].

6 Conclusion

In this work, we proposed better parameters for the HQC post-quantum cryptosystem submitted to the NIST standardization process. These parameters have been investigated thoroughly thanks to the observation that the codes used in the original submission (BCH and repetition

Table 5 New proposed parameters for the HQC cryptosystem (security is in bits)

Instance	n_1	n_2	δ	n	k	w	$w_{\mathbf{r}} = w_{\mathbf{e}}$	security λ	p_{fail}
HQC-128	398	51	36	20,323	128	66	75	128	$< 2^{-128}$
HQC-192	702	61	57	42,829	192	100	114	192	$< 2^{-192}$
HQC-256	796	85	60	67,679	256	131	149	256	$< 2^{-256}$

The tensor product code used consists of a $\mathbb{1}_{n_2}$ repetition code and a $[n_1, k]$ shortened BCH code with a decoding capacity of δ errors

Table 6 New proposed parameters for the HQC-RMRS cryptosystem (security is in bits)

Instance	n_1	n_2	n	k	w	$w_{\mathbf{r}} = w_{\mathbf{e}}$	security λ	p_{fail}
HQC-RMRS-128	46	384	17,669	128	66	75	128	$< 2^{-128}$
HQC-RMRS-192	56	640	35,851	192	100	114	192	$< 2^{-192}$
HQC-RMRS-256	90	640	57,637	256	131	149	256	$< 2^{-256}$

The concatenated code used consists of a $[n_2, 8, n_2/2]$ Reed–Muller code as the internal code, and a $[n_1, k, n_1 - k + 1]$ Reed–Solomon code as the external code

codes) are far from being optimal. We have derived theoretical bounds on the optimal code length in order for the resulting scheme to have negligible DFR. This requirement is crucial both to avoid key-recovery attacks exploiting decryption failures, and to apply the generic HHK transform that turns an IND-CPA public key encryption scheme into an IND-CCA KEM. We also provided a finer analysis of the error weight distribution, that matches simulation more closely, resulting in lower DFR. Finally, we suggested an alternative auxiliary code for HQC, namely concatenated Reed–Muller and Reed–Solomon codes, that yield better overall parameters together with an efficient implementation. The improvements suggested in this work have been integrated in the current version of the NIST submission, still in consideration in the fourth round of the standardization process.

A HHK transform applied to HQC

Let $\mathcal{E}$ be an instance of the HQC public key encryption scheme, as described in Fig. 1, and let $\mathcal{G}$, $\mathcal{H}$, and $\mathcal{K}$ be hash functions. The KEM-DEM version of the HQC cryptosystem is described in Fig. 6.

- $\mathsf{Setup}(1^\lambda)$: as before, except that k will be the length of the symmetric key being exchanged, typically $k = 256$.
- $\mathsf{KeyGen}(\mathsf{param})$: exactly as before.
- $\mathsf{Encapsulate}(\mathsf{pk})$: generate $\mathbf{m} \xleftarrow{\$} \mathbb{F}_2^k$ (this will serve as a seed to derive the shared key). Derive the randomness $\theta \leftarrow \mathcal{G}(\mathbf{m})$. Generate the ciphertext $c \leftarrow (\mathbf{u}, \mathbf{v}) = \mathcal{E}.\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m}, \theta)$, and derive the symmetric key $K \leftarrow \mathcal{K}(\mathbf{m}, \mathbf{c})$. Let $\mathbf{d} \leftarrow \mathcal{H}(\mathbf{m})$, and send $(\mathbf{c}, \mathbf{d})$.
- $\mathsf{Decapsulate}(\mathsf{sk}, \mathbf{c}, \mathbf{d})$: Decrypt $\mathbf{m}' \leftarrow \mathcal{E}.\mathsf{Decrypt}(\mathsf{sk}, \mathbf{c})$, compute $\theta' \leftarrow \mathcal{G}(\mathbf{m}')$, and (re-)encrypt $\mathbf{m}'$ to get $\mathbf{c}' \leftarrow \mathcal{E}.\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m}', \theta')$. If $\mathbf{c} \neq \mathbf{c}'$ or $\mathbf{d} \neq \mathcal{H}(\mathbf{m}')$ then abort. Otherwise, derive the shared key $K \leftarrow \mathcal{K}(\mathbf{m}, \mathbf{c})$.

Fig. 6 Description of our proposal HQC.KEM

References

1. Aguilar Melchor C., Aragon N., Bettaieb S., Bidoux L., Blazy O., Bos J., Deneuville J., Dion A., Gaborit P., Lacan J., Persichetti E., Robert J.-M., Véron P., Zémor G.: Hamming Quasi-Cyclic (HQC). NIST PQC Round **4**, 1–52 (2023).
2. Aguilar-Melchor C., Blazy O., Deneuville J.-C., Gaborit P., Zémor G.: Efficient encryption from random quasi-cyclic codes. IEEE Trans. Inf. Theory **64**(5), 3927–3943 (2018).
3. Barg A., Forney G.D.: Random codes: minimum distances and error exponents. IEEE Trans. Inf. Theory **48**(9), 2568–2573 (2002).
4. Canto Torres R., Sendrier N.: Analysis of information set decoding for a sub-linear error weight. In: International Workshop on Post-Quantum Cryptography, pp. 144–161. Springer (2016).
5. Gabidulin E.M., Ourivski A.V., Honary B., Ammar B.: Reducible rank codes and their applications to cryptography. IEEE Trans. Inf. Theory **49**(12), 3289–3293 (2003).
6. Guo Q., Johansson T., Stankovski P.: A key recovery attack on MDPC with CCA security using decoding errors. In: Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, 4–8 December 2016, Proceedings, Part I 22, pp. 789–815. Springer (2016).
7. Hofheinz D., Hövelmanns K., Kiltz E.: A modular analysis of the Fujisaki–Okamoto transformation. In: Theory of Cryptography Conference, pp. 341–371. Springer (2017).
8. Lin S., Costello D.J.: Error Control Coding, vol. 2. Prentice Hall, Englewood Cliffs (2004).
9. Löndahl C., Johansson T., Koochak Shooshtari M., Ahmadian-Attari M., Aref M.R.: Squaring attacks on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension. Des. Codes Cryptogr. **80**, 359–377 (2016).
10. MacWilliams F.J., Sloane N.J.A.: The Theory of Error Correcting Codes, vol. 16. Elsevier, Amsterdam (1977).
11. McEliece R.J.: A public-key cryptosystem based on algebraic coding theory. DSN Prog. Rep. **44**, 114–116 (1978).
12. Niederreiter H.: Knapsack-type cryptosystems and algebraic coding theory. Probl. Control Inf. Theory **15**(2), 159–166 (1986).
13. NIST: Post-Quantum Cryptography Standardization. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions.
14. Sendrier N.: Decoding one out of many. In: International Workshop on Post-Quantum Cryptography, pp. 51–67. Springer (2011).