HQC Beyond the BSC: Towards Error Structure-Aware Decoding

Marco Baldi[1], Sebastian Bitzer[2], Nicholas Lilla[1], and Paolo Santini[1]

[1] Università Politecnica delle Marche, Ancona, Italy
[2] Technical University of Munich, Munich, Germany
m.baldi@univpm.it, sebastian.bitzer@tum.de,
p.santini@univpm.it, lilla.nicholas@gmail.com

Abstract. In Hamming Quasi-Cyclic (HQC), one of the finalists in the NIST competition for the standardization of post-quantum cryptography, decryption relies on decoding a noisy codeword through a public error-correcting code. The noise vector has a special form that depends on the secret key (a pair of sparse polynomials). However, the decoder, which is currently employed in HQC, is agnostic to the secret key, operating under the assumption that the error arises from a Binary Symmetric Channel (BSC). In this paper, we demonstrate that this special noise structure can instead be leveraged to develop more powerful decoding strategies.

We first study the problem from a coding-theoretic perspective. The current code design, which admits a non-zero decryption failure rate, is close to optimal in the setting of a decoder that is agnostic to the error structure. We show that there are code-decoder pairs with a considerably shorter code length that can guarantee unique decoding by taking the error structure into account. This result is non-constructive, i.e., we do not provide an explicit code construction and it remains open whether efficient decoding is possible. Nevertheless, it highlights the potential that can be tapped by taking the error structure into account. We then argue that, in practice, the matter of decoding in HQC can be related to solving an instance of the noisy syndrome decoding problem, in which the parity-check matrix is constituted by the polynomials in the secret key. We show that, using decoders for Low-Density Parity-Check (LDPC) and Moderate-Density Parity-Check (MDPC) codes, one can significantly reduce the entity of the noise and, de facto, also the Decoding Failure Rate (DFR) of the HQC decoder.

This preliminary study leaves some open questions and problems. While it shows that decoding in HQC can be improved, the modeling of the DFR gets more complicated: even for the basic decoder we propose in this paper, we have not been able to devise a reliable DFR model. This is likely due to the fact that the decoder structure resembles the iterative nature of LDPC/MDPC decoders, for which devising a reliable DFR estimation is a well-known difficult problem.

Keywords: HQC · Decryption Failure Rate · Noisy Syndrome Decoding

1 Introduction

In code-based encryption schemes, the public key is usually a disguised representation of a secret error-correcting code $\mathscr{C}$ equipped with an efficient decoding algorithm. Recovering the structure of $\mathscr{C}$ shall be computationally infeasible: this is one of the security assumptions (the other being the hardness of decoding arbitrary linear codes). For instance, LEDAcrypt, BIKE, and Classic McEliece, some code-based cryptosystems participating in the NIST competition for the standardization of post-quantum cryptography [10], follow this paradigm: LEDAcrypt [5] relies on Low-Density Parity-Check (LDPC) codes, BIKE [4] exploits a special family of LDPC codes known as Moderate-Density Parity-Check (MDPC) codes, while Classic McEliece [2] employs binary Goppa codes.

HQC [9], another alternate in the NIST competition, departs from such a paradigm: no secret error-correcting code is required. Instead, a publicly known code $\mathscr{C}$, equipped with an efficient decoder Dec, is employed. The possibility to perform efficient decoding is based on the fact that the legitimate receiver, using the secret key, can perform some efficient manipulations of the ciphertext (essentially, a couple of polynomial multiplications and sums) resulting in $\mathbf{c} = \mathbf{t} + \mathbf{z}$, with $\mathbf{t} \in \mathscr{C}$ and

$$\mathbf{z} = \mathbf{x} \cdot \mathbf{r}^{(2)} + \mathbf{y} \cdot \mathbf{r}^{(1)} + \mathbf{e}. \tag{1}$$

All terms in the previous formula are sparse polynomials in the ring $\mathbb{F}_2[X]/(X^n - 1)$; hence $\mathbf{z}$ is sparse as well. In particular, $\mathbf{x}$ and $\mathbf{y}$ constitute the secret key while $\mathbf{r}^{(1)}$, $\mathbf{r}^{(2)}$ and $\mathbf{e}$ are ephemeral and are chosen uniformly at random by the sender. Decoding $\mathbf{c}$ through the public code $\mathscr{C}$ allows to correct $\mathbf{z}$.

The error $\mathbf{z}$ is modeled through a Binary Symmetric Channel (BSC); the code $\mathscr{C}$ is chosen accordingly, and, in principle, any error-correcting code $\mathscr{C}$ can be used. In all four rounds of the NIST competition, the authors of HQC have selected $\mathscr{C}$ as the tensor product between two error-correcting codes. This results in a good trade-off between computational efficiency, error-correction capability and the possibility to derive a solid and reliable Decoding Failure Rate (DFR) modeling. We remark that the availability of such a model is crucial, as IND-CCA2 security can only be achieved if the DFR is negligible in the security parameter.

1.1 Our Contribution

The decoder employed in HQC is agnostic to the secret key, i.e., it never explicitly takes advantage of the particular structure (1) of the error vector $\mathbf{z}$. However, the secret-key polynomials $\mathbf{x}$ and $\mathbf{y}$ are part of how $\mathbf{z}$ is computed: as we show in this paper, this information can be employed to devise more powerful decoding strategies.

Coding-Theory Perspective First, with coding theory arguments, we show that taking the structure of $\mathbf{z}$ and the knowledge of $\mathbf{x}$ and $\mathbf{y}$ into account enables

shorter code lengths. To this end, we first revisit fundamental bounds on the required code length that are implied by modeling errors as caused by a BSC. These bounds show that the code construction used in the round-4 version of HQC achieves a shorter length than would be possible if correct decryption has to be guaranteed. On the other hand, an arbitrary code-decoder pair cannot do considerably better than the current choice, that is, as long as the code design and the decoding algorithm follow the assumption that errors are caused by a BSC.

The situation changes when we consider the exact structure of $\mathbf{z}$ and make the decoder aware of $\mathbf{x}$ and $\mathbf{y}$. Using an argument similar to the classical Gilbert-Varshamov (GV) bound, we show that codes able to guarantee correct decryption exist with significantly shorter lengths than under the BSC model. As is typical for GV-like arguments, this result is not constructive, i.e., we do not provide an explicit code construction or an efficient decoding algorithm.

The Relation with Noisy-Syndrome Decoding When $\mathscr{C}$ is a tensor product code, the problem of recovering $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$ can be formulated as a noisy syndrome decoding problem [6]. Indeed, decoding the inner code yields an estimate $\widehat{\mathbf{z}}$ for $\mathbf{z}$. We view $\widehat{\mathbf{z}}$ as a noisy syndrome, as it can be expressed as

$$\widehat{\mathbf{z}} = \underbrace{\big(\mathrm{Rot}(\mathbf{x})||\mathrm{Rot}(\mathbf{y})\big)}_{\mathbf{H}}\,\mathbf{r}^\top + \Delta\mathbf{z} = \mathbf{H}\mathbf{r}^\top + \Delta\mathbf{z},$$

where $\mathrm{Rot}(\mathbf{x})$ and $\mathrm{Rot}(\mathbf{y})$ are the circulant matrices whose first rows are $\mathbf{x}$ and $\mathbf{y}$, respectively, while $\mathbf{r}$ is the length-$2n$ vector formed by the coefficients of the unkwnown polynomials $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$. The product $\mathbf{H}\mathbf{r}^\top$ is a syndrome affected by the noise contained in $\Delta\mathbf{z}$. The amount of noise depends on the quality of decoding: the lower the number of decoding errors, the smaller the weight of $\Delta\mathbf{z}$.

Since $\mathbf{x}$ and $\mathbf{y}$ are sparse polynomials, $\mathbf{H}$ is sparse as well. Then, $\mathbf{H}$ can be interpreted as the parity-check matrix of a Low-Density Parity-Check (LDPC) or Moderate-Density Parity-Check (MDPC) code, and $\mathbf{r}$ can be estimated using decoders for such codes. Note that these algorithms, such as the Bit Flipping decoder, naturally tolerate a syndrome affected by a moderate amount of noise.

Let $\widehat{\mathbf{r}}^{(1)}$ and $\widehat{\mathbf{r}}^{(2)}$ be the obtained estimates: they can be used to reduce the noise affecting $\mathbf{c}$, as one can update $\mathbf{c}$ as

$$\begin{aligned}
\widehat{\mathbf{c}} &= \mathbf{c} - \mathbf{x}\cdot\widehat{\mathbf{r}}^{(2)} - \mathbf{y}\cdot\widehat{\mathbf{r}}^{(1)} \\
&= \mathbf{t} + \mathbf{z} - \mathbf{x}\cdot\widehat{\mathbf{r}}^{(2)} - \mathbf{y}\cdot\widehat{\mathbf{r}}^{(1)} \\
&= \mathbf{t} + \mathbf{x}\cdot\underbrace{\big(\mathbf{r}^{(2)} - \widehat{\mathbf{r}}^{(2)}\big)}_{\Delta\mathbf{r}^{(2)}} + \mathbf{y}\cdot\underbrace{\big(\mathbf{r}^{(1)} - \widehat{\mathbf{r}}^{(1)}\big)}_{\Delta\mathbf{r}^{(1)}} + \mathbf{e} = \mathbf{t} + \underbrace{\mathbf{x}\cdot\Delta\mathbf{r}^{(2)} + \mathbf{y}\cdot\Delta\mathbf{r}^{(1)} + \mathbf{e}}_{\mathbf{z}^*}.
\end{aligned}$$

The better the estimates $\widehat{\mathbf{r}}^{(1)}$ and $\widehat{\mathbf{r}}^{(2)}$, the lower the weights of $\Delta\mathbf{r}^{(1)}$ and $\Delta\mathbf{r}^{(2)}$ and, consequently, the lower the weight of $\mathbf{z}^*$. In particular, if the weights of $\Delta\mathbf{r}^{(1)}$ and $\Delta\mathbf{r}^{(2)}$ are lower than the weights of $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$, then with high probability the noise affecting $\mathbf{t}$ is reduced.

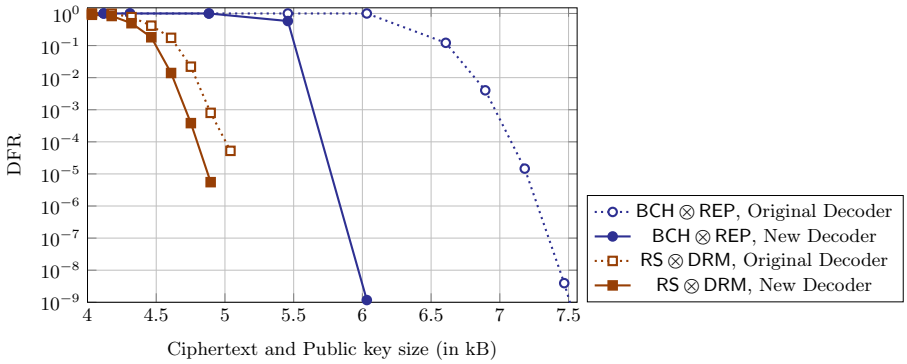


Fig. 1: DFR vs ciphertext plus public key sizes. The DFR has been estimated using numerical simulations by varying the lengths of the component codes. The ciphertext has length $\approx 2n$, while the public key has length $\approx n$.

In this paper, we consider a straightforward decoding strategy, which can be seen as a single iteration of the Bit Flipping (BF) decoder. We show (both theoretically as well as through numerical simulations) that, even with this simple algorithm, one can obtain pretty good estimates $\widehat{\mathbf{r}}^{(1)}$ and $\widehat{\mathbf{r}}^{(2)}$.

Reducing the DFR After the initial filtering step, one can proceed by applying the standard decoder of $\mathscr{C}$ on $\widehat{\mathbf{c}}$. This is already sufficient to improve decoding significantly so that shorter codes can be used to achieve the desired DFR: this reduces both the ciphertext and public key size. In Figure 1 we show an example of the resulting gains, considering the choices for $\mathscr{C}$ which have been used by HQC in the NIST competition: the product of a BCH and a Repetition code (which we indicate as $\mathsf{BCH} \otimes \mathsf{REP}$), and the product of a Reed-Solomon and a Duplicated Reed-Muller code (which we indicate as $\mathsf{RS} \otimes \mathsf{DRM}$).

Limitations and Open Questions This work leaves some open questions. First, many different techniques exist to estimate $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$ as, in principle, any LDPC/MDPC decoder may be used to decode the noisy syndrome. Each of these techniques comes with different trade-offs: defining the best strategy to exploit knowledge of $\mathbf{x}$ and $\mathbf{y}$ is beyond the scope of this paper.

Secondly, and more importantly, one of the key advantages of HQC is the possibility to derive a reliable and closed-form formula for the DFR. While more involved decoders may improve error-correction capability and overall performance, they also come with a more complex DFR analysis. For instance, even for the simple decoder we present in this paper, we have not been able to derive a reliable theoretical model for the DFR. This is due to some technical caveats that arise when iterative decoders (such as the one we propose in this paper) are considered. In our view, this is the most important open question: can we design a decoder with a lower DFR while also allowing for a reliable DFR prediction?

1.2 Paper Organization

The paper is organized as follows. In Section 2, we establish the notation used throughout the paper and recall the main properties of HQC. In Section 3, we provide coding-theoretic arguments that support the possibility of improving performance. Section 4 describes how HQC decoding relates to a noisy syndrome decoding problem and analyzes the use of the BF decoder in such a context. The impact on the DFR of HQC is discussed in Section 5, while Section 6 ends the paper with some concluding remarks.

2 Notation and Background

We use standard notations for finite fields: for q a prime power, $\mathbb{F}_q$ denotes a finite field with q elements while $\mathbb{F}_q[X]$ is the ring of polynomials with coefficients in $\mathbb{F}_q$. For the majority of the paper, we work with the binary finite field $\mathbb{F}_2$. For an integer n, we denote $\mathcal{R} := \mathbb{F}_2[X]/(X^n - 1)$. We frequently view the elements of $\mathcal{R}$ as length-n vectors over $\mathbb{F}_2$, relying on the following canonical representation:

$$\sum_{i=1}^{n} a_i X^{i-1} = \mathbf{a} \in \mathcal{R} \iff (a_1, a_2, \cdots, a_n) = \mathbf{a} \in \mathbb{F}_2^n.$$

For a polynomial $\mathbf{a} \in \mathcal{R}$, we indicate by $\mathrm{wt}(\mathbf{a})$ its Hamming weight, i.e., the number of non-zero coefficients. By support of a polynomial, which we indicate as $\mathrm{Supp}(\mathbf{a})$, we refer to the set of indices i for which $a_i \neq 0$; notice that $\mathrm{wt}(\mathbf{a}) = |\mathrm{Supp}(\mathbf{a})|$. Given two polynomials $\mathbf{a}$ and $\mathbf{b}$, we indicate the j-th coefficient of their product as $(\mathbf{a} \cdot \mathbf{b})_j$. For $w \in \mathbb{N}$, $w \leqslant n$, we define $\mathcal{R}_w = \{\mathbf{a} \in \mathcal{R} \mid \mathrm{wt}(\mathbf{a}) = w\}$.

Probability Distributions Given a set A, we write $a \xleftarrow{\$} A$ when a is drawn uniformly at random from A. We use $\mathcal{B}_{n,\rho}$ to indicate the Bernoulli distribution over $\mathbb{F}_2^n$ with parameter ρ, i.e., the distribution that returns vectors of length n and such that any entry is 1 with probability ρ and 0 with probability $1 - \rho$. If $\mathbf{a} \in \mathbb{F}_2^n$ is distributed according to $\mathcal{B}_{n,\rho}$, we write $\mathbf{a} \sim \mathcal{B}_{n,\rho}$.

2.1 Background on Coding Theory

A linear code $\mathscr{C} \subseteq \mathbb{F}_q^n$ with length n and dimension k is a linear subspace of $\mathbb{F}_q^n$ containing q^k vectors, called *codewords*. A compact representation for a code is a *generator matrix*, that is, a matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ with rank k, whose row span yields $\mathscr{C}$, that is, $\mathscr{C} = \left\{\mathbf{mG} \mid \mathbf{m} \in \mathbb{F}_q^k\right\}$. The minimum distance of a code is the minimum Hamming distance between two distinct codewords and, for linear codes, corresponds to the minimum weight of a non-zero codeword.

We say $\mathscr{C}$ is an *error-correcting code* whenever it can be equipped with an efficient algorithm Dec that, on input some $\mathbf{c} = \mathbf{mG} + \mathbf{e}$, returns $\mathbf{mG}$ with high probability when $\mathbf{e}$ has sufficiently low weight. When the decoder does not return

$\mathbf{mG}$, we encounter a *decoding failure*.[3] The probability (over the channel's and decoder's randomness) that a decoding failure event occurs is called the Decoding Failure Rate (DFR).

Tensor Product Codes Let q_1 and q_2 be two (possibly equal) prime powers. Given two codes $\mathscr{C}_1 \subseteq \mathbb{F}_{q_1}^{n_1}$ and $\mathscr{C}_2 \subseteq \mathbb{F}_{q_2}^{n_2}$ with dimensions k_1 and k_2, we denote by $\mathscr{C} = \mathscr{C}_1 \otimes \mathscr{C}_2$ their *tensor product*. Sometimes, $\mathscr{C}_1$ is called *outer code* and $\mathscr{C}_2$ *inner code*. Let $\mathsf{Enc}_1 : \mathbb{F}_{q_1}^{k_1} \mapsto \mathscr{C}_1$ and $\mathsf{Enc}_2 : \mathbb{F}_{q_2}^{k_2} \mapsto \mathscr{C}_2$ be the encoding functions for $\mathscr{C}_1$ and $\mathscr{C}_2$. Analogously, we denote by Dec_1 and Dec_2 their decoding algorithms. Finally, we denote by $\mathsf{Enc}_{\mathbb{F}_{q_1},\mathbb{F}_{q_2}}$ a function that, on input a vector in $\mathbb{F}_{q_1}^{n_1}$, returns a sequence with values over $\mathbb{F}_{q_2}$ and length n_1'; we require that n_1' is a multiple of k_2. [4] The inverse function is indicated as $\mathsf{Dec}_{\mathbb{F}_{q_1},\mathbb{F}_{q_2}}$. If the two finite fields are the same, the function $\mathsf{Enc}_{\mathbb{F}_{q_1},\mathbb{F}_{q_2}}$ is simply the identity (and so is its inverse). Thus, we also have $n_1' = n_1$ hence k_2 must divide n_1.

We now show how encoding of a message $\mathbf{m} \in \mathbb{F}_q^{k_1}$ works. First, $\mathbf{m}$ is encoded into a codeword of $\mathscr{C}_1$. Then, we apply $\mathsf{Enc}_{\mathbb{F}_{q_1},\mathbb{F}_{q_2}}$ to obtain a a string of length n_1' and values over $\mathbb{F}_{q_2}$. This is divided into n_1'/k_2 chunks: each has length k_2 and is encoded into a codeword of $\mathscr{C}_2$. The resulting codeword has length $n_1' n_2/k_2$. See Figure 2 for a graphical representation of the encoding procedure.

Decoding a tensor product code is done by first decoding the $\frac{n_1'}{k_2}$ codewords of $\mathscr{C}_2$ and then decoding their concatenation through $\mathscr{C}_1$. Namely, let $\mathbf{c} = \left(\mathbf{c}^{(1)} || \cdots || \mathbf{c}^{(n_1'/k_2)}\right)$ be a received word, with each $\mathbf{c}^{(i)}$ having length n_2 (here, we use $||$ to indicate the concatenation of two row vectors). Then, decoding of $\mathbf{c}$ is done by first decoding each $\mathbf{c}^{(i)}$ and mapping the result back to $\mathbb{F}_{q_1}$ through $\mathsf{Dec}_{\mathbb{F}_{q_1},\mathbb{F}_{q_2}}$; all these words are concatenated and the obtained word is decoded through $\mathscr{C}_1$. Compactly, this process is described as

$$\mathsf{Dec}_1\left(\mathsf{Dec}_{\mathbb{F}_{q_1},\mathbb{F}_{q_2}}\left(\mathsf{Dec}_2\left(\mathbf{c}^{(1)}\right)\right) \Big|\Big| \cdots \Big|\Big| \mathsf{Dec}_{\mathbb{F}_{q_1},\mathbb{F}_{q_2}}\left(\mathsf{Dec}_2\left(\mathbf{c}^{(n_1'/k_2)}\right)\right)\right).$$

2.2 HQC in a Nutshell

Figure 3 summarizes the main operations in HQC. Given the scope of this paper, we omit all the operations that are not relevant from a coding-theory perspective (but, obviously, are relevant from a cryptographic perspective!), such as the use of seeds and the IND-CCA2 conversion; for full details, we refer the interested reader to [9].

In our description, the encoding and decoding processes of the public code $\mathscr{C}$ are indicated as Enc and Dec, respectively. The outer code is generically defined over a finite field of size q, while the inner code is always binary.

[3] Generically, decoding failures happen when the decoder either halts on a vector that is not a codeword or on a codeword different from the transmitted one.

[4] For instance, if $q_1 = 2^\ell$ and $q_2 = 2$, a codeword of $\mathscr{C}_1$ can be represented as a binary string of length $\ell \cdot n_1$.

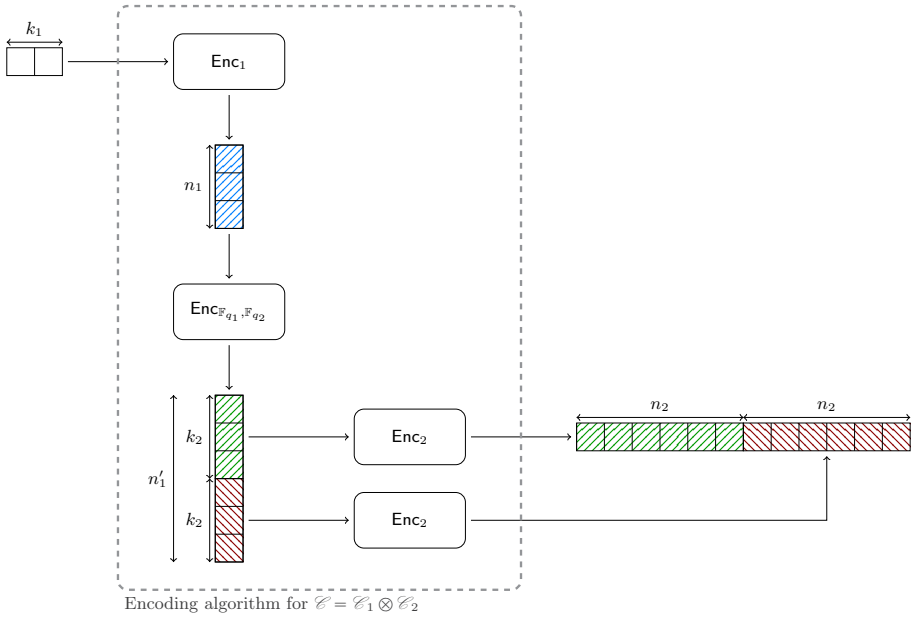


Fig. 2: Example of the encoding procedure for a tensor code obtained from two component codes defined over two fields $\mathbb{F}_{q_1}$ and $\mathbb{F}_{q_2}$ such that $q_1 = q_2^2$. The function $\mathsf{Enc}_{\mathbb{F}_{q_1},\mathbb{F}_{q_2}}$ maps a symbol from $\mathbb{F}_{q_1}$ into two symbols of $\mathbb{F}_{q_2}$. Code parameters are $k_1 = 2$, $n_1 = 3$, $n_1' = 2 \cdot n_1 = 6$, $k_2 = 3$ and $n_2 = 6$. The resulting code has dimension $k = k_1 = 2$ and $n = \frac{n_1'}{k_2} \cdot n_2 = 12$.

Due to the special structure of the ciphertext and public key, one has $\mathbf{c} = \mathbf{t} + \mathbf{z}$, where $\mathbf{t} \in \mathscr{C}$ and

$$\mathbf{z} = \mathbf{x} \cdot \mathbf{r}^{(2)} + \mathbf{y} \cdot \mathbf{r}^{(1)} + \mathbf{e}. \tag{2}$$

Since all the polynomials in the above equation are sparse, $\mathbf{z}$ has low weight. This allows for efficiently correcting errors through the decoding algorithm Dec.

Choice of $\mathscr{C}$ The practical performance of the scheme heavily depends on the choice of $\mathscr{C}$. Indeed, one should choose a family of error-correcting codes that yields a good trade-off between error-correction capability, compactness, and computational efficiency. First, notice that both the public key size and the ciphertext size are linear in n.[5] Hence, one would desire n to be as small as possible; however, this value cannot be too small as we must guarantee that, with sufficiently large probability, Dec corrects (efficiently) the errors in $\mathbf{z}$.

[5] The public key size is $n + \lambda$ bits, with λ denoting the security level, because $\mathbf{h}$ can be compressed by the seed with which it has been generated. The ciphertext size, instead, corresponds to $2n + 2\lambda$. The term 2λ is the overhead due to the IND-CCA2 conversion.

Key generation:
1. sample $\mathbf{h} \xleftarrow{\$} \mathcal{R}$;
2. sample $\mathbf{x}, \mathbf{y} \xleftarrow{\$} \mathcal{R}_w \times \mathcal{R}_w$;
3. set $\mathsf{sk} := (\mathbf{x}, \mathbf{y})$ and $\mathsf{pk} := \{\mathbf{h}\ ,\ \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y}\}$.

Encryption: on input $\mathbf{m} \in \mathbb{F}_q^k$ and $\mathsf{pk} := \{\mathbf{h}, \mathbf{s}\}$, do:
1. sample $\mathbf{r}^{(1)}, \mathbf{r}^{(2)} \xleftarrow{\$} \mathcal{R}_{w_r} \times \mathcal{R}_{w_r}$, $\mathbf{e} \xleftarrow{\$} \mathcal{R}_{w_e}$;
2. set $\mathbf{u} = \mathbf{r}^{(1)} + \mathbf{h} \cdot \mathbf{r}^{(2)}$;
3. compute $\mathbf{t} = \mathsf{Enc}(\mathbf{m})$;
4. set $\mathbf{v} = \mathbf{t} + \mathbf{e} + \mathbf{s} \cdot \mathbf{r}^{(2)}$;
5. the ciphertext is $\{\mathbf{u}, \mathbf{v}\}$.

Decryption: on input $(\mathbf{u}, \mathbf{v}) \in \mathcal{R}^2$ and $\mathsf{sk} := (\mathbf{x}, \mathbf{y})$, do:
1. compute $\mathbf{c} = \mathbf{v} + \mathbf{y} \cdot \mathbf{u}$;
2. run $\mathsf{Dec}(\mathbf{c})$.

Fig. 3: Description of HQC.

To guarantee IND-CCA2 security, the DFR must be negligible in the security parameter, i.e., less than $2^{-\lambda}$. This leads to another requirement: devising a theoretical, closed-form formula for the DFR must be feasible. For this reason, the decoding process cannot be too involved: as it is well known, devising reliable models for the DFR is extremely complicated, and only a few decoding algorithms (namely, the simplest ones) allow for it.

Taking this into account, the authors of HQC chose to use tensor product codes, as they allow for a simple DFR modeling and, at the same time, offer a good error-correction capability. While the design of HQC has essentially remained the same throughout the years, its parameters and the choice of $\mathscr{C}$ have changed. Two choices for $\mathscr{C}$ have been considered so far:

- Version $\mathsf{BCH} \otimes \mathsf{Rep}$ from first round:
  - Outer code: BCH code defined over $\mathbb{F}_2$
  - Inner code: Repetition code defined over $\mathbb{F}_2$
- Version $\mathsf{RS} \otimes \mathsf{DRM}$ from second round:
  - Outer code: Reed-Solomon code defined over $\mathbb{F}_{2^8}$
  - Inner code: Duplicated Reed-Muller code defined over $\mathbb{F}_2$

3 Coding-Theoretic Analysis

This section highlights from a coding-theoretic perspective the improvements that are possible due to the knowledge of the error structure at the decoder (in particular due to knowing $\mathbf{x}, \mathbf{y}$). To this end, we first revisit decoding agnostic to the secret key and its limitations. Then, we derive a suitable generalization of the Gilbert-Varshamov (GV) bound tailored to the particular setting we encounter in HQC. For simplicity, throughout this section, we assume that the used code is of the same length as the vectors in $\mathcal{R}$.

3.1 Approximating HQC Errors via a Binary Symmetric Channel

The HQC error being of the form $\mathbf{z} = \mathbf{x} \cdot \mathbf{r}^{(2)} + \mathbf{y} \cdot \mathbf{r}^{(1)} + \mathbf{e}$ implies that

$$\mathrm{wt}(\mathbf{z}) \leqslant 2 \cdot w_r \cdot w + w_e := w_{\max}.$$

Besides the maximum weight of $\mathbf{z}$, the probability that a single coordinate of $\mathbf{z}$ is non-zero over the choice of $\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \mathbf{e}$ can be computed as

$$\Pr[z_i = 1] = 2\rho'_{w,w_r}(1-\rho_{w,w_r})\left(1-\frac{w_e}{n}\right) + \left((1-\rho_{w,w_r})^2 + \rho^2_{w,w_r}\right)\frac{w_e}{n} := \rho_z$$

where $\rho_{w,w_r} = \binom{n}{w}^{-1} \sum_{\ell \text{ odd}} \binom{w_r}{\ell}\binom{n-w_r}{w-\ell}$ [3]. For completeness, the proof is included in Appendix A for completeness. Deriving further statements beyond the maximum weight and the error probability of a single coordinate is involved. Therefore, the following heuristic is used to derive an analytical expression for the DFR.

Heuristic 1 (Binomial Approximation [3]) *Under the simplifying assumption that the coordinates of* $\mathbf{z}$ *are independent random variables, the probability of* $\mathbf{z}$ *is modeled as*

$$\Pr[\mathbf{z}] = \begin{cases} N \cdot \rho_z^{\mathrm{wt}(\mathbf{z})} \cdot (1-\rho_z)^{n-\mathrm{wt}(\mathbf{z})} & \text{if } \mathrm{wt}(\mathbf{z}) \leqslant \min\{w_{\max}, n\}, \\ 0 & \text{else,} \end{cases}$$

with normalization factor $N = \Pr[\mathrm{wt}(z) > w_{\max} \mid \mathcal{B}_{n,\rho_z}]^{-1}$. *In particular, the weight* $\mathrm{wt}(\mathbf{z})$ *follows a truncated binomial distribution.*

Heuristic 1 implies that HQC can operate without DFR only at the cost of a severe rate restriction. The following lemma emphasizes this.

Lemma 1 (DFR = 0 under Heuristic 1).
Under Heuristic 1, any code-decoder pair that can guarantee correct decryption satisfies a code size of

$$|\mathscr{C}| \leqslant \frac{2^n}{\sum_{i=0}^{w_{\max}} \binom{n}{i}}.$$

Proof. To guarantee correct decoding under Heuristic 1, $\mathscr{C}$ needs to be able to correct all patterns of weight at most $w_{\max}$. The classical Hamming bound implies $|\mathscr{C}| \leqslant 2^n / \sum_{i=0}^{w_{\max}} \binom{n}{i}$. □

Note that the cryptographic setting determines the code dimension of the public code; for example, in the round-4 version of HQC, $k = \lambda$ is picked. In this setting, Lemma 1 can be understood as a lower bound on the required code length, which, in turn, determines the sizes of the public key and ciphertext.

In practice, reduced sizes can be achieved by tolerating a non-zero DFR. IND-CCA2 security can be guaranteed for a DFR of at most $2^{-\lambda}$. The following theorem, inspired by [7], gives a lower bound on the DFR for a given code size and dimension. Note that a related bound has been developed in [1].

Lemma 2 (Elias sphere-packing bound under Heuristic 1). *Let $\mathscr{C}$ be an arbitrary linear code of length n and dimension k. Denote as ρ_z the error probability of the BSC as in Heuristic 1. Let $t \in [n]$ be minimal such that $\sum_{i=0}^{t} \binom{n}{i} > 2^{n-k}$. Then, any decoder for $\mathscr{C}$ encounters at least a DFR of*

$$\sum_{i=t+1}^{w_{\max}} \binom{n}{i} \rho_z^i (1-\rho_z)^{n-i}.$$

Proof. Denote as $\mathcal{D}_{\mathscr{C}} \subset \mathbb{F}_2^n$ the decoding region of the linear code $\mathscr{C}$, i.e., $\mathbf{t} = \mathsf{Enc}(\mathbf{t} + \mathbf{z})$ for all $\mathbf{z} \in \mathcal{D}_{\mathscr{C}}$ and all $\mathbf{t} \in \mathscr{C}$. Note that $|\mathcal{D}_{\mathscr{C}}| = 2^{n-k}$. Then, since $\Pr[\mathbf{z}_1] > \Pr[\mathbf{z}_2]$ for $\mathrm{wt}(\mathbf{z}_1) < \mathrm{wt}(\mathbf{z}_2)$,

$$\epsilon = \Pr[\mathbf{z} \notin \mathcal{D}_{\mathscr{C}}] \geqslant \Pr[\mathbf{z} \notin \mathcal{B}_{\mathscr{C}}],$$

where $\mathcal{B}_{\mathscr{C}}$ is a Hamming ball with $|\mathcal{B}_{\mathscr{C}}| \geqslant |\mathcal{D}_{\mathscr{C}}|$. □

Again, by fixing $k = \lambda$ and a sufficiently low DFR, Lemma 2 can be understood as a lower bound on the required code length. The following example illustrates how the bounds compare with the code construction used in HQC.

Example 1. The concatenation of the Reed-Muller code and the Reed-Solomon code employed in round 4 by HQC uses $n = 17996$ for a security level of $\lambda = 128$ bit. The code dimension is $k = \lambda$, and the achieved DFR is at most $2^{-\lambda}$. Lemma 2 implies that the minimum code length for achieving the required DFR is at least $n = 13438$. Achieving guaranteed correctness of the decoding would require at least $n = 21822$ according to Lemma 1.

Example 1 shows that the current HQC parameters are already close to optimal, assuming that HQC errors follow Heuristic 1. This does not hold once we replace the binomial approximation with the actual structure of the HQC errors, as we will see in the following.

3.2 Beyond the BSC: Leveraging the Error Structure, x, and y

Let us denote the set of all possible error patterns given $\mathbf{x}, \mathbf{y}$ as

$$\mathcal{E}_{\mathbf{x},\mathbf{y}} = \left\{ \mathbf{z} = \mathbf{x} \cdot \mathbf{r}^{(2)} + \mathbf{y} \cdot \mathbf{r}^{(1)} + \mathbf{e} \mid \mathrm{wt}(\mathbf{r}^{(1)}) = w_r, \mathrm{wt}(\mathbf{r}^{(2)}) = w_r, \mathrm{wt}(\mathbf{e}) = w_r \right\}.$$

Then, the set of all possible error patterns for arbitrary $\mathbf{x}, \mathbf{y}$ is given by

$$\mathcal{E} = \bigcup_{\mathbf{x},\mathbf{y}:\ \mathrm{wt}(\mathbf{x})=\mathrm{wt}(\mathbf{y})=w} \mathcal{E}_{\mathbf{x},\mathbf{y}}.$$

There is a well-known condition for characterizing whether a code can uniquely correct all patterns in $\mathcal{E}$; see, e.g., [8].

Lemma 3 (Generalized GV bound).
A linear code $\mathscr{C}$ can correct all patterns in $\mathcal{E}$ uniquely if and only if $\mathscr{C} \cap \Delta\mathcal{E} = \{0\}$, where $\Delta\mathcal{E}$ denotes the difference set of $\mathcal{E}$, i.e., $\Delta\mathcal{E} \coloneqq \{\mathbf{z} - \mathbf{z}' \mid \mathbf{z}, \mathbf{z}' \in \mathcal{E}\}$. As a consequence, there exists a code of cardinality

$$|\mathscr{C}| \leqslant \frac{2^n}{|\Delta\mathcal{E}|}$$

that can correct all errors in $\mathcal{E}$.

Note that Lemma 3 generalizes the standard Hamming-metric Gilbert-Varshamov bound to arbitrary error patterns. We obtain the following lemma for our particular set consisting of HQC errors.

Lemma 4 (GV-like bound with error structure).
Let $\mathcal{E}$ be the set containing all possible HQC error patterns generated with parameters w and w_r. Then, $|\Delta\mathcal{E}| \leqslant \binom{n}{w}^4 \binom{n}{w_r}^6$ which implies that there exists a code $\mathscr{C}$ of dimension

$$k \geqslant n - 4w \log_2(\tfrac{n \cdot e}{w}) - 6w_r \log_2(\tfrac{n \cdot e}{w_r})$$

that can correct arbitrary HQC errors, i.e., guarantee correct decryption.

Proof. $\Delta\mathcal{E}$ contains vectors of the form

$$\mathbf{x} \cdot \mathbf{r}^{(2)} + \mathbf{y} \cdot \mathbf{r}^{(1)} + \mathbf{e} + \mathbf{x}' \cdot \mathbf{r}^{(2)'} + \mathbf{y}' \cdot \mathbf{r}^{(1)'} + \mathbf{e}'$$

with $\mathrm{wt}(\mathbf{x}) = \mathrm{wt}(\mathbf{x}') = \mathrm{wt}(\mathbf{y}) = \mathrm{wt}(\mathbf{y}') = w$, $\mathrm{wt}(\mathbf{r}^{(1)}) = \mathrm{wt}(\mathbf{r}^{(1)'}) = \mathrm{wt}(\mathbf{r}^{(2)}) = \mathrm{wt}(\mathbf{r}^{(2)'})\,\mathrm{wt}(\mathbf{e}) = \mathrm{wt}(\mathbf{e}') = w_r$. The number of elements is upper-bounded by the number of choices for the components. This argument implies

$$|\mathcal{E}| \leqslant \binom{n}{w}^4 \binom{n}{w_r}^6.$$

Using the relation $\binom{a}{b} \leqslant \left(\frac{n \cdot e}{k}\right)^k$, the result follows via Lemma 3. □

Lemma 4 considers the particular structure of HQC errors but neglects that the decoder can use the knowledge of $\mathbf{x}, \mathbf{y}$. In the following, we analyze the impact of this additional information. To this end, let us first consider the case that the code $\mathscr{C}$ is only supposed to work for a particular choice of $\mathbf{x}, \mathbf{y}$. Then, $\mathscr{C} \cap \Delta\mathcal{E}_{\mathbf{x},\mathbf{y}} = \{\mathbf{0}\}$ is sufficient. For HQC, the public code is fixed and independent of the choice of the private key. Hence, the instantiation of $\mathscr{C}$ is supposed to work for all possible choices of $\mathbf{x}, \mathbf{y}$. Nevertheless, as the following lemma shows, knowing $\mathbf{x}, \mathbf{y}$ is valuable information for the decoder.

Lemma 5. *A code $\mathscr{C}$ with a decoder knowing $\mathbf{x}, \mathbf{y}$ can correct arbitrary HQC errors if and only if $\mathscr{C} \cap \Delta\mathcal{E}^* = \{\mathbf{0}\}$, where*

$$\Delta\mathcal{E}^* \coloneqq \bigcup_{\mathbf{x},\mathbf{y}:\mathrm{wt}(\mathbf{x})=\mathrm{wt}(\mathbf{y})=w} \Delta\mathcal{E}_{\mathbf{x},\mathbf{y}}.$$

Proof. We require that $\mathscr{C} \cap \Delta\mathcal{E}_{\mathbf{x},\mathbf{y}} = \{\mathbf{0}\}$ for all $\mathbf{x}$ and $\mathbf{y}$. This is equivalent to

$$\bigcup_{\mathbf{x},\mathbf{y}:\mathrm{wt}(\mathbf{x})=\mathrm{wt}(\mathbf{y})=w} (\mathscr{C} \cap \Delta\mathcal{E}_{\mathbf{x},\mathbf{y}}) = \mathscr{C} \cap \bigcup_{\mathbf{x},\mathbf{y}:\mathrm{wt}(\mathbf{x})=\mathrm{wt}(\mathbf{y})=w} \Delta\mathcal{E}_{\mathbf{x},\mathbf{y}} = \{0\}. \quad \square$$

The restriction on $\mathscr{C}$ due to Lemma 5 is less strict than the restriction implied by ignoring $\mathbf{x}, \mathbf{y}$, i.e., $\Delta\mathcal{E}^* \subseteq \Delta\mathcal{E}$. To see this, observe that $\Delta\mathcal{E}^*$ contains all vectors of the shape

$$\mathbf{r}^{(2)'} \cdot \mathbf{x} + \mathbf{r}^{(1)'} \cdot \mathbf{y} + \mathbf{e}$$

with $\mathrm{wt}(\mathbf{x}) = \mathrm{wt}(\mathbf{y}) = w$ and $\mathrm{wt}(\mathbf{r}^{(1)'}) = \mathrm{wt}(\mathbf{r}^{(2)'}) = \mathrm{wt}(\mathbf{e}') \in \{0, 2, \ldots, 2w_r\}$. This corresponds to the special case of $\mathbf{x} = \mathbf{x}'$ and $\mathbf{y} = \mathbf{y}'$ for the elements of $\Delta\mathcal{E}$ which are of the form

$$\mathbf{r}^{(2)} \cdot \mathbf{x} + \mathbf{r}^{(1)} \cdot \mathbf{y} + \mathbf{e} + \mathbf{r}^{(2)'} \cdot \mathbf{x}' + \mathbf{r}^{(1)'} \cdot \mathbf{y}' + \mathbf{e}'.$$

Theorem 1 (GV-like bound with error structure and $\mathbf{x}, \mathbf{y}$). *Let the decoder know $\mathbf{x}, \mathbf{y}$. Then, there exists a code $\mathscr{C} \subset \mathbb{F}_2^n$ of dimension*

$$k \geqslant n - 2w \log_2(\tfrac{n \cdot e}{w}) - 6w_r \log_2(\tfrac{n \cdot e}{2w_r}) - \log_2(w_r)$$

that can correct arbitrary HQC errors, i.e., guarantee correct decryption.

Proof. According to Lemma 5, we require $\mathscr{C} \cap \Delta\mathcal{E}^* = \{\mathbf{0}\}$. To bound the number of elements in $\Delta\mathcal{E}^*$, observe that

$$\begin{aligned} |\Delta\mathcal{E}^*| = \left| \bigcup_{\mathbf{x},\mathbf{y}:\mathrm{wt}(\mathbf{x})=\mathrm{wt}(\mathbf{y})=w} \Delta\mathcal{E}_{\mathbf{x},\mathbf{y}} \right| &\leqslant \binom{n}{w}^2 |\Delta\mathcal{E}_{\mathbf{x},\mathbf{y}}| \leqslant \binom{n}{w}^2 \left(\sum_{i=0}^{2w_r} \binom{n}{i} \right)^3 \\ &\leqslant \left(\frac{n \cdot e}{w}\right)^{2w} w_r \left(\frac{n \cdot e}{2w_r}\right)^{6w_r}, \end{aligned}$$

where we have used $\binom{a}{b} \leqslant \left(\frac{a \cdot e}{b}\right)^b$. The statement follows due to Lemma 3. $\square$

The approach to estimating the sizes of $\Delta\mathcal{E}$ and $\Delta\mathcal{E}^*$ in Lemma 4 and in Theorem 1 is imprecise in general since it ignores the possibility of collisions. Therefore, we expect that these bounds can be tightened.

Nevertheless, Theorem 1 already shows the potential of considering the error structure. This becomes clear from Table 1, which gives an overview of the bounds described in this section. The used error model, the implied decoding algorithm, and the resulting DFR are compared. Further, we compare the lower bounds on minimum required lengths under Heuristic 1 with the code lengths that are achievable by considering the error structure. For simplicity, w, w_r, and w_e were fixed to the choice of HQC in round 4. Note that higher n would require increasing these parameters, while a smaller code length might allow reducing them. Even when this effect is disregarded, it can already be observed that shorter code lengths are enabled by incorporating the error structure. This observation becomes more pronounced for higher security levels but is already significant for $\lambda = 128$. The following example elaborates on this finding.

Table 1: Overview of the bounds described in this section and comparison with the current HQC instantiation of round 4. Required and achievable code lengths for code dimension $k = \lambda$.

	HQC [9]	Lem. 1	Lem. 2	Lem. 4	Thm. 1
error model	BSC-like	BSC-like	BSC-like	structured	structured
decoder	multistage	ML	unique	unique	unique + $\mathbf{x}, \mathbf{y}$
DFR	$2^{-\lambda}$	$2^{-\lambda}$	0	0	0
	used n	LB on required n		UB on achievable n	
NIST 1	17 669	13 438	21 882	5417	3800
NIST 3	35 851	27 913	49 403	8243	5782
NIST 5	57 637	45 150	83 767	10 804	7576

Example 2. Again, we consider the HQC parameters of round 4 that achieve NIST-I security, i.e., $\lambda \approx 128$ bit. That is, $n = 17669$, $w_r = 75$, $w = 66$ and the code concatenation has a dimension of $k = 128$. However, the current code-decoder design does not factor in the particular structure of the error vectors. In doing so, Lemma 4 guarantees that a code of length $n \leqslant 5417$ exists that can guarantee correct decoding. By taking $\mathbf{x}$ and $\mathbf{y}$ into account, Theorem 1 can reduce the upper bound on the required length to $n \leqslant 3800$.

The described improvement would increase the bandwidth efficiency of HQC considerably. Note, however, that the result is non-constructive: The proposed GV-like bounds do not provide an explicit code construction. Further, whether a computationally efficient decoder exists that accounts for the error structure remains open. We approach this second question in the following section.

4 Exploiting x and y to Reconstruct $\mathbf{r^{(1)}}$ and $\mathbf{r^{(2)}}$

In this section, we describe how, exploiting knowledge of $\mathbf{x}$ and $\mathbf{y}$, one can estimate coefficients of $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$. We first present the general idea, then analyze it for the case of $\mathscr{C}_2$ being a repetition code because this makes the analysis easier. Indeed, in this case, we can get a rather tight theoretical estimate for the probability distribution of the number of correctly guessed coefficients.

4.1 Decoding as a Noisy Syndrome Decoding Problem

The idea is that, on input $\mathbf{c} = \mathbf{t} + \mathbf{z}$, with $\mathbf{t} \in \mathscr{C}$, one can use decoding through the inner code $\mathscr{C}_2$ to get an estimate of the error term $\mathbf{z}$. From this estimate, one can then guess coefficients of $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$. These guesses are later used to remove some of the noise in $\mathbf{c}$. Indeed, let $\widehat{\mathbf{r}}^{(1)} \in \mathcal{R}$ and $\widehat{\mathbf{r}}^{(2)} \in \mathcal{R}$ denote the

estimates for $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$, respectively. Then, one can update $\mathbf{c}$ as

$$\begin{aligned}\widehat{\mathbf{c}} &= \mathbf{c} - \mathbf{x}\cdot\widehat{\mathbf{r}}^{(2)} - \mathbf{y}\cdot\widehat{\mathbf{r}}^{(1)} \\ &= \mathbf{t} + \mathbf{x}\cdot\widehat{\mathbf{r}}^{(2)} - \mathbf{y}\cdot\widehat{\mathbf{r}}^{(1)} \\ &= \mathbf{t} + \underbrace{\mathbf{x}\cdot\underbrace{\left(\mathbf{r}^{(2)} - \widehat{\mathbf{r}}^{(2)}\right)}_{\Delta\mathbf{r}^{(2)}} + \mathbf{y}\cdot\underbrace{\left(\mathbf{r}^{(1)} - \widehat{\mathbf{r}}^{(1)}\right)}_{\Delta\mathbf{r}^{(1)}} + \mathbf{e}}_{\mathbf{z}^*} = \mathbf{t} + \mathbf{z}^*.\end{aligned}$$

Notice that, if the estimates $\widehat{\mathbf{r}}^{(1)}$ and $\widehat{\mathbf{r}}^{(2)}$ are accurate, then $\Delta\mathbf{r}^{(1)}$ and $\Delta\mathbf{r}^{(2)}$ will have a low Hamming weight, perhaps considerably lower than w_r (remember that w_r is the weight of $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$). If this happens, then the weight of $\mathbf{z}^*$ is with a high probability smaller than that of $\mathbf{z}$: Consequently, the decoding of $\widehat{\mathbf{c}}$ is more probable to succeed than the decoding of the original $\mathbf{c}$.

Estimating $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$ Let us write $\mathbf{t} = \left(\mathbf{t}^{(1)}||\mathbf{t}^{(2)}||\cdots||\mathbf{t}^{(n/n_2)}\right)$ with each $\mathbf{t}^{(i)}$ being a codeword of $\mathscr{C}_2$, thus, of length n_2. Analogously, we write $\mathbf{z} = \left(\mathbf{z}^{(1)}||\mathbf{z}^{(2)}||\cdots||\mathbf{z}^{(n/n_2)}\right)$ where, again, each $\mathbf{z}^{(i)}$ has length n_2. Then, we can write $\mathbf{c} = \left(\mathbf{c}^{(1)}||\mathbf{c}^{(2)}||\cdots||\mathbf{c}^{(n/n_2)}\right)$, with $\mathbf{c}^{(i)} = \mathbf{t}^{(i)} + \mathbf{z}^{(i)}$.

Notice that since $\mathbf{z}$ has low weight, each $\mathbf{z}^{(i)}$ has low weight with high probability. Thus, with high probability decoding of $\mathbf{c}^{(i)}$ returns $\mathbf{t}^{(i)}$. Even when decoding fails, the output of the decoder is expected to be a codeword close to $\mathbf{t}^{(i)}$ (since the weight of $\mathbf{z}^{(i)}$ is generically low). We resume this reasoning by indicating $\widetilde{\mathbf{t}}^{(i)} = \mathsf{Dec}\left(\mathbf{c}^{(i)}\right)$ where, again, $\widetilde{\mathbf{t}}^{(i)} = \mathbf{t}^{(i)}$ holds with large probability. Then, we get a presumably good estimate of $\mathbf{z}$ by computing

$$\begin{aligned}\widehat{\mathbf{z}} &= \mathbf{c} - \left(\widetilde{\mathbf{t}}^{(1)}||\widetilde{\mathbf{t}}^{(2)}||\cdots||\widetilde{\mathbf{t}}^{(n/n_2)}\right) \\ &= \left(\mathbf{t}^{(1)}||\mathbf{t}^{(2)}||\cdots||\mathbf{t}^{(n/n_2)}\right) - \left(\widetilde{\mathbf{t}}^{(1)}||\widetilde{\mathbf{t}}^{(2)}||\cdots||\widetilde{\mathbf{t}}^{(n/n_2)}\right) + \mathbf{z} = \Delta\mathbf{t} + \mathbf{z}.\end{aligned}$$

Remember that $\mathbf{z} = \mathbf{x}\cdot\mathbf{r}^{(2)} + \mathbf{y}\cdot\mathbf{r}^{(1)} + \mathbf{e}$, hence

$$\widehat{\mathbf{z}} = \mathbf{x}\cdot\mathbf{r}^{(2)} + \mathbf{y}\cdot\mathbf{r}^{(1)} + \mathbf{e} + \Delta\mathbf{t}.$$

Since $\mathbf{e}$ has low weight by design and $\Delta\mathbf{t}$ is expected to have moderately low weight because of the above reasoning, the vector $\widehat{\mathbf{z}}$ can be seen as a noisy version of the vector $\left(\mathbf{x}\cdot\mathbf{r}^{(2)}||\mathbf{y}\cdot\mathbf{r}^{(1)}\right)$.

Let $\mathbf{r}\in\mathbb{F}_2^{2n}$ be the vector formed by the coefficients of $\mathbf{r}^{(2)}$ followed by those of $\mathbf{r}^{(1)}$. Moreover, let $\mathbf{H} = \left(\mathrm{Rot}(\mathbf{x})||\mathrm{Rot}(\mathbf{y})\right)$, where $\mathrm{Rot}(\mathbf{x})$ and $\mathrm{Rot}(\mathbf{y})$ are the circulant matrices whose first rows are $\mathbf{x}$ and $\mathbf{y}$, respectively. Then, we have

$$\widehat{\mathbf{z}} = \mathbf{H}\mathbf{r}^\top + \Delta\mathbf{z}.$$

The problem can be formulated as follows: on input a sparse parity-check matrix $\mathbf{H}\in\mathbb{F}_2^{n\times 2n}$ and a noisy syndrome $\widehat{\mathbf{z}}$, find a vector $\mathbf{r}\in\mathbb{F}_2^{2n}$ with weight $2w_r$ such that $\mathrm{wt}(\widehat{\mathbf{z}} - \mathbf{H}\mathbf{r}^\top)$ is minimum. This is precisely a Noisy Syndrome Decoding instance [6].

Input: noisy codeword $\mathbf{c} \in \mathbb{F}_2^n$, threshold $T \in \mathbb{N}$, secret key $(\mathbf{x}, \mathbf{y}) \in \mathcal{R}_w^2$, parity-check matrix $\mathbf{H} = \big(\mathrm{Rot}(\mathbf{x})||\mathrm{Rot}(\mathbf{y})\big) \in \mathbb{F}_2^{n\times 2n}$

Output: noisy codeword $\widehat{\mathbf{c}} \in \mathbb{F}_2^n$

/* Estimate error vector by decoding $\mathscr{C}_2$ */
1) compute $\mathbf{t} = \mathsf{Dec}_2(\mathbf{c})$
2) set $\widehat{\mathbf{z}} = \mathbf{c} - \mathbf{t}$

/* Guess coefficients of $\mathbf{r}^{(2)}$ */
3) Set $\widehat{\mathbf{r}}^{(2)} = (0, \cdots, 0)$;
4) for $i = 1, \cdots, n$, do:
5) set $b_i = \widehat{\mathbf{z}} \star \mathbf{h}_i$;// i-th counter
6) if $b_i \geqslant T$, set $\widehat{r}_i^{(2)} = 1$;

/* Guess coefficients of $\mathbf{r}^{(1)}$ */
7) Set $\widehat{\mathbf{r}}^{(1)} = (0, \cdots, 0)$
8) for $i = n+1, \cdots, 2n$, do:
9) set $b_i = \widehat{\mathbf{z}} \star \mathbf{h}_i$;// i-th counter
10) if $b_i \geqslant T$, set $\widehat{r}_i^{(1)} = 1$;

/* Update noisy codeword */
11) compute $\widehat{\mathbf{c}} = \mathbf{c} - \mathbf{x} \cdot \widehat{\mathbf{r}}^{(2)} - \mathbf{y} \cdot \widehat{\mathbf{r}}^{(1)}$.

Fig. 4: BF decoder to estimate coefficients of $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$.

Any solver for this problem can then be used to retrieve $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$. In the following, we consider a straightforward and efficient algorithm that allows us to guess a relatively large number of coefficients in $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$.

Using Bit Flipping Figure 4 describes the algorithm we consider. In the algorithm, we denote by $\mathbf{h}_i$ the i-th column of $\mathbf{H}$ and by $\star$ the integer product between two vectors defined over $\mathbb{F}_2$.

The algorithm is a translation of the basic BF decoder for LDPC codes. For every i, the algorithm computes the counter $\widehat{\mathbf{z}} \star \mathbf{h}_i$ and guesses a non-zero coefficient in $\mathbf{r}$ if the corresponding counter is high enough. Notice that the decoder output corresponds to the updated version of $\mathbf{c}$.

4.2 The Case of Repetition Codes

We show that, when the inner code is a repetition code, one can theoretically estimate the probability distribution of $\mathrm{wt}(\Delta\mathbf{r}^{(1)})$ and $\mathrm{wt}(\Delta\mathbf{r}^{(2)})$. In particular, we assume the distribution is identical for both polynomials. First, we need the following technical propositions, with their proofs in the appendix.

Proposition 1 (Probability to wrongly guess a 1-coefficient). *Let $i \in \{1, 2\}$ and $j \in \{1, \cdots, n\}$ such that $r_j^{(i)} = 1$. Let the inner code be a*

repetition code of odd length n_2 and error-correction capability $t = \lfloor \frac{n_2-1}{2} \rfloor$. We denote the probability that $\widehat{r}_j^{(i)} = 0$ as $\tau_{1\to 0}$ and estimate

$$\tau_{1\to 0} = \sum_{\ell=0}^{T-1} \binom{w}{\ell} \widehat{\rho}^{\,\ell} (1-\widehat{\rho})^{w-\ell},$$

with T the BF threshold and

$$\widehat{\rho} = (1-\widetilde{\rho}) \sum_{\ell=0}^{t-1} \binom{n_2-1}{\ell} \rho_z^\ell (1-\rho_z)^{n_2-1-\ell} + \widetilde{\rho} \sum_{\ell=t}^{n_2-1} \binom{n_2-1}{\ell} \rho_z^\ell (1-\rho_z)^{n_2-1-\ell},$$

where

$$\begin{aligned}\widetilde{\rho} = \widetilde{\rho}_{w,w_r-1} &\left(1-\rho_{w,w_r}\left(1-\frac{w_e}{n}\right)-(1-\rho_{w,w_r})\frac{w_e}{n}\right)\\ &+(1-\widetilde{\rho}_{w,w_r-1})\left(\rho_{w,w_r}\left(1-\frac{w_e}{n}\right)+(1-\rho_{w,w_r})\frac{w_e}{n}\right),\end{aligned}$$

$$\widetilde{\rho}_{w,w_r-1} = \sum_{\substack{\ell\in[1;\min\{w-1,w_r-1\}]\\ \ell \text{ odd}}} \frac{\binom{w-1}{\ell}\binom{n-w}{w_r-1-\ell}}{\binom{n-1}{w_r-1}}.$$

Proposition 2 (Probability to wrongly guess a 0-coefficient). *Let $i \in \{1,2\}$ and $j \in \{1,\cdots,n\}$ such that $r_j^{(i)} = 0$. Let the inner code be a repetition code of odd length n and error-correction capability $t = \lfloor \frac{n-1}{2} \rfloor$. We denote the probability that $\widehat{r}_j^{(i)} = 1$ as $\tau_{0\to 1}$ and estimate*

$$\tau_{0\to 1} = \sum_{\ell=T}^{w} \binom{w}{\ell} \widehat{\rho}^{\,\ell} (1-\widehat{\rho})^{w-\ell},$$

with T the BF threshold and

$$\widehat{\rho} = \widetilde{\rho} \sum_{\ell=0}^{t-1} \binom{n_2-1}{\ell} \rho_z^\ell (1-\rho_z)^{n_2-1-\ell} + (1-\widetilde{\rho}) \sum_{\ell=t}^{n_2-1} \binom{n_2-1}{\ell} \rho_z^\ell (1-\rho_z)^{n_2-1-\ell},$$

where

$$\begin{aligned}\widetilde{\rho} = \widetilde{\rho}_{w,w_r} &\left(1-\rho_{w,w_r}\left(1-\frac{w_e}{n}\right)-(1-\rho_{w,w_r})\frac{w_e}{n}\right)\\ &+(1-\widetilde{\rho}_{w,w_r})\left(\rho_{w,w_r}\left(1-\frac{w_e}{n}\right)+(1-\rho_{w,w_r})\frac{w_e}{n}\right),\end{aligned}$$

$$\widetilde{\rho}_{w,w_r} = \sum_{\substack{\ell\in[1;\min\{w-1,w_r\}]\\ \ell \text{ odd}}} \frac{\binom{w-1}{\ell}\binom{n-w}{w_r-\ell}}{\binom{n-1}{w_r}}.$$

Using the above results, we can derive the weight distribution for each $\Delta\mathbf{r}^{(i)}$.

Proposition 3 (Weight distribution of $\Delta\mathbf{r}^{(i)}$).
The probability that $\Delta\mathbf{r}^{(i)}$ has weight $w_r^{(i)}$ can be estimated as

$$\Pr\left[\mathrm{wt}(\Delta\mathbf{r}^{(i)}) = w_r^{(i)}\right] = \sum_{j=0}^{\min\{w_r, w_r^{(i)}\}} \Pr\left[N_1 = j\right] \Pr\left[N_0 = w_r^{(i)} - j\right],$$

where N_0 and N_1 denote number of wrongly guessed 0-coefficients and 1-coefficients, which follow the distribution

$$\Pr\left[N_0 = j\right] = \binom{n - w_r}{j} (\tau_{0\to 1})^j (1 - \tau_{0\to 1})^{n - w_r - j},$$

$$\Pr\left[N_1 = j\right] = \binom{w_r}{j} (\tau_{1\to 0})^j (1 - \tau_{1\to 0})^{w_r - j}.$$

Proof. Whenever a one-coefficient of $\mathbf{r}^{(2)}$ is guessed correctly, the weight of $\Delta\mathbf{r}^{(2)}$ decreases by 1, while a wrong estimate does not change the weight of $\Delta\mathbf{r}^{(2)}$ in comparison with $\mathbf{r}^{(2)}$. Analogously, guessing a zero-coefficient correctly does not change the weight, while a wrong guess increases the weight by 1. Let N_0 denote the number of wrongly guessed 0-coefficients and N_1 the number of wrongly guessed 1-coefficients. Then, the weight of $\Delta\mathbf{r}^{(2)}$ is $N_0 + N_1$, and

$$\Pr\left[\mathrm{wt}(\Delta\mathbf{r}^{(i)}) = w_r^{(i)}\right] = \sum_{j=0}^{\min\{w_r, w_r^{(i)}\}} \Pr\left[N_1 = j, N_0 = w_r^{(i)} - j\right].$$

To conclude the proof, it is enough to assume that coefficients are guessed independently so that both N_0 and N_1 are the sums of Bernoulli variables with parameters $\tau_{0\to 1}$ and $\tau_{1\to 0}$.

Empirical validation The above formulas are somewhat convoluted: many parameters are employed, and evaluating their interplay is difficult. Further, Proposition 3 models the coefficients of $\Delta\mathbf{r}^{(i)}$ as independent, which is not valid in general, as the inner decoding operates on multiple symbols of $\mathbf{c}$ jointly. We can comment on the effectiveness of the proposed approach using numerical simulations; to this end, consider Figure 5.

As we can see, the approximation as independent random variables works well in practice. Further, there are values of T for which both $\tau_{1\to 0}$ and $\tau_{0\to 1}$ are low. That is, the probability of getting a wrong estimate for a coefficient is low, regardless of its value. This guarantees that both $\widehat{\mathbf{r}}^{(1)}$ and $\widehat{\mathbf{r}}^{(2)}$ are close to $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$, which, in turn, implies that the weights of $\Delta\mathbf{r}^{(1)}$ and $\Delta\mathbf{r}^{(2)}$ are decreased with high probability.

Notice also that if T is too low, then too many 0s are estimated as 1s, and the weight of $\Delta\mathbf{r}^{(1)}$ and $\Delta\mathbf{r}^{(2)}$ may even be larger than that of $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$.

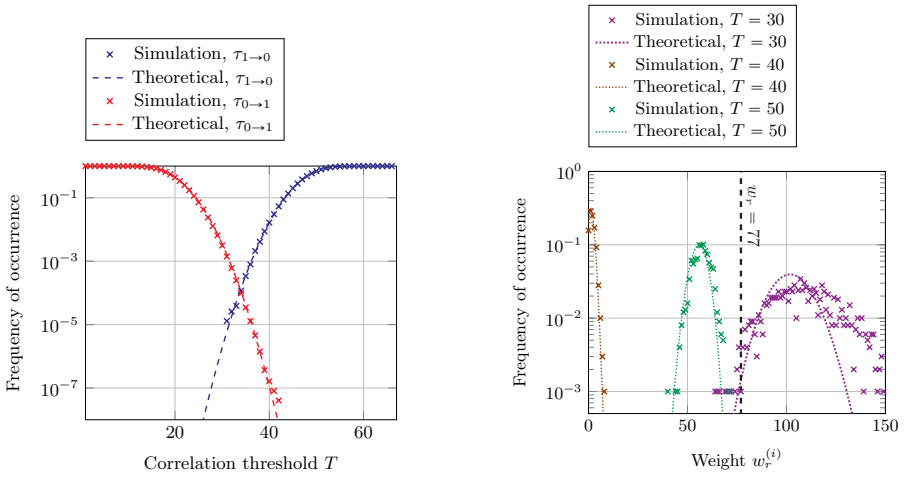


Fig. 5: Comparison of numerical and theoretical estimates for HQC version $\mathsf{BCH} \otimes \mathsf{REP}$. Round-1 parameters of HQC offering 128 bits of security were used, i.e., $n_1 = 766$, $n_2 = 31$, $w = 67$, and $w_r = w_e = 77$. Figure 5a shows the values of $\tau_{1\to0}$ and $\tau_{0\to1}$, Figure 5b reports the distribution of the weight of $\Delta\mathbf{r}^{(1)}$ and $\Delta\mathbf{r}^{(2)}$. The numerical estimates have been obtained averaging over 10^3 trials.

5 Reducing the DFR with a Structure-Aware Decoder

We now show that the technique from the previous section improves HQC decoding: for the same code length n, our decoder has a lower DFR than the original HQC decoder.

The algorithm, summarized in Figure 6, can be viewed as a conventional HQC decoder, enhanced by an initial noise-reducing stage performed according to the algorithm in Figure 4. The final decoding step requires decoding the inner codes a second time, which gives the proposed algorithm an iterative nature. The results of the second inner decoding correlate with the results of the initial decoding: inner blocks that failed in the initial phase are more likely to fail again in the final decoding step. As a consequence of these dependencies, we were not able to derive a reliable DFR model. While the analysis of the proposed decoder is complex, its running time remains efficient, as the following proposition demonstrates.

Proposition 4 (Complexity of proposed decoder). *The computational complexity of the proposed decoder is in*

$$\mathcal{O}\left(T_{\text{inner}} + T_{\text{outer}} + nw + ww_r\right),$$

where T_{inner} *and* T_{outer} *denote the running times of the inner and outer decoders.*

Proof. The initial inner decoding has cost T_{inner}. In the noisy syndrome decoding step, each counter is computed using $\mathcal{O}(w)$ operations since w coefficients are

Input: noisy codeword $\mathbf{c} \in \mathbb{F}_2^n$, secret key $\mathbf{x}, \mathbf{y}$

Output: decrypted message $\mathbf{m}$

/* Estimate error vector by decoding $\mathscr{C}_2$ */
1) set $\widehat{\mathbf{z}} = \mathbf{c} - \mathsf{Dec}_2(\mathbf{c})$;

/* Estimate $\widehat{\mathbf{r}}^{(1)}$, $\widehat{\mathbf{r}}^{(2)}$ */
2) derive $\widehat{\mathbf{r}}^{(1)}$, $\widehat{\mathbf{r}}^{(2)}$ as in Figure 4;

/* Update noisy codeword */
3) compute $\widehat{\mathbf{c}} = \mathbf{c} - \mathbf{x} \cdot \widehat{\mathbf{r}}^{(2)} - \mathbf{y} \cdot \widehat{\mathbf{r}}^{(1)}$;

/* standard HQC decoder */
4) decode $\mathbf{m} = \mathsf{Dec}(\widehat{\mathbf{c}})$.

Fig. 6: Proposed error structure-aware decoder for HQC.

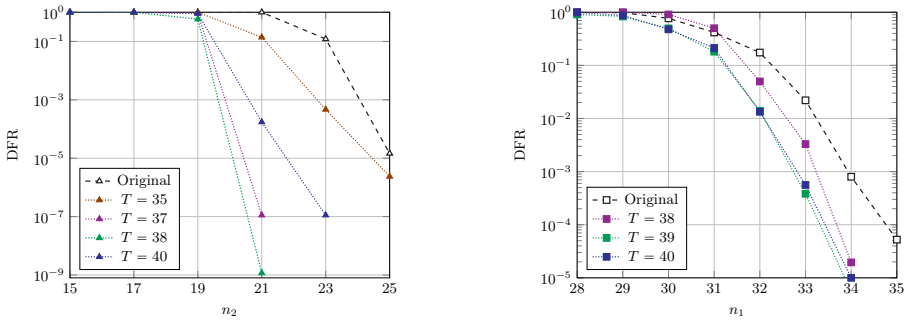
summed. Therefore, the estimation of $\widehat{\mathbf{r}}^{(1)}, \widehat{\mathbf{r}}^{(2)}$ takes time $\mathcal{O}(nw)$. For a properly chosen threshold T, the weights of the polynomials $\widehat{\mathbf{r}}^{(i)}$ do not exceed w_r. Hence, $\mathbf{x}\widehat{\mathbf{r}}^{(2)} + \mathbf{y}\widehat{\mathbf{r}}^{(1)}$ requires $\mathcal{O}(ww_r)$ operations leveraging the sparse nature of the polynomials. The final HQC decoding procedure has cost $T_{\text{inner}} + T_{\text{outer}}$. □

Remark 1. The proposition considers a non constant time implementation of the decoder. Still, we expect that a constant time implementation does not lead to a significant penalty: indeed, any of the strategies employed for BF decoders can probably be employed for our decoder as well.

Proposition 4 shows that the increase in complexity due to the additional noise filtering step is rather limited due to the sparsity of the involved polynomials. Nevertheless, a considerable improvement in decoding performance can be achieved as can be observed empirically, see Figure 7.

In Figure 7a, we focus on the concatenation of BCH and repetition code. The DFR of the decoder is simulated for several threshold values T while varying the inner code length n_2. It can be observed that the optimal threshold value for this setup is $T = 38$, which results in a DFR that is 10^9 times lower than the one of the original decoder when $n_2 = 21$.

In Figure 7b, the concatenation of Reed-Muller and Reed-Solomon code is considered. Since the length of the inner RM code is not flexible, we vary the outer length n_1 instead. Here, the optimal threshold is identified as $T = 39$. Although the improvement is not as substantial as for the concatenation of BCH and repetition code, it is still evident that the DFR of the proposed decoding algorithm is significantly lower than that of the original decoder.



(a) $\mathsf{BCH} \otimes \mathsf{REP}$ (b) $\mathsf{RS} \otimes \mathsf{DRM}$

Fig. 7: Simulation of the DFR of the original HQC decoder and the proposed one, as a function of either n_1 or n_2. Figure 7a considers the $\mathsf{BCH} \otimes \mathsf{REP}$ version with $n_1 = 766$, $t_{\mathrm{BCH}} = 57$, $w = 67$, $w_r = 77$; Figure 7b considers the $\mathsf{RS} \otimes \mathsf{DRM}$ version with $n_{\mathrm{RM}} = 128$ (original length RM code), $mult = 3$ (multiplication factor RM code), $w = 66$, $w_r = w_e = 75$.

6 Conclusion

In this work, we have taken an initial step towards replacing the BSC model currently used in HQC with a channel model that accounts for the specific error structure tied to the secret key.

First, we analyzed the potential improvements in code length that can be achieved. The derived GV-like bounds show that significantly shorter codes than those currently used in HQC can guarantee correct decoding. This stands in contrast to the limited room for improvements under the BSC model. However, while our study reveals a promising direction for more effective code-decoder pairs, it is non-constructive.

Second, we proposed a new error structure-aware decoder. This decoder is a modification of the standard multistage decoder of HQC, incorporating an additional filtering stage. The filtering step leverages the secret key to estimate coefficients of the unknown polynomials that form the error vector. As a result, the weight of errors to be corrected can be reduced, potentially leading to a lower DFR. We have developed a preliminary analysis of the behavior of the proposed decoder. However, precisely modeling the DFR remains a challenge due to the iterative nature of the algorithm.

Acknowledgements. Marco Baldi is supported by the Italian Ministry of University and Research (MUR) PRIN 2022 program under the "Mathematical Primitives for Post Quantum Digital Signatures" (P2022J4HRR) and "Post quantum Identification and eNcryption primiTives: dEsign and Realization (POINTER)" (2022M2JLF2) projects, and under the Italian Fund for Applied Science (FISA 2022), Call for tender No. 1405 published on 13-09-2022 by MUR

- project title "Quantum-safe cryptographic tools for the protection of national data and information technology assets" (QSAFEIT) - No. FISA 2022-00618 - CUP I33C24000520001 - Grant Assignment Decree no. 15461 adopted on 02.08.2024 by the Italian MUR.

Sebastian Bitzer acknowledges the financial support by the Federal Ministry of Education and Research of Germany in the program of "Souverän. Digital. Vernetzt.". Joint project 6G-life, project identification number: 16KISK002.

References

[1] C. Aguilar-Melchor et al. "Efficient error-correcting codes for the HQC post-quantum cryptosystem". In: *Designs, Codes and Cryptography* (2024), pp. 1–20.

[2] M. Albrecht R. et al. *Classic McEliece: conservative code-based cryptography*. `https://classic.mceliece.org/nist/mceliece-20201010.pdf`. 2020.

[3] N. Aragon, P. Gaborit, and G. Zémor. "HQC-RMRS, an instantiation of the HQC encryption framework with a more efficient auxiliary error-correcting code". In: *arXiv preprint arXiv:2005.10741* (2020).

[4] N. Aragon et al. *BIKE - Bit Flipping Key Encapsulation*. `https://bikesuite.org/`. 2023.

[5] M. Baldi et al. "LEDAcrypt: QC-LDPC Code-Based Cryptosystems with Bounded Decryption Failure Rate". In: *Code-Based Cryptography*. Ed. by M. Baldi, E. Persichetti, and P. Santini. Cham: Springer International Publishing, 2019, pp. 11–43. ISBN: 978-3-030-25922-8.

[6] J.-C. Deneuville, P. Gaborit, and G. Zémor. "Ouroboros: A simple, secure and efficient key exchange protocol based on coding theory". In: *Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8*. Springer. 2017, pp. 18–34.

[7] P. Elias. "Coding for noisy channels". In: *IRE WESCON Convention Record*. Vol. 2. 1955, pp. 94–104.

[8] H.-A. Loeliger. "On the basic averaging arguments for linear codes". In: *Communications and Cryptography: Two Sides of One Tapestry* (1994), pp. 251–261.

[9] C. A. Melchor et al. "Hamming quasi-cyclic (HQC)". In: *NIST PQC Round* 2.4 (2018), p. 13.

[10] NIST Computer Security Resource Center. *Post-Quantum Cryptography Standardization*. 2017. URL: `https://csrc.nist.gov/Projects/post-quantum-cryptography/`.

A Proofs of Propositions 1 and 2

We require two preliminary results, which we give and prove here.

Proposition 5. *For two polynomials* $\mathbf{a} \in \mathcal{R}_{w_a}$ *and* $\mathbf{b} \xleftarrow{\$} \mathcal{R}_{w_b}$*, an arbitrary coefficient (say, the first one) in the product* $\mathbf{a} \cdot \mathbf{b}$ *is set with probability*

$$\rho_{w_a,w_b} = \sum_{\substack{\ell \in [1;\min\{w_a,w_b\}] \\ \ell \ odd}} \frac{\binom{w_a}{\ell}\binom{n-w_a}{w_b-\ell}}{\binom{n}{w_b}}.$$

Proof. According to the rules of polynomial multiplication, we have:

$$c_\ell = \sum_{\substack{i,j \\ i+j \equiv \ell \mod n}} a_i \cdot b_j, \quad \text{for } \ell \in \{0, 1, \ldots, n-1\}.$$

We have $c_\ell = 1$ only if, in the above equation, the number of terms $a_i \cdot b_j$ equal to 1 is odd. The probability to have exactly ℓ terms $a_i \cdot b_j$ which are equal to 1 is $\frac{\binom{w_a}{\ell}\binom{n-w_a}{w_b-\ell}}{\binom{n}{w_b}}$. □

Proposition 6. *Let* $\mathbf{x}, \mathbf{y} \in \mathcal{R}_w$*,* $\mathbf{r}^{(1)} \xleftarrow{\$} \mathcal{R}_{w_r}$*,* $\mathbf{r}^{(2)} \xleftarrow{\$} \mathcal{R}_{w_r}$*, and* $\mathbf{e} \xleftarrow{\$} \mathcal{R}_{w_e}$*. An arbitrary coefficient of the polynomial* $\mathbf{z} = \mathbf{x} \cdot \mathbf{r}^{(2)} + \mathbf{y} \cdot \mathbf{r}^{(1)} + \mathbf{e}$ *follows a Bernoulli distribution with parameter*

$$\rho_z = 4\rho_{w,w_r}^2\rho_e - 2\rho_{w,w_r}^2 - 4\rho_{w,w_r}\rho_e + 2\rho_{w,w_r} + \rho_e.$$

Under the assumption that the coefficients behave as independent random variables, the weight of $\mathbf{z}$ *follows a binomial distribution.*

Proof. According to Proposition 5, the i-th coefficient of both products $\widehat{\mathbf{x}} = \mathbf{x} \cdot \mathbf{r}^{(2)}$ and $\widehat{\mathbf{y}} = \mathbf{y} \cdot \mathbf{r}^{(1)}$ is Bernoulli distributed with parameter $\rho_{w,w_r^{(2)}}$ and $\rho_{w,w_r^{(1)}}$. Then, the i-th coefficient of $\mathbf{z}$ will be set with probability

$$\begin{aligned}&\Pr[\widehat{x}_i = 1] \cdot \Pr[\widehat{y}_i = 0] \cdot \Pr[e_i = 0] + \Pr[\widehat{x}_i = 0] \cdot \Pr[\widehat{y}_i = 1] \cdot \Pr[e_i = 0] \\ &\quad + \Pr[\widehat{x}_i = 0] \cdot \Pr[\widehat{y}_i = 0] \cdot \Pr[e_i = 1] + \Pr[\widehat{x}_i = 1] \cdot \Pr[\widehat{y}_i = 1] \cdot \Pr[e_i = 1].\end{aligned}$$

Substituting the probabilities with the associated Bernoulli parameters, we get

$$\begin{aligned}\rho_z = (1-\rho_{w,w_r})(1-\rho_{w,w_r})\rho_e + (1-\rho_{w,w_r})\rho_{w,w_r}(1-\rho_e) \\ + \rho_{w,w_r}(1-\rho_{w,w_r})(1-\rho_e) + \rho_{w,w_r}\rho_{w,w_r}\rho_e.\end{aligned}$$

After some manipulations, we obtain the expression for ρ_z. □

We now proceed to prove Proposition 1; the proof for Proposition 2 is carried out in the same way and is hence omitted.

A.1 Proof of Proposition 1

For $i \in \mathrm{supp}(\mathbf{r}^{(2)})$, we derive the probability distribution of the counter value. We express $\mathbf{r}^{(2)}$ as

$$\mathbf{r}^{(2)} = \underbrace{(0, \cdots, 0, 1, 0, \cdots, 0)}_{\substack{1 \text{ in position } j, \\ 0 \text{ elsewhere}}} + \widetilde{\mathbf{r}}^{(2)},$$

where $\widetilde{\mathbf{r}}^{(2)}$ is equal to $\mathbf{r}^{(2)}$ apart from the coefficient in position j, which is 0 (instead of 1). Observe that $\mathrm{wt}(\widetilde{\mathbf{r}}^{(2)}) = w_r - 1$. The counter associated to $r_j^{(2)}$ is obtained by summing the coefficients of the estimated error vector $\widehat{\mathbf{z}}$ in the positions indexed by $\mathrm{supp}(\mathbf{x}) + j = \{s + j \mod n \mid s \in \mathrm{supp}(\mathbf{x})\}$. For each $\ell \in \mathrm{supp}(\mathbf{x}) + j$, we have:

$$z_\ell = 1 + \underbrace{(\mathbf{x} \cdot \widetilde{\mathbf{r}}^{(2)})_\ell + (\mathbf{y} \cdot \mathbf{r}^{(\mathbf{1})} + \mathbf{e})_\ell}_{\widetilde{z}_\ell} = 1 + \widetilde{z}_\ell,$$

where z_ℓ is the ℓ-th coefficient of $\widetilde{\mathbf{z}} = \mathbf{x} \cdot \widetilde{\mathbf{r}}^{(2)} + \mathbf{y} \cdot \mathbf{r}^{(\mathbf{1})} + \mathbf{e}$. With arguments analogous to those in Proposition 6, we see that $(\mathbf{y} \cdot \mathbf{r}^{(\mathbf{1})} + \mathbf{e})_j$ is Bernoulli distributed with parameter

$$\rho_{w,w_r}\left(1 - \frac{w_e}{n}\right) + (1 - \rho_{w,w_r})\frac{w_e}{n}.$$

Also $(\mathbf{x} \cdot \widetilde{\mathbf{r}}^{(2)})_j$ can be considered as Bernoulli distributed; in particular, the probability that its ℓ-th coefficient is 1 corresponds to

$$\widetilde{\rho}_{w,w_r-1} = \sum_{\substack{\ell \in [1; \min\{w-1, w_r-1\}] \\ \ell \text{ odd}}} \frac{\binom{w-1}{\ell}\binom{n-1-w}{w_r-1-\ell}}{\binom{n-1}{w_r-1}}.$$

Notice that this probability slightly differs from the one in Proposition 5. This is because we are considering only $w - 1$ coefficients of $\mathbf{x}$ and $n - 1$ coordinates for $\widetilde{\mathbf{r}}^{(2)}$ (as one is set to 0).

Putting everything together, we get that $\widetilde{z}_j$ is equal to 1 with probability

$$\begin{aligned}\widetilde{\rho} = \underbrace{\widetilde{\rho}_{w,w_r-1}}_{\Pr\left[(\mathbf{x}\cdot\widetilde{\mathbf{r}}^{(2)})_\ell = 1\right]} & \underbrace{\left(1 - \rho_{w,w_r}\left(1 - \frac{w_e}{n}\right) - (1 - \rho_{w,w_r})\frac{w_e}{n}\right)}_{\Pr\left[(\mathbf{y}\cdot\mathbf{r}^{(1)} + \mathbf{e})_\ell = 0\right]} \\ & + \underbrace{(1 - \widetilde{\rho}_{w,w_r-1})}_{\Pr\left[(\mathbf{x}\cdot\mathbf{r}^{(2)})_\ell = 0\right]} \underbrace{\left(\rho_{w,w_r}\left(1 - \frac{w_e}{n}\right) + (1 - \rho_{w,w_r})\frac{w_e}{n}\right)}_{\Pr\left[(\mathbf{y}\cdot\mathbf{r}^{(1)} + \mathbf{e})_\ell = 1\right]}\end{aligned}$$

Let $\perp_{\mathrm{in}}(\ell)$ indicate the event that the inner code codeword in which the ℓ-th coordinate is contained is wrongly decoded. The complementary event (i.e., a decoding success) is indicated as $\overline{\perp}_{\mathrm{in}}(\ell)$. Then, we have

$$\Pr\left[\widehat{z}_\ell = 1\right] = \Pr\left[\overline{\perp}_{\mathrm{in}}(\ell) \mid \widetilde{z}_\ell = 0\right] \cdot \Pr\left[\widetilde{z}_\ell = 0\right] + \Pr\left[\perp_{\mathrm{in}}(\ell) \mid \widetilde{z}_\ell = 1\right] \Pr\left[\widetilde{z}_\ell = 1\right]. \tag{3}$$

Indeed, the first term (i.e., $\Pr\left[\overline{\perp}_{\text{in}}(\ell) \mid \widetilde{z}_\ell = 0\right] \cdot \Pr\left[\widetilde{z}_\ell = 0\right]$) is the probability that $z_\ell = 1$ and decoding is successful, so z_ℓ is correctly estimated. Analogously, the second term (i.e., $\Pr\left[\perp_{\text{in}}(\ell) \mid \widetilde{z}_\ell = 1\right] \Pr\left[\widetilde{z}_\ell = 1\right]$) corresponds to the probability that $z_\ell = 0$ but there is a decoding failure, so z_ℓ is wrongly estimated.

Let $\text{supp}_{\text{in}}(\ell)$ denote the set of indices that correspond to the same inner codeword as position ℓ. Then, $\widehat{z}_\ell = z_\ell = 1$ if and only if the remaining $n_2 - 1$ positions allow correct decoding of position ℓ, despite position ℓ being erroneous. This happens whenever the number of set coefficients in $\text{supp}_{\text{in}}(\ell) \backslash \{\ell\}$ is not greater than $t - 1$, where $t = \lfloor \frac{n_2 - 1}{2} \rfloor$ denotes the error-correction capability of the inner repetition code. The DFR analysis of HQC assumes independence between the coefficients of the error $\mathbf{z}$ [3]. Similarly, we assume that the positions in $\text{supp}_{\text{in}}(\ell) \backslash \{\ell\}$ are independently Bernoulli distributed with parameter ρ_z as in Proposition 6. Then, the required probabilities can be calculated as

$$\Pr\left[\overline{\perp}_{\text{in}}(\ell) \mid \widetilde{z}_\ell = 0\right] = \sum_{k=0}^{t-1} \binom{n_2 - 1}{k} \rho_z^k (1 - \rho_z)^{n_2 - 1 - k}.$$

With analogous reasoning, we obtain

$$\Pr\left[\perp_{\text{in}}(j) \mid \widetilde{z}_j = 1\right] = \sum_{k=t}^{n_2 - 1} \binom{n_2 - 1}{k} \rho_z^k (1 - \rho_z)^{n_i - 1 - k}.$$

Then, writing $\widehat{\rho} = \Pr\left[\widehat{z}_\ell = 1\right]$, we obtain (3) as

$$\widehat{\rho} = (1 - \widetilde{\rho}) \sum_{k=0}^{t-1} \binom{n_i - 1}{k} \rho_z^k (1 - \rho_z)^{n_2 - 1 - k} + \widetilde{\rho} \sum_{k=t}^{n_2 - 1} \binom{n_2 - 1}{k} \rho_z^k (1 - \rho_z)^{n_2 - 1 - k}. \tag{4}$$

We model $\sum_{j \in \text{supp}\, \mathbf{x} + i} \widehat{z}_j$ as a sum of independent random variables. Consequently, $\sum_{j \in \text{supp}\, \mathbf{x} + i} \widehat{z}_j$ follows the binomial distribution with w trials and success probability $\widehat{\rho}$ and we obtain

$$\tau_{1 \to 0} = \sum_{\ell=0}^{T-1} \binom{w}{\ell} \widehat{\rho}^{\,\ell} (1 - \widehat{\rho})^{w - \ell}.$$