Better Codes for the HQC Cryptosystem

Cyrius Nugier[1][0000−0003−1276−0296] and Jean-Christophe Deneuville[1][0000−0002−5128−6729]

Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse
{cyrius.nugier, jean-christophe.deneuville}@enac.fr

Abstract. In the HQC cryptosystem, the length n of the code determines several concrete parameters such as the bandwidth usage, the memory consumption, or the decoding efficiency. In this paper, we show that currently known methods to explicitly generate asymptotically good (especially with high relative distances), binary codes with efficient associated procedures cannot be used to improve n. We also show that concatenated codes are currently better suited, and by exhausting small codes, find a closer to optimal concatenated code for HQC, which improves upon currently used codes.

1 Introduction

In 1978, Robert J. McEliece set up the ground for using error-correcting codes to obtain a public key encryption scheme. He proved that this can be achieved by encoding a plaintext into a codeword, and then blur the resulting vector with a fixed weight error vector. The legitimate recipient then uses its secret decoder to remove the noise and recover the message. McEliece was able to reduce the hardness of message recovery attacks to solving the syndrome decoding problem. However, key recovery attacks rely on a more intricate notion: distinguishing the family of codes being used from random linear codes.

Numerous works trying to improve the efficiency of the McEliece framework have turned insecure due to distinguishing attacks. Although this has been dealt by Alekhnovich's blueprint [1], no efficient construction was proposed until HQC [2].

HQC uses two different codes: the first one being a random quasi-cyclic code of index 2 (meaning rate $\frac{1}{2}$) that serves to bind the private and public keys and no efficient decoding algorithm is known for that code; the second is left to the choice of the person generating the parameters (concatenated RMRS are suggested in [3]), and the associated efficient decoding algorithm is known to everyone.

Therefore, to produce a ciphertext, a large error (of weight above the correction capacity of the code) is added to the encoded message such that even the knowledge of the decoder does not help decoding. Only the secret key allows to remove sufficiently many errors to make the ciphertext decodable. The cryptosystem is presented in Figure 1.

Parameters:
$n, k, w, w_{\mathbf{r}}, w_{\mathbf{e}}, \mathcal{C}$ of generator matrix $\mathbf{G} \in \mathbb{F}_2^{n \cdot k}$,
$\mathcal{R} = \mathbb{F}_2^n[X]/(X^n - 1)$,
$\mathcal{R}_w = \{\mathbf{x} \in \mathcal{R} | HW(\mathbf{x}) = w\}$
KeyGen():
$\mathbf{h} \xleftarrow{\$} \mathcal{R}$,
$sk = (\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{R}_w \times \mathcal{R}_w$,
$pk = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y})$
Encrypt(pk, $\mathbf{m} \in \mathbb{F}_2^k$):
$\mathbf{e} \xleftarrow{\$} \mathcal{R}_{w_{\mathbf{e}}}$,
$\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \xleftarrow{\$} \mathcal{R}_{w_{\mathbf{r}}} \times \mathcal{R}_{w_{\mathbf{r}}}$,
$ct = (\mathbf{u}, \mathbf{v}) = (\mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2, \mathbf{mG} + \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{e})$
Decrypt(sk, ct):
$\mathbf{m}' = \mathcal{C}.Decode(\mathbf{v} - \mathbf{u} \cdot \mathbf{y})$

Fig. 1. The HQC cryptosystem.

The most determinant parameter in HQC's performance is n. The private, public key, and ciphertext (denoted sk, pk, and ct) are of size $2n$, and the product of elements in $\mathcal{R}$ is computed as the product of a n-by-n circulant matrix and a n-by-1 vector. All other parameters have constraints and are already at their minimum values: k must be at least the security parameter λ since it is also the size of the message. The noise levels $w, w_{\mathbf{r}}, w_{\mathbf{e}}$ have values large enough to ensure the hardness of the Syndrome Decoding (SD) problem. These values are also the smallest possible in order to decrease the remaining noise level before decoding in Decrypt. This noise level, called Bit Error Rate (BER) (denoted $p^\star$) creates a constraint on the code $\mathcal{C}$ since it should fail to decode legitimate ciphertexts with negligible probability, called Decoding Failure Rate $dfr(p^\star) \leq 2^-\lambda$. The smaller $p^\star$, the smaller the code length n can be.

Every parameter except n is therefore minimal to ensure the cryptosystem security requirements. We recall the values used for the NIST submission [3] in table 1. Note that in order to avoid structural (folding) attacks such as [4] and [5], the length of the code has to be a primitive number.

Table 1. Parameter sets for HQC

Category	n_{HQC}	k	w	$w_{\mathbf{r}}$	$w_{\mathbf{e}}$
I	17669	128	66	75	75
III	35851	192	100	114	114
V	57637	256	131	149	149

Codes of interest in the HQC cryptosystem therefore need to meet the following requirements:

- The code should be binary;
- The dimension should at least equal to the security parameter: $k \geq \lambda$;
- The code should be asymptotically good, meaning that asymptotically both the rate k/n and relative distance d/n should be non-zero;
- The construction of the code should be explicit;
- Encoding should be efficient;
- Decoding should be efficient, such that

$$DFR = -\log_2(dfr(p^{\star}_{(n,w,w_{\mathbf{e}},w_{\mathbf{r}})})) \geq \lambda$$

In Section 2, it is shown that the current methods are not satisfying in producing such a code if it is constructed as a single binary code. In Section 3, we instead look at the construction of a concatenated code and explore all options to propose new codes with a smaller length n than in HQC.

2 Designing Binary codes

2.1 Decoding Failure Rate

The DFR can be upper bounded in multiple ways. The most conservative upper bound consists in considering that as long as the number of errors is greater than the error-correction capacity, the decoding fails. This yields the following inequality:

$$dfr(p^{\star}) \leq \sum_{i=t+1}^{n} \binom{n}{i} (p^{\star})^i (1-p^{\star})^{n-i}. \tag{1}$$

This bound is only met in Maximum Distance Separable (MDS) codes, where every $\mathbb{F}_2^n$ element is at most at distance $t = \frac{d-1}{2}$ of a codeword. In the design document of HQC, another bound is introduced [6], which fits more tightly the experimental observations. The idea is that in non-MDS codes, it is possible for minimum distance decoding to correct more than t errors. Any point at a distance slightly greater than t of any codeword still has a high probability that it is not closer to another codeword.

$$dfr(p^{\star}) = (2^k - 1) \sum_{i=d/2}^{d} \binom{d}{i} (p^{\star})^i (1-p^{\star})^{d-i} \tag{2}$$

This formula is designed for Reed-Muller (RM) codes of order 1 because their structure guarantees that every two nonzero codewords are at distance d of each other. This can be used as an upper bound for the general case where the average distance between codewords is greater than d.

2.2 Bounds on codes characteristics

The first theoretical limit on the minimal length of the code is given in [7]. It is $n_{min} = 13534$, 31411 and 45064 for categories I, III, and V respectively. This is 14% to 24% lower than the current codes used in HQC. Given a chosen n, there is a minimal d that gives a $DFR \geq \lambda$, as can be seen in Figure 2.

It is known as the Plotkin bound that binary codes can have at most relative distances $\delta = \frac{d}{n}$ of $\frac{1}{2}$. Although some code families reach this bound (for instance, Reed-Muller codes of order 1), there is no guarantee of the existence of a $[n, k, n \cdot \frac{1}{2}]$ code for most n and k. However, it is also known, as the Gilbert-Varshamov bound, that for every $0 \leq \delta < \frac{1}{2}$ there exists a family of binary codes with rate $R \geq 1 - H(\delta)$ and relative distance δ, where H is the binary entropy function:

$$H(p) = -p \log_2 p - (1-p) \log_2 (1-p).$$

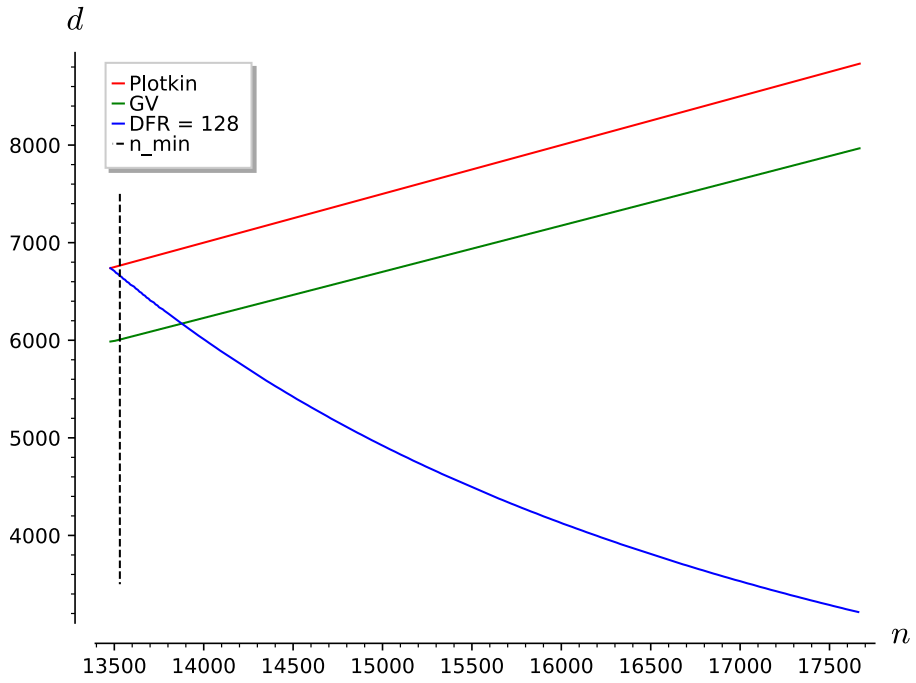

Fig. 2. Above the blue line, codes are usable for HQC. Below the green line, codes are guaranteed to exist. Above the red line codes are guaranteed to not exist. The dashed black line represents the theoretical minimum for HQC [7]. $k = 128$.

This two bounds gives us a space in which we know codes exist, but with no explicit construction. This search space is the area above the blue and below the green line. Note that the codes that are the furthest from the GV bound

(and therefore probably easier to find) with $n = n_{HQC}$ are $[17669, 128, 3211]$, $[35851, 192, 6603]$, and $[57637, 256, 10433]$. The smallest code for HQC with a guaranteed existence is $[13883, 128, 6173]$ (a code with $n = 13878$ exists, but it would need to be extended so that its length is a primitive number). The hypothetical smallest $n \geq n_{min}$ code is $[13613, 128, 6543]$, and for information, the intersection of the Plotkin Bound and the DFR line is $[13478, 128, 6739]$.

The situation is very similar for category V, however, for category III, the $n_{min} = 31469$ vertical line is placed on the left side of the intersection of the blue and green lines, as the DFR of the GV code $[31469, 192, 14309]$ has a DFR of 378. This suggests that it could be interesting to increases the noise values $w, w_{\mathbf{r}}, w_{\mathbf{e}}$ to lower n_{min} while increasing SD security.

2.3 Generating codes with expansion methods

To illustrate the difficulty of generating codes in the search region, in category I, the closest Bose-Chaudhuri-Hocquenghem (BCH) Code with $n_{min} \leq n \leq n_{HQC}$ and $k \geq 128$ is the $[1023, 133, 179]$ BCH code duplicated 17 times, i.e. $[17391, 133, 3043]$, which has a relative distance of only 0.175 while the GV bound guarantees that codes with the same length and distances up to 7835 exist (relative distance 0.45).

However, there exists some families of codes known to reach the GV bound, for instance the random codes. In [8] section 2.1 shows that random codes of rate $R = 1 - H(\delta)$ have an average relative distance of δ. Two problems arise when using these codes for HQC: first, there is no guarantee on the distance, and computing it is exponential in $k = \lambda$, which is not doable by design. Second, There would be no easy to determine decoding algorithm that runs in polynomial time, since the algorithm working for any code, minimum distance decoding, is exponential.

A few methods are known that try to replicate the GV-reaching properties of random codes while preserving a structure, in order to have both distance properties and efficient decoding. These are based on expanding a starting code $[n_0, k, n \cdot \frac{1-\epsilon_0}{2}]$ into a larger one $[n_1, k, n \cdot \frac{1-\epsilon_1}{2}]$, where $n_1 > n_0$ and $\epsilon_1 < \epsilon_0$. This section shows that all the currently known constructions based on this approach ϵ_1 low enough to be useful for HQC only when n_1 is larger than the current parameters for HQC. For information, the HQC-equivalent binary codes mentioned earlier have ϵ slightly above 0.63 for all categories, and lower epsilons are required to reach lower n.

The AGHP constructions In these constructions and all that follows, the starting code is viewed as the set $\mathcal{S}_0$ of columns of its generating matrix (n elements of $\mathbb{F}_2^k$). This set is said to be (n, k, ϵ)-biased if the distance between two codewords is always between $(\frac{1-\epsilon}{2})n$ and $(\frac{1+\epsilon}{2})n$. The main idea is to generate sets with arbitrarily small bias. Note that these methods all start with a fixed k and do not change it.

In [9], it is shown that a (n, k, ϵ)-biased set can be constructed with support size $n \leq \frac{2k^2}{\epsilon^2}$. With $k = 128/192/256$ and ϵ always smaller than 1, the minimal support (code length) is $n = 32768/73728/131072$ which is above n_{HQC} for all security categories.

Random sampling This method allows to create a set $\mathcal{S}_1$ of bias smaller than the starting set by sampling all combinations of t elements from the starting set $\{(z_1, \cdots z_t) | z_i \in \mathcal{S}_0\}$, and defining $\mathcal{S}_1$ as the set of all $\bigoplus_{i=1}^{t} z_i$. This set is of support n_0^t and has a bias of ϵ_0^t.

The case $t = 1$ leaves the set as is. For $t = 2$, if we wanted to have a n_1 less than n_{HQC}, there would have to be some $n_0 \leq 132/189/240$. For category I and $k = 128$, it is possible to retrieve from [10] that the best known code is $[132, 128, 2]$, which has $\epsilon_0 \approx 0.97$, which gives an unsatisfying $\epsilon_1 \approx 0.94$. For categories III and V, $n_0 < k$ and therefore there is no possible starting code that improves the bias with support small enough. For the same reason, a bigger t will also not work, for all categories.

Random walks of length one This technique, attributed in [11] to Rozenman and Wigderson [12] consists of replacing the uniform sampling by a pseudo-random one. To do so, they rely on an expander graph. These are of order n, are D-regular and are noted (n, D, λ). Such regular undirected graphs have Hermitian transition matrices with real eigenvalues $\lambda_1 = 1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. λ denotes the maximum between λ_2 and $-\lambda_n$. Good expander graphs will have a lower λ, with optimal expanders called Ramanujan that verify $\lambda \leq \frac{2\sqrt{D-1}}{D}$.

Taking all the (i, j) edges of the expander graph allows to create the set of all $z_i \oplus z_j$, which is proved to have bias $\epsilon_1 = \lambda + \epsilon_0^2$. In order to have a bias improvement $\epsilon_1 < \epsilon_0$, we need to have $\lambda < 0.25$. With a Ramanujan expander, having a λ this low requires at least $D \geq 63$ and therefore a $n_0 \leq 280$. While a $[280, 128, 70]$ code is not known yet, it is still slightly above the Griesmer lower bound on n for this k and d, which indicates that it may exist.

However, if we take the problem from the other side, the closest currently known code (as listed in [10]) is $[256, 128, 38]$, which has $\epsilon_0 \approx 0.70$ and therefore requires $\lambda < 0.209$ to give an improvement, which implies $D \geq 91$, and therefore a $n_1 \geq 23296 > n_{HQC}$. The problem is the same or worse for the other security categories, given that less good codes with $k = 192$ or $k = 256$ are explicitly known.

Random walks of length t This construction is discussed in detail in [11], as a way to get codes closer to the GV bound. It requires the introduction of a much smaller second graph (D, D_2, λ_2) and a map between the D edges of each vertex of the first graph and the vertices of the second graph. By using this map, walks are made alternatively on the two graphs, which gives a better bias. It produces a set with support $n_1 = n_0 D_2^t$, with a bias $\epsilon_1 = (\epsilon_0 + 2\beta + 2\lambda_2)^{\lfloor t/2 \rfloor}$, for a small positive β.

Even with $\beta = 0$, improving the bias requires $t \geq 4$. In order to have $\lambda_2 = 0.25$ small enough to make any improvement, even with Ramanujan graphs, the same $D_2 \geq 63$ is required, which already gives $D_2^t \gg n_{HQC}$.

Ta-Shma s-wide walks This method produces explicit binary codes asymptotically even closer to the GV bound [11]. To do so, the second graph is taken to be (D^s, D_2, λ_2). It changes the definition of walks in the first graph to have s coordinates, and at step t, the walk over the second graph corresponds to a change to the coordinate $t \, mod \, s$ of the path in the first graph.

This construction, while asymptotically a better construction than the previously mentioned ones, has a support size of $n_0 D^s D_2^t$. There is not a single choice of D, D_2, t and s that would give any bias improvement for values of n small enough to be useful in HQC. Additionally, codes created by this method do not have known efficient decoders without the following adaptation.

Jeronimo construction This last method [13] solves the decoding limitation of Ta-Shma codes by adapting the construction with a small compromise on GV proximity. However, the proposed decoding is a list decoding, which cannot be used for HQC. The proposed unique decoding algorithm only works to half the minimum distance of the code. In order for the code to meet the DFR requirement, it would need to have twice the distance. Even if some such codes are still under the GV bound, this reduces ϵ, when the supports of codes created with this construction are larger than Ta-Shma ones.

To conclude this section, binary codes that could be used for HQC are currently either too big to be defined explicitly with a formal proof on their minimum distance, or too small to be generated through an expansion method. It remains possible, instead of using an unique binary code, to use multiple codes in different characteristics, a technique commonly known as code concatenation.

3 Concatenated codes

Code concatenation is a technique allowing to benefit from codes that have a very high relative distance at the expense of a very low rate on one hand, and on the other hand from codes with very high rate but a less than ideal relative distance [14].

In a concatenated code, a message is composed of k_2 elements of size k_1. It is first encoded with a code $\mathcal{C}_2[n_2, k_2, d_2]_{q^{k_1}}$, returning n_2 elements of size k_1. Each of those is then encoded in a $\mathcal{C}_1[n_1, k_1, d_1]_q$ code, each returning n_1 elements in $\mathbb{F}_q$. A code defined this way is a $[n_1 n_2, k_1 k_2, d_1 d_2]_q$ code. Decoders must then be used in reverse order. First, the q-ary input is decoded with the $\mathcal{C}_1$ decoder over blocks of size n_1, returning n_2 elements of size k_1 that are then decoded with the $\mathcal{C}_2$ decoder.

There is a well known result on the lower bound of rates reachable by concatenated codes given a relative distance called the Zyablov bound [15]. Codes

meeting this bound are constructed by setting $\mathcal{C}_1$ as a code with good relative distance, for instance a code on the GV bound, and setting $\mathcal{C}_2$ as an MDS code. This construction allows concatenated codes with asymptotically low rates to approach the GV bound.

3.1 HQC codes

In the Round 4 HQC cryptosystem, the codes in use are, first, a Duplicated Reed-Muller of order 1 for $\mathcal{C}_1$. This family of codes is optimal since they have a relative distance $\delta = \frac{1}{2}$, which is even better than the GV bound for codes of this length. Second, a shortened Reed-Solomon code is used for $\mathcal{C}_2$, which is an MDS code.

The exact codes currently used in HQC are shown in Table 2, where SD is the hardness level of the syndrome decoding as estimated by [16]. The SD hardness is minimal to meet the NIST requirements. The values are based on the complexity of brute-forcing AES with key sizes of 128/192/256, which adds 15 to λ. The remaining margin is to ensure to be out of reach of some attacks such as DOOM (an additional $\log_2(\sqrt{n})$ bits of security are required) [17]. Notice that $n > n_1 n_2$ because n has to be a primitive number to avoid folding attacks [4]. It is therefore the smallest primitive number greater than $n_1 n_2$.

Table 2. Current concatenated codes for HQC

Cat.	n	k	SD	DFR	Reed-Muller $[n_1, k_1, d_1]_2$	Reed-Solomon $[n_2, k_2, d_2]_{2^{k_1}}$
I	17669	128	152	141	$[384, 8, 192]$	$[46, 16, 31]_{2^8}$
III	35851	192	222	201	$[640, 8, 320]$	$[56, 24, 33]_{2^8}$
V	57637	256	283	272	$[640, 8, 320]$	$[90, 32, 59]_{2^8}$

Although these are good codes for HQC, there is no design explanation on the choice of these specific codes. Also, it can be noted that the DFRs are above their expected minimal value of λ, which seems to indicate that a margin to reduce n exists. The goal of the following section is to explain how we found codes with a length n lower than the codes used in HQC.

3.2 Experiments and results

As discussed in previous sections, concatenated codes are better suited for an HQC usage. The choice of Duplicated order 1 Reed-Muller with Shortened Reed-Solomon codes is optimal for the concatenated code construction, therefore, we can only search for other combinations of codes in the same families. With the objective of finding the single best possible concatenated codes, we evaluated the DFR over all possible combinations of Duplicated order 1 Reed-Muller and

Shortened Reed-Solomon codes. The Duplicated Reed-Muller codes have the following form: $[m \cdot 2^{k_1-1}, k_1, m \cdot 2^{k_1-2}]$, and Shortened Reed-Solomon codes have the form $[n_2, k_2, n_2 - k_2 + 1]$, which reduces the search space to four variables: k_1, m, k_2, n_2. The limits are the following :

- k_1 is between 2 and $\lfloor \log_2(n_{HQC}) \rfloor$. This way, $n_1 = 2^{k_1-1}$ is at least small enough for n_2 to be an integer and $n_1 n_2 \leq n_{HQC}$.
- k_2 must be large enough so that $k_1 k_2 \geq \lambda$
- m has to be such that $n_{min} \leq m \cdot 2^{k_1-1} \cdot n_2 \leq n_{HQC}$ (or more precisely, the largest primitive number smaller than n_{min} plus one) n_2 , and small enough for n_2 to be an integer.
- Finally, n_2 goes from k_2 to $2^{k_1} - 1$.

We ran DFR computations for each of these codes and for each security parameter. We then extracted all codes with $DFR \geq \lambda$ and sorted them by length. The best codes for each security category is presented in Table 3. By precaution, we also ran tests with noise levels one higher than the ones currently used in the NIST submission. Only one code in this configuration returned with length at least equivalent to the official one, which is noted III* ($w = 101, w_{\mathbf{r}} = w_{\mathbf{e}} = 115$). SD hardness and DFR are rounded down.

Table 3. Best concatenated codes for HQC

Cat.	n	k	SD	DFR	Reed-Muller $[n_1, k_1, d_1]_2$	Reed-Solomon $[n_2, k_2, d_2]_{2^{k_1}}$
I	16901	130	132	128	$[512, 10, 256]$	$[33, 13, 21]_{2^{10}}$
III	35339	200	222	197	$[512, 10, 256]$	$[69, 20, 50]_{2^{10}}$
III*	35851	200	223	202	$[512, 10, 256]$	$[70, 20, 51]_{2^{10}}$
V	56333	264	283	267	$[1024, 11, 512]$	$[55, 24, 32]_{2^{11}}$

Here are some remarks about the results:

- The reduction of n is 4.4% / 1.4% / 2.2% according to the category (III* does not change n, just the SD security).
- The dimension k is higher than λ.
- The DFR margins are reduced as expected.
- The dimension of the Reed-Muller codes are no longer 8. It also seems that the duplication is not a good idea to optimize n (as $m = 1$ in each of our proposed codes).

3.3 Impact of the changes

As stated in the Introduction, in HQC, all key sizes are linearly dependent on n, therefore, the gain will be 4.4% / 1.4% / 2.2%. Throughout the scheme, all

random samplings, as well as additions in $\mathcal{R}$ are run in $\mathcal{O}(n)$ time, and can expect the same improvements. All products in $\mathcal{R}$ have a complexity in $\mathcal{O}(n^2)$, and can expect improvements of 8.5% / 2.8% / 4.4%. This improvements may vary depending on the computer architecture and available parallelism.

However, the decoding algorithm is more dependent on the codes. For instance, order 1 Reed-Muller codes can be decoded using the Fast Hadamard Transform [18], which has a complexity of $\mathcal{O}(n_1 \log n_1)$. Since this decoding has to be done over n_2 blocs, the total complexity is in $\mathcal{O}(n_1 n_2 \log n_1)$, which then represents a change of +0.3% / -4.8% / +4.9%.

For the Reed-Solomon decoding, the Berlekamp-Massey algorithm is used, which has a complexity of $\mathcal{O}(n_2^2)$ operations in $\mathbb{F}_{2^{k_1}}$ [19]. The resulting changes in the number of operations is of -48% / +52% / -62%. The complexity of each operation is very dependent on the implementation and the computer architecture, and may change because of the changes in k_1.

In order to get a general idea on the effect of the proposed changes on the decryption computation time, we evaluated the cycle counts of each subroutine on a single code 32-bit RISC-V, with no SIMD capacity or other hardware accelerator for HQC and obtained the results displayed in Table 4.

Table 4. Cycle counts and relative proportions for HQC Decrypt subroutines on a 32-bit RISC-V

Category	$\mathbf{v} - \mathbf{u} \cdot \mathbf{y}$	Reed-Muller	Reed-Solomon
I	39231162	456405	4276642
	89.23%	1.04%	9.73%
III	120618333	674453	5499330
	95.13%	0.54%	4.33%
V	220572495	1083912	13424979
	93.83%	0.46%	5.71%

The expected changes on the whole Decrypt routine computation times considering these proportions should be around -8.6% / -2.0% / -6.1%.

4 Conclusion

This paper uses two approaches to try to optimize the length n of the code used in the NIST round 4 HQC cryptosystem. The first approach consists in finding a single binary code with enough error correction capacity in order to keep the decryption error rate below the security parameter for each security category. While the GV bound ensures the existence of better codes, we show that the current methods that creates codes close to this bound asymptotically are not able to reach this proximity for n small enough for HQC. Now that

near-optimal expanders have been found, a good research perspective could be to find trade-offs between asymptotical bias and minimum support for a given bias.

The second approach is therefore to use codes in different characteristics (concatenated codes). The families of codes currently in use in HQC are optimal for concatenated codes. Our contribution is to provide the single best concatenated codes by an exhaustive search over all parameters of Duplicated Reed-Muller - Shortened Reed-Solomon concatenations. We found 3 optimal codes improving the code length and one improving security. We also proposed an estimate of the impact of the change of codes over HQC performance.

Acknowledgements

This work was supported by ICO, Institut Cybersécurité Occitanie, funded by Région Occitanie, France. The authors would like to express their gratitude to Eleonora Guerrini for insightful discussions on an early version of this work.

References

1. M. Alekhnovich, "More on average case vs approximation complexity," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE, 2003, pp. 298–307.
2. C. Aguilar-Melchor, O. Blazy, J.-C. Deneuville, P. Gaborit, and G. Zémor, "Efficient encryption from random quasi-cyclic codes," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3927–3943, 2018.
3. C. Aguilar Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, and G. Zémor, "Hamming quasi-cyclic (hqc)," *NIST PQC Round 2*, vol. 2, no. 4, p. 13, 2018.
4. J. Faugère, A. Otmani, L. Perret, F. de Portzamparc, and J. Tillich, "Structural weakness of compact variants of the mceliece cryptosystem," in *2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, June 29 - July 4, 2014.* IEEE, 2014, pp. 1717–1721. [Online]. Available: https://doi.org/10.1109/ISIT.2014.6875127
5. C. Löndahl, T. Johansson, M. Shooshtari, M. Ahmadian Attari, and M. Aref, "Squaring attacks on mceliece public-key cryptosystems using quasi-cyclic codes of even dimension," *Designs, Codes and Cryptography*, vol. 80, 06 2015.
6. C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, and G. Zémor, "Hamming quasi-cyclic (hqc) fourth round version," 2024. [Online]. Available: https://pqc-hqc.org/documentation.html
7. C. A. Melchor, N. Aragon, J. Deneuville, P. Gaborit, J. Lacan, and G. Zémor, "Efficient error-correcting codes for the HQC post-quantum cryptosystem," *Des. Codes Cryptogr.*, vol. 92, no. 12, pp. 4511–4530, 2024. [Online]. Available: https://doi.org/10.1007/s10623-024-01507-6
8. R. G. Gallager, *Low-Density Parity-Check Codes.* The MIT Press, 09 1963. [Online]. Available: https://doi.org/10.7551/mitpress/4347.001.0001
9. N. Alon, O. Goldreich, J. Håstad, and R. Peralta, "Simple constructions of almost k-wise independent random variables," *Random Structures & Algorithms*, vol. 3, no. 3, pp. 289–304, 1992.

10. M. Grassl, "Bounds on the minimum distance of linear codes and quantum codes," Online available at http://www.codetables.de, 2007, accessed on 2025-01-10.
11. A. Ta-Shma, "Explicit, almost optimal, epsilon-balanced codes," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 2017, pp. 238–251.
12. A. Bogdanov, "A different way to improve the bias via expanders," *Topics in (and out) the theory of computing, Lecture*, vol. 12, p. 2012, 2012.
13. F. G. Jeronimo, D. Quintana, S. Srivastava, and M. Tulsiani, "Unique decoding of explicit epsilon-balanced codes near the gilbert-varshamov bound," in *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2020, pp. 434–445.
14. G. Forney, "Convolutional codes i: Algebraic structure," *IEEE Transactions on Information Theory*, vol. 16, no. 6, pp. 720–738, 1970.
15. V. V. Zyablov, "An estimate of the complexity of constructing binary linear cascade codes," *Problemy Peredachi Informatsii*, vol. 7, no. 1, pp. 5–13, 1971.
16. A. Esser, J. A. Verbel, F. Zweydinger, and E. Bellini, "{SoK: CryptographicEstimators - a Software Library for Cryptographic Hardness Estimation}," in *AsiaCCS*. {ACM}, 2024.
17. N. Sendrier, "Decoding one out of many," Cryptology ePrint Archive, Paper 2011/367, 2011. [Online]. Available: https://eprint.iacr.org/2011/367
18. A. E. Ashikhmin and S. N. Litsyn, "Fast decoding algorithms for first order reed-muller and related codes," *Designs, Codes and Cryptography*, vol. 7, no. 3, pp. 187–214, 1996.
19. J. Massey, "Shift-register synthesis and bch decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.