A Study of Error Floor Behavior in QC-MDPC Codes

Sarah Arpin[1], Tyler Raven Billingsley[2], Daniel Rayor Hast[3], Jun Bo Lau[4], Ray Perlner[5], and Angela Robinson[5(✉)]

[1] Department of Mathematics, University of Colorado Boulder, Boulder, USA
[2] Department of Mathematics, Rose-Hulman Institute of Technology, Terre Haute, USA
[3] Department of Mathematics and Statistics, Boston University, Boston, USA
[4] Department of Mathematics, University of California San Diego, San Diego, USA
[5] Computer Security Division, National Institute of Standards and Technology, Gaithersburg, USA
angela.robinson@nist.gov

Abstract. We present experimental findings on the decoding failure rate (DFR) of BIKE, a fourth-round candidate in the NIST Post-Quantum Standardization process, at the 20-bit security level. We select parameters according to BIKE design principles and conduct a series of experiments. We directly compute the average DFR on a range of BIKE block sizes and identify both the waterfall and error floor regions of the DFR curve. We then study the influence on the average DFR of three sets $\mathcal{C}$, $\mathcal{N}$, and $2\mathcal{N}$ of near-codewords—vectors of low weight that induce syndromes of low weight—defined by Vasseur in 2021. We find that error vectors leading to decoding failures have small maximum support intersection with elements of these sets; further, the distribution of intersections is quite similar to that of sampling random error vectors and counting the intersections with $\mathcal{C}$, $\mathcal{N}$, and $2\mathcal{N}$. Our results indicate that these three sets are not sufficient in classifying vectors expected to cause decoding failures. Finally, we study the role of syndrome weight on the decoding behavior and conclude that the set of error vectors that lead to decoding failures differ from random vectors by having low syndrome weight.

Keywords: BIKE · Error-correcting codes · McEliece · PQC · QC-MDPC

1 Introduction

In 2016, the U.S. National Institute of Standards and Technology (NIST) announced a Post-Quantum Cryptography (PQC) standardization process aimed at updating NIST's public-key cryptographic standards to include post-quantum cryptography, that is, cryptographic algorithms that are thought to be

J. H. Cheon and T. Johansson (Eds.): PQCrypto 2022, LNCS 13512, pp. 89–103, 2022.
https://doi.org/10.1007/978-3-031-17234-2_5

secure against attacks by a quantum computer. One of the remaining code-based candidates in the NIST PQC Standardization process is BIKE, a cryptosystem based on quasi-cyclic moderate density parity check (QC-MDPC) codes.

The BIKE cryptosystem was originally designed for ephemeral use, that is in settings where a KEM key pair is generated for every key exchange. The requirement for BIKE to be used ephemerally provides a countermeasure to a reaction attack by GJS [10] wherein an attacker can use knowledge of messages that lead to decoding failures to recover the private key of a scheme. During the second and third round of the NIST PQC process, BIKE proposed parameter sets that were designed to provide security in the static-key setting [1], that is, a setting where KEM key pairs can be reused for several key exchanges. In fact all the parameter sets in the third round specification of BIKE are designed to be secure in the static-key setting, although they do not formally claim to be secure in this setting. While security in the ephemeral setting can be provided by a scheme meeting the weaker IND-CPA security notion, security in a static-key setting requires a scheme meeting the stronger IND-CCA2 security notion. Achieving IND-CCA2 security requires that BIKE's decoder has a sufficiently low decoding failure rate (DFR), both because the security proof of BIKE in the IND-CCA2 setting assumes a low DFR, and because if a QC-MDPC cryptosystem with a sufficiently high DFR is used in the static-key setting, it would allow an attacker to perform the GJS attack with a high probability of success.

By design, it is not feasible to directly compute an average DFR for BIKE at cryptographically relevant security levels. It is possible to measure DFRs for smaller code sizes and then use extrapolation methods to estimate the DFR for larger parameters [9,17]. One must consider the phenomenon known as the *error floor* region of DFR curves to avoid an underestimate of DFR for larger code sizes. It is known that for LDPC and MDPC codes, the logarithm of the DFR drops significantly faster than linearly, and then linearly as the signal-to-noise ratio is increased [15,21]. Thus a typical DFR curve contains a concave *waterfall* region followed by a near-linear *error floor* region. One must accurately predict the error floor of a DFR curve to accurately predict the DFR for cryptographically relevant code sizes.

The error floor regions for low density parity check (LDPC) codes have been extensively analyzed in the literature. These are codes which can be defined by parity check matrices $H_{k\times n}$ with row Hamming weight on the order of $O(1)$, or up to $O(\log(2n))$. For each parity check matrix, there is a corresponding bipartite graph, known as a Tanner graph. Much analysis of iterative LDPC decoding behavior focuses on properties of Tanner graph representations of the code [4,14–16,24], such as identifying *stopping sets* and *trapping sets*.

Recent work [22,23] has considered several factors affecting the DFR of QC-MPDC codes: choice of decoder [17,20], classes of weak keys, and sets of problematic error patterns. It was noted that error vectors with a small Hamming distance from problematic error patterns—error vectors of low weight that emit syndromes of low weight—are significant contributors to the error floors of QC-

MDPC codes and it was concluded that these vectors were rare enough to not affect the overall DFR predictions for higher code sizes.

In this work, we examine the error floor behavior of QC-MDPC codes and focus on a scaled-down version of BIKE. Existing analysis of the DFR for BIKE [9,17] relies on extrapolations based only on modifying the block size, but this analysis is only accurate if an upper bound can be established for the DFR at which the transition to error floor behavior occurs (see e.g. Assumption 3 on page 7 of [17]). Vasseur's thesis uses experiments with error vectors based on known classes of codewords and near-codewords to give an upper bound for the transition DFR. We try to directly measure the transition point and see if it can be modeled based on the known contributions to error floor behavior described in Vasseur's thesis, but we cannot directly measure the transition point for cryptographic size parameters, since that transition occurs at too low a DFR. We use the Black-Grey-Flip decoder [9], the recommended BIKE decoder as of the time of writing, and filter out any keys belonging to the classes of weak keys defined by [22]. We consider the three sets of *near codewords* as defined in [22] and find that error vectors that lead to decoding failures have small (between 2 and 8 bits) support intersections with elements of this set. We conclude that error vectors that emit syndromes of low weight are significant contributors to decoding failures, but are not fully captured by the sets of near codewords defined in [22].

2 Background

2.1 Coding Theory and QC-MDPC Codes

Throughout this document, let $\mathbb{F}_2$ denote the finite field of two elements. For $r \in \mathbb{N}$, $x \in \mathbb{F}_2^r$, let $|x|$ denote the Hamming weight of x. For two vectors $x, y \in \mathbb{F}_2^r$, let $x \star y = (x_0 \cdot y_0, x_1 \cdot y_1, \ldots, x_{r-1} \cdot y_{r-1})$ denote the Schur product. Let $\mathcal{C}(n,k)$ be a binary linear code, $n, k \in \mathbb{N}$. Then $\mathcal{C}\colon \mathbb{F}_2^k \to \mathbb{F}_2^n$ maps information words to codewords and the set of 2^k codewords forms a k-dimensional vector space of $\mathbb{F}_2^n$. Let $\mathcal{B} = \{b_0, b_1, \ldots, b_{k-1}\}$ be a basis for this subspace, $b_i \in \mathbb{F}_2^n$. Then the code $\mathcal{C}$ can be described by a generator matrix

$$G = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix}.$$

The code can equivalently be described by a parity check matrix $H \in \mathbb{F}_2^{n-k \times n}$ which is a generator matrix for the dual code $\mathcal{C}^\perp = \{x \in \mathbb{F}_2^n : \forall c \in \mathcal{C}, x \cdot c = 0\}$. Thus the following relationship holds: $HG^T = 0 \in \mathbb{F}_2^{k \times n-k}$. For any vector $y \in \mathbb{F}_2^n$, and parity check matrix H, the matrix-vector product $Hy^T = s \in \mathbb{F}_2^{n-k}$ is known as the syndrome. For any y such that $Hy^T = 0 \in \mathbb{F}_2^{n-k}$, y is a codeword (i.e., $y \in \mathcal{C}$).

A $v \times v$ circulant matrix is a square matrix such that each row r_{i+1} is one shift to the right of the previous row r_i for $i \in \{0, 1, \ldots, v-1\}$. The ring of $v \times v$ circulant matrices over $\mathbb{F}_2$ is isomorphic to the polynomial ring $\mathbb{F}_2[x]/\langle x^v + 1\rangle$. A quasi-cyclic (QC) matrix is a block sum of circulant matrices.

2.2 BIKE

Bit-flipping Key Encapsulation (BIKE) is a cryptosystem based on binary linear codes with quasi-cyclic structure and moderately sparse private keys [1]. The private key $H \in \mathbb{F}_2^{r \times 2r}$ is composed of two circulant blocks: H_0, H_1 of size $r \times r$ with r prime and such that $x^r - 1$ has only two irreducible factors modulo 2. The columns of H have weight d and the rows h_i of H are such that $|h_i| = w = 2d$ for all $i \in \{0, \ldots, r-1\}$. MDPC code parameters satisfy row weight $w \approx \sqrt{n}$ for n the length of the code.

At a high level, the public-key encryption system underlying the BIKE KEM is composed of three algorithms: key generation, encryption, and decryption. The key generation algorithm generates a private key $H = [H_0|H_1] \in \mathbb{F}_2^{r \times 2r}$ and public key H' is H in systematic form ($H' = H_0^{-1}H$). To encrypt a message m, a sender must encode m into a vector e of suitable weight t, then compute the syndrome $H'e^T = s$. The receiver decrypts by decoding the syndrome s using the secret key H and a predefined syndrome decoding algorithm. The recommended BIKE syndrome decoder as of the time of writing is the Black-Grey-Flip decoder [9].

Let λ denote the security parameter and let H denote a BIKE secret key. The security of BIKE depends on the inability of an attacker to break (variants of) the syndrome decoding problem(s). The best known attacks are information set decoding (ISD) algorithms, first introduced in 1962 by Prange [13] and later improved in dozens of works yielding small change in the overall asymptotic cost. (See [5,12,19] for a non-exhaustive list). Thus, for BIKE to achieve λ bits of security against the best known ISD attacks [7], the BIKE team determined that

$$\lambda \approx t - \frac{1}{2}\log_2 r \approx w - \log_2 r$$

where r denotes the circulant block size of H, w denotes the row weight of H, and t denotes the weight of the error vector in which a message is encoded [1].

2.3 Weak Keys and Near Codewords

For security level λ, the average decoding failure rate $\mathrm{DFR}_{\mathcal{D},\mathcal{H}}$ for an IND-CCA secure cryptosystem should be $\leq 2^{-\lambda}$ where $\mathcal{D}$ denotes the decoder and $\mathcal{H}$ the key space. A set $\mathcal{W} \subset \mathcal{H}$ of keys is said to be *weak* if:

$$\frac{|\mathcal{W}|}{|\mathcal{H}|}\mathrm{DFR}_{\mathcal{D},\mathcal{W}} > 2^{-\lambda} \geq \mathrm{DFR}_{\mathcal{D},\mathcal{H}}.$$

In [22, Chapter 15], Vasseur identifies three types of *weak keys* for the BIKE cryptosystem:

- **Type I:** keys with many consecutive nonzero bits in the rows of one of the cyclic blocks, first identified by [8].
- **Type II:** keys with nonzero bits at many regular intervals in the rows of one of the cyclic blocks.
- **Type III:** keys with many intersections between the columns of the two cyclic blocks.

It is known that some sets of vectors are more likely to cause decoding failures than on average. A (u, v)-*near codeword* for a parity-check matrix H is an error vector e with Hamming weight u whose syndrome $s = He^T$ has weight v [11]. When u, v are small, these near codewords can be likely to cause decoding failures [15]. Based on the structure of BIKE, Vasseur defines three sets with small u, v as follows:

- $\mathcal{C}$: vectors which form the rows of the generator matrix $G = [H_1^T | H_0^T]$; these are codewords of weight w for the secret key $H = [H_0 | H_1]$.
- $\mathcal{N}$: the set of (d, d)-near codewords of the form $(v_0, \mathbf{0})$ or $(\mathbf{0}, v_1)$, where $\mathbf{0} \in \mathbb{F}_2^r$ and v_i is a row of the circulant block H_i of the parity check matrix.
- $2\mathcal{N}$: the set of vectors formed by sums of two vectors in $\mathcal{N}$. Due to the small chance of cancellation, one may consider the set $2\mathcal{N}$ as $(w - \epsilon_0, w - \epsilon_1)$-near codewords for some small $\epsilon_i \geq 0, i \in \{0, 1\}$.

3 Methods

Cryptographically relevant DFRs are too low ($< 2^{-128}$) to directly measure; it is only possible to measure DFRs for smaller code sizes, then use extrapolation methods to estimate the DFR to larger parameters. Some examples of this approach can be found in [8,9,18]. In this ongoing work, we begin by analyzing the decoding behavior for BIKE parameter sets targeting 20 bits of security in several experiments.

Parameters were selected according to BIKE design principles with the maximum error weight t reduced to prevent any inadvertent increase in decoding failures. Initial selected parameters are as follows: $(r, w, t, \lambda) = (523, 30, 18, 20)$. Later we include $389 \leq r \leq 827$ for prime r such that $x^r - 1$ has only two irreducible factors modulo 2.

We use the Black-Grey-Flip (BGF) decoder in all experiments. We used the original threshold selection function, defined in section 2.5.1 of the BIKE v1.0 specification [2], to compute the bit-flip threshold for all instances. The affine threshold functions in the current version of BIKE are derived from this original threshold rule. We precomputed the values used in the threshold function and stored them in a hash table for ease of computation.

Vasseur identifies three classes of weak keys that impede decoding (see Sect. 2.3 for the definitions of these classes) and describes an algorithm for filtering out weak keys [22, Algorithm 15.3]. We implement this algorithm and use it to reject weak keys. The definition of weak key depends on a parameter T, which

Vasseur sets to 10 for BIKE parameters in the cryptographically relevant range ($\lambda \geq 128$). (Note that smaller values of T mean that more keys are excluded.)

We instead use $T = 3$ for the weak key threshold, the smallest value of T for which finding non-weak keys is feasible. This is justified by the following empirical observation: If we set $T = 4$, the decoding failure rate increases enormously; for example, an experiment with $(r, T) = (587, 4)$ observed a DFR on the order of 2^{-8}, compared to around 2^{-20} for $(r, T) = (587, 3)$. Thus, to measure the DFR for non-weak keys, we must set $T = 3$.

We use the Boston University Shared Computing Cluster [6], a heterogeneous Linux-based computing cluster with approximately 21000 cores, to run SageMath implementations of the BGF decoder [1,9] in all experiments. The experiments yielded a graph with both the waterfall and error floor regions for our parameter set in addition to many explicit examples of decoding failures that can be used for future analysis. All raw data and the decoder used for this paper are available at [3].

4 Average DFR over Full Message Space

We first compute an average DFR for all suitable block lengths r as follows. For r in Table 1, we sample a random key H, rejecting any *weak keys of types I, II, III* [22], a random vector e of weight t, compute $s = He^T$, run BGF decoder on input (H, s), and record the total number of failures. This procedure is run N times where N varies flexibly ($N \in \{10^3, 10^4, 10^5, 10^6, 10^7, 10^8\}$) to ensure there are enough decoding failures at each r for robust statistical analysis. In the waterfall region, fewer decoding trials were needed to get a statistically adequate number of decoding failures. As r increased, the number of trials needed increased. For $r > 587$, decoding failures were exceptionally sparse. Since these computations get quite expensive and the log-DFR rate was decreasing only linearly for $r > 587$, we chose not to continue increasing the number of trials. The error vectors tested in the DFR experiment all had weight 18. The results of this experiment are displayed in Table 1 and plotted with best fit curves in Fig. 1.

We define a decoding failure as any instance where, on input (H, s), where s is of the form $s = He^T$, the syndrome decoder output e' is such that $He'^T \neq s$ or $e' \neq e$. The experiment was also designed to record any decoding instances where $He'^T = s$ and $e' \neq e$, but none were discovered.

5 DFR on $\mathcal{A}_{t,\ell}(\mathcal{S})$ Sets

Vasseur identified and studied the influence of the proximity of error vectors to any $\mathcal{S} \in \{\mathcal{C}, \mathcal{N}, 2\mathcal{N}\}$, described in Sect. 2.3, on the DFR [22]. To quantify how close certain error vectors are to such a set $\mathcal{S} \in \{\mathcal{C}, \mathcal{N}, 2\mathcal{N}\}$, Vasseur introduces the set

$$\mathcal{A}_{t,\ell}(\mathcal{S}) = \{v \in \mathbb{F}_2^{2r} : |v \star c| = \ell \text{ for some } c \in \mathcal{S}\},$$

Table 1. Decoding failure rates for r-values such that $389 \leq r \leq 827$, r is prime, and $x^r - 1$ has only two irreducible factors modulo 2. The data was computed using the parameters and methods described above.

r	Decoding failures	Decoding trials	$\log_2$ (DFR)
389	939	10^3	−0.09
419	680	10^3	−0.56
421	652	10^3	−0.62
443	3289	10^4	−1.60
461	1172	10^4	−3.09
467	850	10^4	−3.56
491	1524	10^5	−6.04
509	380	10^5	−8.04
523	946	10^6	−10.05
541	164	10^6	−12.57
547	70	10^6	−13.80
557	177	10^7	−15.79
563	108	10^7	−16.50
587	128	10^8	−19.58
613	61	10^8	−20.64
619	60	10^8	−20.67
653	37	10^8	−21.37
659	35	10^8	−21.45
661	37	10^8	−21.37
677	24	10^8	−21.99
701	20	10^8	−22.25
757	8	10^8	−23.58
827	7	10^8	−23.77

where t is the error vector weight and ℓ is the number of overlaps with an element of $\mathcal{S}$. To convert ℓ to a distance, for $v \in A_{t,\ell}(\mathcal{S})$ we define

$$\delta(v) = |c| + t - 2\ell$$

where c is a vector in $\mathcal{S}$ with $|v \star c| = \ell$. For δ low (equivalently, ℓ high), decoding failures are extremely common; see Fig. 2 for evidence at the 20-bit security level.

It is natural to consider the extent to which $\mathcal{A}_{t,\ell}(\mathcal{S})$ for some ℓ and some $\mathcal{S} \in \{\mathcal{C}, \mathcal{N}, 2\mathcal{N}\}$ captures vectors which cause decoding failures. Our simulations indicate that it is extremely unlikely for a typical decoding failure vector to be in $\mathcal{A}_{t,\ell}(\mathcal{S})$ for any $\mathcal{S}$ with a high ℓ. We define the *max overlap* of a decoding failure vector v with a $\mathcal{A}_{t,\ell}(\mathcal{S})$ set for fixed $\mathcal{S}$ to be the largest value of ℓ for which $v \in A_{t,\ell}(\mathcal{S})$. Using experimental data from $r = 587, N = 10^8$ we recorded

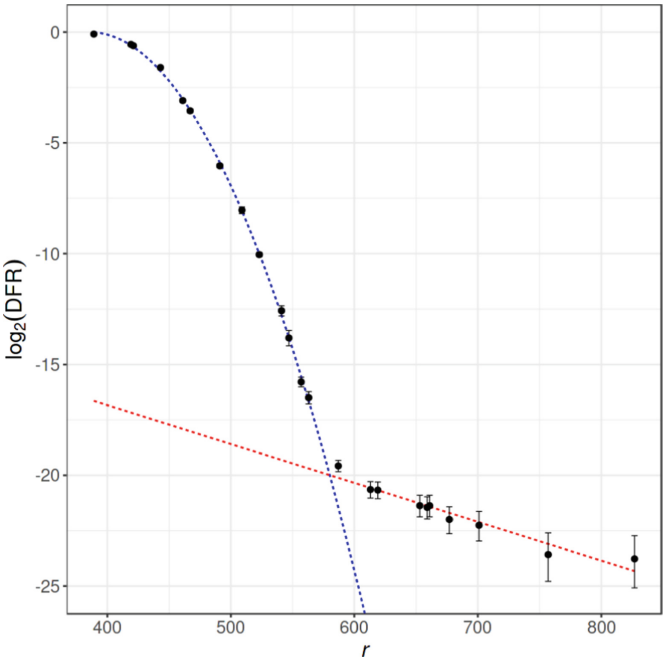


Fig. 1. Decoding failure rates as in Table 1 on a semi-log graph, with a quadratic best fit (blue) in the waterfall region $r < 587$ and a linear best fit (red) in the error floor region $r \geq 587$. (Color figure online)

128 total decoding failures and stored the 128 random error vectors that led to decoding failure. The relationship between these decoding failure vectors and the sets $\mathcal{S}$ is shown below; see Fig. 3a. We also repeated experiments for $r = 613$ and $r = 619$ with $N = 10^8$, recording 61 and 60 decoding failures, respectively. See Figs. 4 and 5 for this data.

Although the maximum value of ℓ is $t = 18$, the recorded values of ℓ never exceed 10. In fact, cases of $\ell = 10$ are quite rare. The values of ℓ recorded in experiments with vectors involved in decoding failures are greater than those of randomly sampled vectors, but it is expected that near-codewords and codewords of low weight overwhelmingly influence decoding failures in the error floor region [11]. From our results, it appears that only a minority of the error vectors producing a decoding failure are unusually close to a near-codeword or codeword of low weight. More analysis is needed to assess the relationships between the special sets $\mathcal{S}$ and decoding failures.

Notice that vectors close to a set $\mathcal{S}$ also have low syndrome weight; see Fig. 6. Moreover, as ℓ decreases, the syndrome weights approach the average.

From this, we are motivated to analyze to what extent syndrome weight predicts decoding failures.

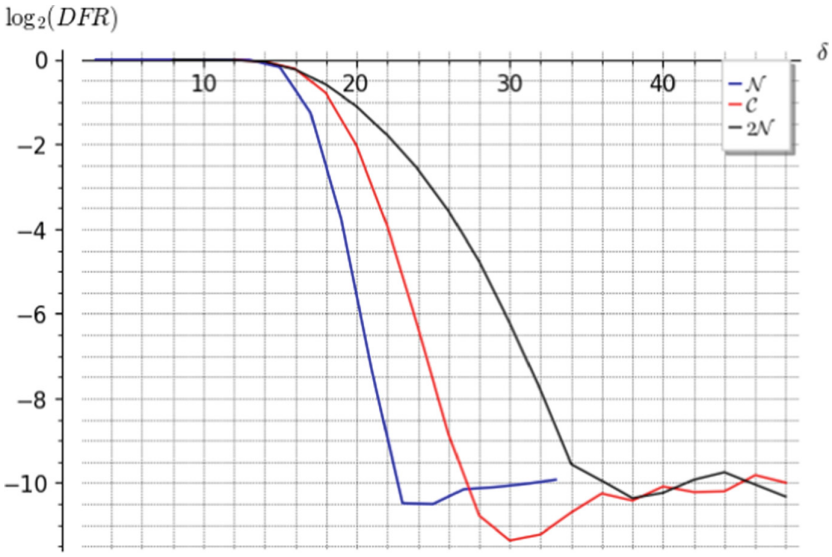


Fig. 2. 20-bit security DFR versus δ for near-codeword sets $\mathcal{C}, \mathcal{N}, 2\mathcal{N}$ for $r = 523$

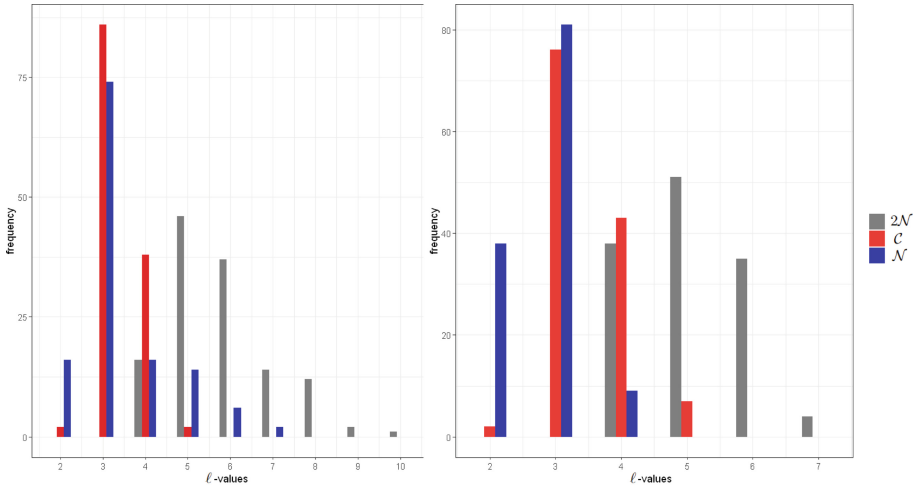
6 Distribution of Syndrome Weight

We investigate the syndrome weights of error vectors causing decoding failures and compare them with those of generic vectors.

Figure 7 and Fig. 8 are obtained by generating 10^3 instances of non-weak parity check matrices H, random error vectors e, and then we compute the average weight of their syndromes $s = He^T$. For the ones causing decoding failures, we extract the information from our DFR computations containing the corresponding parity check matrices and error vectors and then we compute the average weight of their syndromes.

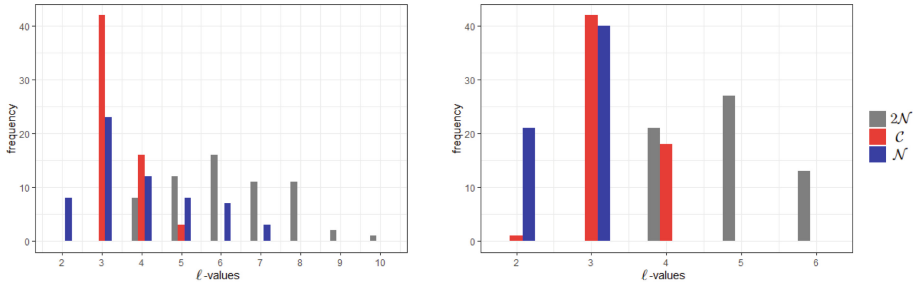
We observe that the syndrome weights of generic vectors tend to follow a normal distribution while the error vectors causing decoding failures have syndrome weights that are more concentrated around the mean, which we hypothesise to be lower than that of the generic vectors; see Fig. 8 for the case $r = 587$, where we compare the syndrome weights of the 128 vectors which caused decoding failures with the syndrome weights of the 10^5 randomly generated vectors of the same weight $t = 18$.

Figure 8 displays histograms of the syndrome weights of generic vectors and error vectors causing decoding failures for $r = 587$. Similarly, for the ten r values with $509 \leq r \leq 653$, we use data from the previous DFR computation and an additional 10^3 simulations of random error vectors to compare their syndrome weights. Using this data, we explore whether or not there is convincing evidence that the syndrome weights of error vectors causing decoding failures are lower than those of generic vectors. The null hypothesis is that there is no difference between the two groups in consideration while the alternative hypothesis is that



(a) Decoding failure vectors (b) Randomly generated vectors

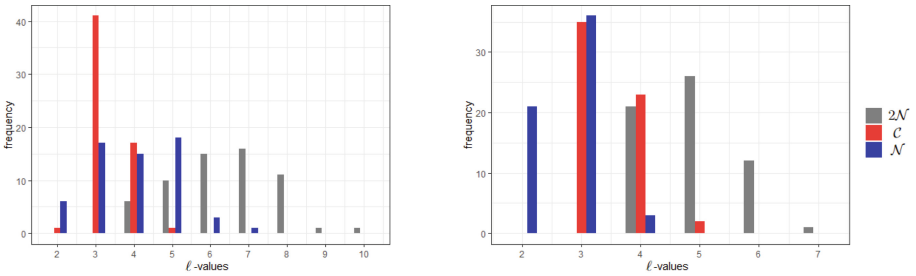
Fig. 3. For the 128 vectors v with $r = 587$, $d = 15$, $t = 18$ which caused decoding failures, we compute the distances from the sets $\mathcal{C}, \mathcal{N}, 2\mathcal{N}$ as measured by the maximum number of intersections with an element of these sets. Here, $\ell := |v \star c|$ for $c \in \mathcal{C}, \mathcal{N}, 2\mathcal{N}$. We do the same computation for 128 randomly generated vectors under the same parameters.



(a) Decoding failure vectors (b) Randomly generated vectors

Fig. 4. For the 61 vectors v with $r = 613$, $d = 15$, $t = 18$ which caused decoding failures, we compute the distances from the sets $\mathcal{C}, \mathcal{N}, 2\mathcal{N}$ as measured by the maximum number of intersections with an element of these sets. Here, $\ell := |v \star c|$ for $c \in \mathcal{C}, \mathcal{N}, 2\mathcal{N}$. We do the same computation for 61 randomly generated vectors under the same parameters.

the generic vectors have higher syndrome weights. Both data come from random, independent sampling and have data sets with more than 30 observations. The difference in sample means may be modeled using a t-distribution. For each r, one could compute the point estimates $m_{\text{generic}} - m_{\text{DF}}$ of population difference $\mu = \mu_{\text{generic}} - \mu_{\text{DF}}$ and standard errors of the point estimate



(a) Decoding failure vectors (b) Randomly generated vectors

Fig. 5. For the 60 vectors v with $r = 619$, $d = 15$, $t = 18$ which caused decoding failures, we compute the distances from the sets $\mathcal{C}, \mathcal{N}, 2\mathcal{N}$ as measured by the maximum number of intersections with an element of these sets. Here, $\ell := |v \star c|$ for $c \in \mathcal{C}, \mathcal{N}, 2\mathcal{N}$. We do the same computation for 60 randomly generated vectors under the same parameters.

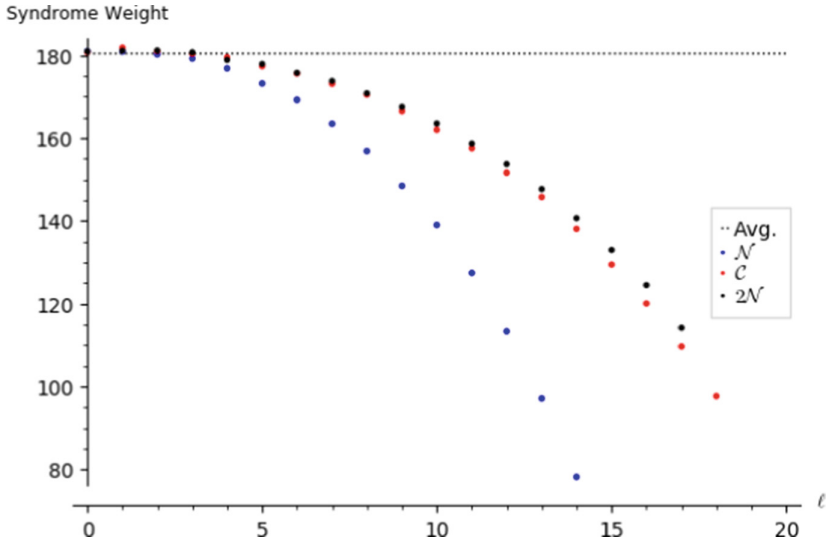


Fig. 6. Syndrome weight of error vectors in $\mathcal{A}_{t,\ell}(\mathcal{S})$ as ℓ (the maximum number of overlaps with an element of the set $\mathcal{S}$) varies, for $r = 587$, $t = 18$. Average syndrome weight for an error vector of weight $t = 18$ was approximately 180.712, plotted as the dotted horizontal line.

$$SE = \sqrt{\frac{\sigma^2_{\text{generic}}}{N_{\text{generic}}} + \frac{\sigma^2_{\text{DF}}}{N_{\text{DF}}}}.$$

With this information, one could compute the test statistic for this (one-tailed) test by the formula $T = \frac{\mu - 0}{SE}$. Using either a t-table or statistics software, we can find appropriate degrees of freedom and from there, the p-value, for each

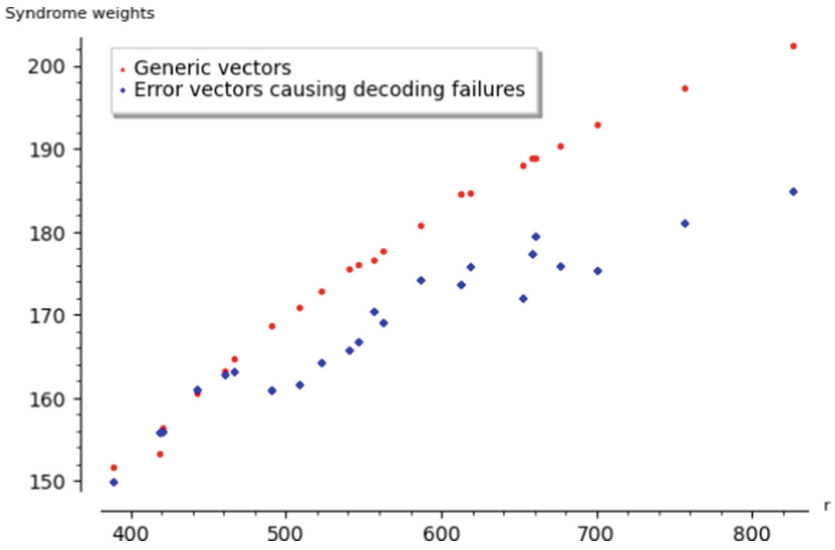
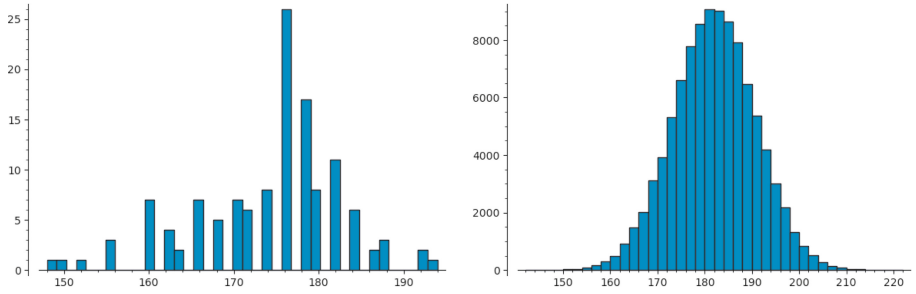


Fig. 7. Syndrome weights of random vectors with $t = 18$ (red circles) and vectors causing decoding failures (blue diamonds). (Color figure online)



(a) Decoding failure vectors

(b) Randomly generated vectors

Fig. 8. A comparison of syndrome weights for $r = 587$ between the 128 error vectors which were found to be involved in decoding failures and 10^5 random vectors. Vertical axis is frequency, and horizontal axis is syndrome weight.

r. Our conclusion is that for the sixteen r-values in the range $509 \leq r \leq 827$, the p-value is less than the significance value $\alpha = 0.01$, and therefore we reject the null hypothesis, i.e., syndrome weights of error vectors causing decoding failures are lower than those of generic vectors. A general summary of the test statistic values $m_{\text{generic}} - m_{\text{DF}}$ and the corresponding p-values can be found in Table 2.

Table 2. Hypothesis test results for $509 \leq r \leq 827$, with the corresponding test statistic values and p-values, indicating the vectors causing decoding failures do have lower syndrome weights than generic vectors for $509 \leq r \leq 701$, notably a selection of r-values where the waterfall region meets the error floor in the DFR graph of Fig. 1.

r	$m_{\text{generic}} - m_{\text{DF}}$	p
509	9.29	<0.00001
523	8.60	<0.00001
541	9.79	<0.00001
547	9.29	<0.00001
557	6.20	<0.00001
563	8.61	<0.00001
587	6.56	<0.00001
613	10.92	<0.00001
619	8.86	<0.00001
653	15.99	<0.00001
659	11.49	<0.00001
661	9.40	<0.00001
677	14.45	<0.00001
701	17.58	<0.00001
757	16.25	0.00278
827	17.53	0.00002

7 Conclusion

In order to claim IND-CCA2 security with confidence for the proposed parameter sets of the BIKE cryptosystem, it is necessary to demonstrate that the BIKE decoder fails with cryptographically low probability on honestly generated ciphertexts. Such a low decoding failure rate cannot be directly measured, but is instead estimated by extrapolation from parameters with directly measurable decoding failure rates. In order for this analysis to be accurate, one must account for error floor behavior.

In our analysis of the BIKE cryptosystem at the 20-bit security level, we find that vectors which cause decoding failures have lower than average syndrome weight. However, identifying where these low syndrome weight vectors come from is still an open question. In [22,23], Vasseur proposes three classes of low syndrome weight vectors: $\mathcal{C}$, $\mathcal{N}$, and $2\mathcal{N}$. Vasseur also describes sets $\mathcal{A}_{t,\ell}(S)$ of vectors which are close to the sets $S \in \{\mathcal{C}, \mathcal{N}, 2\mathcal{N}\}$. In our work, while we do find that Vasseur's sets do contain many vectors that cause decoding failures, we do not find that these classes of vectors are responsible for the bulk of the decoding failures.

It therefore remains for future work to identify further classes of error vectors that might account for the observed decoding failures in our experiments.

If these can be identified it may be possible to predict error floor behavior for larger parameters, and thereby identify parameter sets that have a sufficiently low decoding failure rate to be used for IND-CCA2 security in the BIKE cryptosystem.

Acknowledgements. We would like to thank Valentin Vasseur for helpful discussions and code for reproducing experimental data, Paolo Santini for providing us with an initial SageMath implementation of the BGF decoder, and the anonymous reviewers for helpful feedback and suggestions.

This collaboration was initiated during the Rethinking Number Theory 2 (RNT2) Workshop. Funding for RNT2 came from the Number Theory Foundation and the University of Wisconsin-Eau Claire Department of Mathematics. This work was supported in part by the Simons Collaboration on Arithmetic Geometry, Number Theory, and Computation (Simons Foundation grant #550023).

References

1. Aragon, N., et al.: BIKE: bit flipping key encapsulation - spec v4.2 (2021). https://bikesuite.org/files/v4.2/BIKE_Spec.2021.07.26.1.pdf
2. Aragon, N., et al.: BIKE: bit flipping key encapsulation - spec v1.0 (2017). https://bikesuite.org/files/BIKE.2017.11.30.pdf
3. Arpin, S., Billingsley, T.R., Hast, D.R., Lau, J.B., Perlner, R., Robinson, A.: Raw data and decoder for the paper "A study of error floor behavior in QC-MDPC codes". https://github.com/HastD/BIKE-error-floor. Accessed 23 May 2022
4. Baldi, M.: QC-LDPC Code-Based Cryptography. SECE, Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02556-8
5. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: how $1 + 1 = 0$ improves information set decoding. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 520–536. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_31
6. Boston University Shared Computing Cluster. https://www.bu.edu/tech/support/research/computing-resources/scc/. Accessed 18 Feb 2022
7. Canto Torres, R., Sendrier, N.: Analysis of information set decoding for a sub-linear error weight. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 144–161. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_10
8. Drucker, N., Gueron, S., Kostic, D.: On constant-time QC-MDPC decoders with negligible failure rate. In: Baldi, M., Persichetti, E., Santini, P. (eds.) CBCrypto 2020. LNCS, vol. 12087, pp. 50–79. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-54074-6_4
9. Drucker, N., Gueron, S., Kostic, D.: QC-MDPC decoders with several shades of gray. In: Ding, J., Tillich, J.-P. (eds.) PQCrypto 2020. LNCS, vol. 12100, pp. 35–50. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44223-1_3
10. Guo, Q., Johansson, T., Stankovski, P.: A key recovery attack on MDPC with CCA security using decoding errors. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 789–815. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_29
11. MacKay, D.J.C., Postol, M.S.: Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes. Electron. Notes Theor. Comput. Sci. **74**, 97–104 (2003). MFCSIT 2002, The Second Irish Conference on the Mathematical Foundations of Computer Science and Information Technology

12. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 107–124. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_6
13. Prange, E.: The use of information sets in decoding cyclic codes. IRE Trans. Inf. Theory **8**(5), 5–9 (1962)
14. Price, A., Hall, J.: A survey on trapping sets and stopping sets. arXiv e-prints (2017)
15. Richardson, T.: Error floors of LDPC codes. In: Proceedings of the 41st Annual Allerton Conference on Communication, Control, and Computing, pp. 1426–1435 (2003)
16. Richter, G.: Finding small stopping sets in the Tanner graphs of LDPC codes. In: 4th International Symposium on Turbo Codes and Related Topics, pp. 1–5 (2006)
17. Sendrier, N., Vasseur, V.: About low DFR for QC-MDPC decoding. Cryptology ePrint Archive, Paper 2019/1434 (2019). https://eprint.iacr.org/2019/1434
18. Sendrier, N., Vasseur, V.: On the decoding failure rate of QC-MDPC bit-flipping decoders. In: Ding, J., Steinwandt, R. (eds.) PQCrypto 2019. LNCS, vol. 11505, pp. 404–416. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_22
19. Stern, J.: A method for finding codewords of small weight. In: Cohen, G., Wolfmann, J. (eds.) Coding Theory 1988. LNCS, vol. 388, pp. 106–113. Springer, Heidelberg (1989). https://doi.org/10.1007/BFb0019850
20. Tillich, J.-P.: The decoding failure probability of MDPC codes. In: 2018 IEEE International Symposium on Information Theory (ISIT), pp. 941–945. IEEE (2018)
21. Vasić, B., Chilappagari, S.K., Nguyen, D.V.: Failures and error floors of iterative decoders, chapter 6. In: Declerq, D., Fossorier, M., Biglieri, E. (eds.) Academic Press Library in Mobile and Wireless Communications, pp. 299–341. Academic Press, Oxford (2014)
22. Vasseur, V.: Post-quantum cryptography: a study of the decoding of QC-MDPC codes. Ph.D. thesis, Université de Paris (2021)
23. Vasseur, V.: QC-MDPC codes DFR and the IND-CCA security of BIKE. Cryptology ePrint Archive, Paper 2021/1458 (2021). https://eprint.iacr.org/2021/1458
24. Wang, C.-C., Kulkarni, S.R., Vincent Poor, H.: Finding all small error-prone substructures in LDPC codes. IEEE Trans. Inform. Theory **55**(5), 1976–1999 (2009)