A graph-theoretic approach to analyzing decoding failures of BIKE*

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

[1] Virginia Tech, Department of Mathematics***
[2] Rose-Hulman Institute of Technology, Department of Mathematics
[3] Boston University, Department of Mathematics & Statistics
[4] National Institute of Standards and Technology, Computer Security Division

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 using graph-theoretic approaches. We select parameters according to BIKE design principles and conduct a series of experiments using Rust to generate significantly more decoding failure instances than in prior work using SageMath. For each decoding failure, we study the internal state of the decoder at each iteration and find that for 97% of decoding failures at block size $r = 587$, the decoder reaches a fixed point within 7 iterations. We then consider the corresponding Tanner graphs of each decoding failure instance to determine whether the decoding failures are due to absorbing sets. We find that 81% of decoding failures at $r = 587$ were caused by absorbing sets, and of these the majority were (d, d)-near codewords.

1 Introduction

Bit-flipping Key Encapsulation (BIKE) is a code-based cryptosystem based on quasi-cyclic moderate density parity check codes (QC-MDPC). BIKE is one of three fourth-round finalists still under consideration in the NIST Post-Quantum Standardization process. There is no closed-form analysis of the BIKE DFR, but several works exist which use extrapolation techniques to estimate the DFR for bits of security $\lambda \geq 128$. However, one must consider the phenomenon known as the error floor region of DFR curves to avoid an underestimate of DFR for larger code sizes. In this work, we build on prior work which identifies the error floor for BIKE at $\lambda = 20$ and use a graph-theoretic approach to identify the factors influencing error-floor behavior.

Let $C(n, k)$ be a binary $[n, k]$ linear code with length n, dimension k, represented by parity-check matrix H. For a parity check matrix $H_{k\times n}$, there is a corresponding bipartite graph, known as a Tanner graph, consisting of variable nodes v_i and check nodes c_i where each column h_i, $1 \leq i \leq n$, of H is represented by a variable node, each row x_j, $1 \leq j \leq k$, of H is represented by a check node, and an edge connects variable node v_i and check node c_j if entry $x_{j,i}$ in H is 1.

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 error floor regions of LDPC codes under iterative decoders focuses on properties of Tanner graph representations of the code [2–4], such as identifying *stopping sets*, *trapping sets*, and *absorbing sets*. It is said that the LDPC Tanner graph analyses cannot extend to MDPC codes because the graphs are too dense [2, 5]. Fortunately, at the 20-bit security level it is feasible generate several instances of decoding failures and construct the corresponding Tanner graphs and subgraphs. In this ongoing work, we study relationships between known classes of QC-MDPC matrices that contribute to the DFR [5] and the corresponding Tanner graphs.

*** work was completed while this author was at Universiteit Leiden, Mathematics Institute

* A version of this extended abstract was submitted to PQCrypto 2023 and withdrawn pending further work. We have added only figures and brief clarifying points. (Date: October 23, 2024)

2 QC-MDPC Codes: Experiments and theory

In previous work [1], we used BIKE design principles to generate BIKE DFR data at the 20-bit security level to both identify and analyze the DFR curve error floor. We found that the DFR curve transitioned from waterfall to error floor region around $r = 587$. In this work, we re-run the same experiments using Rust, instead of SageMath, yielding significantly more decoding failure instances for the same DFR. In the following analysis, we focus on the block-size $r = 587$, column weight $d = 15$, and error weight $t = 18$. We are continuing to characterize decoding failures, this time using the language of absorbing sets and comparing this notion with known weaknesses for QC-MDPC codes under iterative decoders.

Definition 1 ((a, b)-absorbing set). *An (a, b)-absorbing set D is a subset of variable nodes of the Tanner graph where the subgraph G_D of the Tanner graph containing the vertices in D and their check node neighbors satisfies the following properties: $|D| = a$, the subset $O(D)$ of check nodes of odd-degree in the subgraph G_D is size b, and each variable node in D has strictly fewer neighbors of odd-degree than of even-degree.*

We compare (d, d)-absorbing sets with (d, d)-near codewords and provide heuristic evidence that the two sets have significant overlap.

2.1 Weak keys and absorbing sets

The supports of distances between the input error and the falsely outputted errors $(e_{in} - e_{out})$ were more likely to be absorbing sets when weak keys were not filtered out. Weak keys are moderate density parity check matrices with properties that make them vulnerable to decoding failure (see [5] for description). Our data shows that the decoding failures associated with these weak keys are less likely to correspond to absorbing sets than decoding failures coming from non-weak keys.

For $r = 587$, we collected 557 decoding failures with filtering to remove weak keys; 442 of these (79.4%) were absorbing sets, including 415 that were $(15, 15)$-absorbing sets (all other (a, b) parameters occurred only once or twice). See colum 1 of Table 1. For $r = 587$, we collected 1980 decoding failures without filtering out weak keys; 1184 of these (59.8%) were absorbing sets, including 705 that were $(15, 15)$-absorbing sets. In this ongoing work, we seek to distinguish the decoding behavior that leads to failure in the weak-key vs. non-weak-key setting as it relates to absorbing sets.

2.2 (d, d)-near codewords and absorbing sets

A (u, v)-near codeword is a weight u vector with syndrome weight v. The family of (d, d)-near codewords is known to cause decoding failures for BIKE [1], [5, Def. 16.3]. To study the impact of this phenomena on the DFR waterfall more closely, for $r = 587$ we fixed a single non-weak parity check matrix H and generated 5546 decoding failures, of which 4498 were (a, b)-absorbing. In 4389 of these absorbing cases, $e_{in} - e_{out}$ is a $(15, 15)$-absorbing set, and all of these cases were also $(15, 15)$-near codewords. We briefly justify this:

From $u = 15$, we know the $(15, 15)$-near codewords correspond to subgraphs with 15 variable nodes and by $d = 15$, each variable node has 15 neighbors. To better determine whether these near codewords should be absorbing sets, we must determine how many neighbors of each variable node have odd-degree. Out of all check nodes in the subgraphs, there are only 15 of odd-degree by $v = 15$. We use the maximum number of unsatisfied parity check equations for each decoding failure instance ("maxupc") and determine find that for all $(15, 15)$-near codewords, maxupc $\in \{3, 7\}$. We conclude that for each variable node, at most 7 of its neighbors have odd-degree. We thus hypothesize that all $(15, 15)$-near codewords are $(15, 15)$-absorbing sets.

We are continuing to analyze the absorbing sets that are not (d, d)-near codewords. In particular, working with our $r = 587$ fixed key data we found that the decoder got trapped into a cycle of length-1 in all of the (a, b)-absorbing decoding failure cases, except for one. Even the non-absorbing decoding failures overwhelmingly ended in a decoder cycle of length 1 (904 of the 1048 non-absorbing decoding failures observed). Itemizing the (a, b)-absorbing sets at the 20-bit security level is one step towards characterizing the error floor behavior for BIKE at $\lambda \geq 128$-bits of security.

$r = 587$, 10^{10} random keys; 557 decoding failures; 438 (a, b)-absorbing.		$r = 587$, fixed key, 10^{10} trials; 5546 decoding failures; 4498 (a, b)-absorbing.		$r = 827$, fixed key, 10^{10} trials; 214 decoding failures; 213 (a, b)-absorbing.	
(a, b)	Frequency	(a, b)	Frequency	(a, b)	Frequency
(15, 15)	415	(15, 15)	4389	(15, 15)	213
(7, 47)	2	(16, 96)	6		
(8, 48)	2	(16, 90)	4		
(8, 54)	2	(15, 85)	3		
		(13, 39)	3		
		(19, 97)	3		

Table 1: Experimental data on decoding failure (a, b)-absorbing structure.

Acknowledgements. We thank Christine Kelley for fruitful discussions concerning absorbing sets.

References

1. Sarah Arpin, Tyler Raven Billingsley, Daniel Rayor Hast, Jun Bo Lau, Ray Perlner, and Angela Robinson. A study of error floor behavior in qc-mdpc codes. In Jung Hee Cheon and Thomas Johansson, editors, *Post-Quantum Cryptography*, pages 89–103, Cham, 2022. Springer International Publishing.
2. Marco Baldi. *QC-LDPC Code-Based Cryptography.* SpringerBriefs in Electrical and Computer Engineering. Springer, Cham, 01 2014.
3. Emily McMillon, Allison Beemer, and Christine A. Kelley. Analysis of absorbing sets using cosets and syndromes. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 367–372, 2020.
4. Tom Richardson. Error floors of LDPC codes. In *Proc. 41st Annual Allerton Conf. on Communication, Control, and Computing*, pages 1426–1435, 01 2003.
5. Valentin Vasseur. *Post-quantum cryptography: a study of the decoding of QC-MDPC codes.* PhD thesis, Université de Paris, Mar 2021.