

# Analyzing the decoding failure rate of QC-MDPC Codes

Sarah Arpin and Jean-Pierre Tillich

National Institute of Standards and Technology PQC Seminar

January 7, 2025

joint work with:

Jun Bo Lau, Ray Perlner, Angela Robinson, and Valentin Vasseur.

# BIKE<sup>2</sup>

**BIKE** (**Bi**t-Flipping **K**ey **E**ncapsulation) is a code-based KEM (key encapsulation mechanism) based on QC-MDPC (Quasi-Cyclic Moderate-Density Parity-Check) codes.

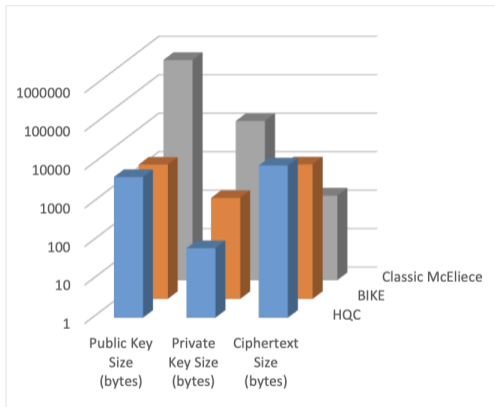
- ▶ Relevance to post-quantum cryptography
  - ▶ One of three remaining KEMs in round 4 of the NIST PQC Standardization process.
- ▶ IND-CCA security
  - ▶ The GJS<sup>1</sup> key-recovery attack exploits decoding failures in an IND-CCA security model.
  - ▶ Decoding failure rate (DFR) of a code-based KEM that claims IND-CCA security must be sufficiently low to prevent GJS attack.

---

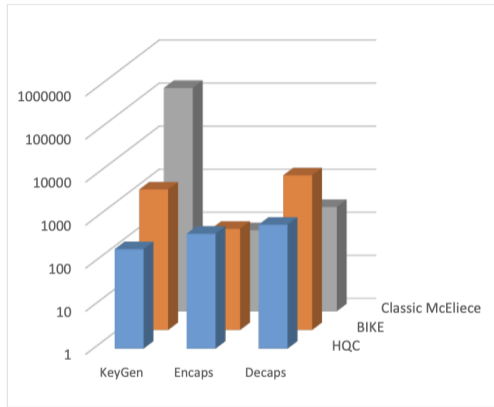
<sup>1</sup>*A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors*, Qian Guo, Thomas Johansson, and Paul Stankovski (2016).

<sup>2</sup>BIKE: Bit flipping key encapsulation - <https://bikesuite.org>

# How does BIKE compare to other round 4 KEMs?<sup>3</sup>



**Figure 2:** Results of comparing the cryptographic indicators for the L3 level



**Figure 5:** Performance comparison results for L3 stability level (kilocycles)

<sup>3</sup> *Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece* Oleksandr Kuznetsov, Sergey Kandiy, Emanuele Frontoni, and Oleksii Smirnov (2023).

## QC-MDPC codes

- ▶ A **Moderate-Density Parity-Check** code (MDPC code) is a binary  $[n, k]$  linear code that has a parity check matrix  $H$  such that each row has weight  $w \approx \sqrt{n}$ .
- ▶ A **circulant matrix** is a matrix in which each column is obtained by shifting the previous column down one step. For example:

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

- ▶ A **QC-MDPC** code is a MDPC code with a parity check matrix composed of circulant blocks.

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

The code  $\mathcal{C}$  is the nullspace of the matrix  $H$ .

## Code-based cryptosystem: Niederreiter (1986)

**Syndrome Decoding Problem:** Given a parity check matrix  $H$ , a target weight  $t$ , and a syndrome  $s := He^T$ , find a vector  $e'$  of weight  $|e'| \leq t$  such that  $H(e')^T = s$ .

---

$\mathcal{C}$ : a linear code with an efficient syndrome decoding algorithm  $\mathcal{A}$  which can correct  $t$  or fewer errors.

- ▶  $H'$ : a  $r \times n$  parity check matrix for the code  $\mathcal{C}$  (private),
- ▶  $S$ : a  $r \times r$  invertible matrix (private),
- ▶  $P$ : an  $n \times n$  permutation matrix (private),
- ▶  $H := SH'P$  is the public key.

**Encrypt:** A message is converted to a binary string  $m$  of weight  $|m| \leq t$ . The ciphertext is  $s := Hm^T$ .

**Decrypt:**

Remove  $S$ :  $S^{-1}s = H'Pm^T$ .

Apply the decoding algorithm:  $\mathcal{A}(H'Pm^T) = Pm^T$ .

Extract  $m$ :  $P^{-1}(Pm^T) = m^T$ .

# BIKE at a high level

- ▶ Parity check matrix  $H$  is quasi-cyclic, meaning  $H = [H_0|H_1]$  is composed of two circulant blocks.
- ▶ Message encoded as error vector  $e \in \mathbb{F}_2^{2r}$  of weight  $t$ .
- ▶ Ciphertext is syndrome  $s = He^T \in \mathbb{F}_2^r$ .
- ▶ Decrypt using Black-Grey-Flip (BGF) syndrome decoder.<sup>4</sup>

BGF syndrome decoding is not deterministic: there is the possibility for a decoding failure.

## Parameters

$r$ : block length,  $n := 2r$

$w$ : row weight of secret key

$t$ : maximum error weight

$\lambda$ : security parameter

## Design principles

$r$  prime

$x^r - 1$  has only two irreducible factors

$w \approx t \approx \sqrt{n}$

$w = 2d, d$  odd

Security parameter  $\lambda \approx t - \frac{1}{2} \log_2 r$ . Want:  $DFR < 2^{-\lambda}$ .

---

<sup>4</sup>The BGF decoder: *QC-MDPC decoders with several shades of gray*, Drucker–Gueron–Kostic

## Polynomial representation

An  $r \times r$  circulant matrix can be represented as a polynomial in  $\mathbb{F}_2[x]/(x^r - 1)$ : Let  $h = (h_0, \dots, h_{r-1})^T$  denote the first column of a circulant matrix. Polynomial rep.:

$$h(x) = \sum_{i=0}^{r-1} h_i x^i.$$

Since BIKE is composed of *two* circulant blocks  $H = [H_0|H_1]$ , we represent a BIKE **parity check matrix**  $H$  by  $H = (h_0(x), h_1(x)) \in (\mathbb{F}_2[x]/(x^r - 1))^2$ .

**Error** vector  $e$ , write  $e = (e_0(x), e_1(x))$ . **Syndrome**  $s(x) = e_0(x) \cdot h_0(x) + e_1(x) \cdot h_1(x)$ .

### Example

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad e = (0, 0, 0, 0, 0, 1)$$

$H = (h_0(x), h_1(x)) = (1, 1 + x)$ ,  $e(x) = (0, x^2) \in (\mathbb{F}_2[x]/(x^3 - 1))^2$ .

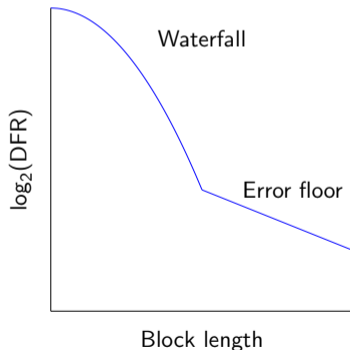
$s = He^T = (1, 0, 1)^T$ , and  $s(x) = 0 \cdot 1 + (1 + x) \cdot x^2 = 0 + x^2 + x^3 = 1 + x^2$ .

# What is an error floor?

Graphs of DFRs on a log scale for low- to moderate-density parity check codes with iterative decoders display a phenomenon:

- ▶ Initial, rapid decrease of decoding failures (**waterfall region**)
- ▶ Eventual plateau, more linear decrease (**error floor region**)

To accurately predict the DFR for higher code length (signal-to-noise ratio), one must account for the error floor region.



# LDPC code approach

Represent code in Tanner graph form:

- ▶ Sparse bipartite graph.
- ▶ Results on minimum distance based on girth (length of shortest cycle).
- ▶ Prevalence of small, closed loops increase probability of decoding failure.

## Definition

Let  $H$  be a parity-check matrix describing a code  $C$ . A  $(u, v)$ -**near codeword** is an error vector  $e$  of weight  $u$  whose syndrome  $s = He^T$  has weight  $v$ .

McKay, Postol (2003): near codewords with small  $u, v$  and low-weight codewords cause high error floor for certain LDPC codes.

Marco Baldi. QC-LDPC Code-Based Cryptography (2014)

David J.C. MacKay, Michael S. Postol. Weaknesses of Margulis & Ramanujan-Margulis Low-Density Parity-Check Codes (2003)

Tom Richardson. Error floors of LDPC codes (2003)

Gerd Richter. Finding small stopping sets in the Tanner graphs of LDPC codes (2006)

## Syndrome Decoding: Step-by-step

The BGF decoder used by BIKE is complicated enough to make explicit analysis challenging. Step-by-step is a simpler variant for analysis.

**Input:** A parity check matrix  $H$  and a syndrome vector  $s$ .

**Output:** An error pattern  $e$  satisfying  $He^T = s$ .

Initialize:  $e = 0$ ,  $s' = s$ .

While  $s' \neq 0$ :

    Compute threshold  $T := T(s')$ .

    Sample a random column  $h_j$  of  $H$ , with  $j \in \{0, 1, \dots, n-1\}$ .

    If  $|h_j \star s'| \geq T$ , then: Flip bit  $j$  of  $e$  and set  $s' = s' + h_j$ .

Once  $s' = 0$ , **return**  $e$ .

The threshold computation function determines how large  $|h_j \star s'|$  should be in order for a flip of  $e_j$  to decrease syndrome weight.

## Example: Syndrome Decoding

Given:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad s = (1, 0, 1)^T$$

- ▶ Compute the threshold function  $T = 2$ .
- ▶ Randomly select  $j = 5$ , and note  $h_5 \cdot s = 2 \geq T = 2$ .
- ▶  $e = (0, 0, 0, 0, 0, 1)^T$ .
- ▶  $s' = (1, 0, 1)^T + (1, 0, 1)^T = \mathbf{0}$
- ▶ Return  $e = (0, 0, 0, 0, 0, 1)^T$ .

## Near codewords

Let  $H = [H_0|H_1]$  have polynomial representation  $(h_0(x), h_1(x))$ .

The set of **near codewords** is:

$$\mathcal{N} := \{(x^s h_0(x), 0) : s \in \{0, 1, \dots, r-1\}\} \cup \{(0, x^s h_1(x)) : s \in \{0, 1, \dots, r-1\}\} \subseteq \mathbb{F}_2^n.$$

Near codewords are weight  $d$  and have syndrome weight  $d$ :

$$s(x) = x^s h_0(x) \cdot h_0(x) = x^s (h_0(x))^2 = x^s h_0(x^2),$$

since we are squaring in  $\mathbb{F}_2[x]/(x^r - 1)$ .

Near codewords yield **lower than expected syndrome weight**.

## Overlap with a near codeword

Near codewords, and error vectors with high overlap with near codewords, are difficult to decode because they result in unusually low syndrome weight:

### Example

Fix  $r = 7$ ,  $\mathbb{F}_2[x]/(x^7 - 1)$ . Let  $H = (h_0, h_1)$  with  $h_0(x) = 1 + x^2$  and  $h_1(x) = x^2 + x^3$ .  
 $e(x) = e_0(x) \oplus e_1(x)$ , for  $e_0(x) = x + x^3$  and  $e_1(x) = x^5$ .

Naively,  $|e_0(x)| = 2$ ,  $|h_0(x)| = 2$ ,  $|e_1(x)| = 1$ , and  $|h_1(x)| = 2$ , so one might expect  $|s(x)| = |e_0(x)| \cdot |h_0(x)| + |e_1(x)| \cdot |h_1(x)| = 2 \cdot 2 + 1 \cdot 2 = 6$ .

However, since  $e_0(x) = x \cdot h_0(x)$  this vector will have *forced cancellations*:

$$\begin{aligned} s(x) &= x \cdot h_0(x) \cdot h_0(x) + e_1(x) \cdot h_1(x) \\ &= x \cdot (1 + x^2) \cdot (1 + x^2) + x^5 \cdot (x^2 + x^3) \\ &= x \cdot (1 + \mathbf{2x^2} + x^4) + 1 + x = x + x^5 + 1 + x = 1 + x^5. \end{aligned} \tag{1}$$

The syndrome weight is  $|s(x)| = 2$ , which is lower than expected. ( $-2$ ) is due to a forced cancellation, from the overlap with the near codeword  $h_0(x)$ .

# BIKE at Small Parameters<sup>5</sup>

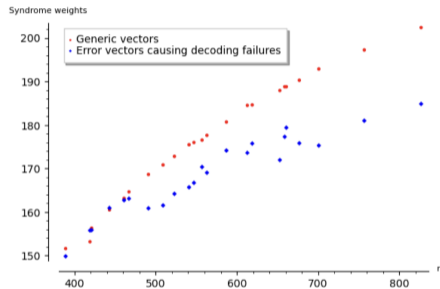


Fig. 7: Syndrome weights of random vectors with  $t = 18$  (red circles) and vectors causing decoding failures (blue diamonds).

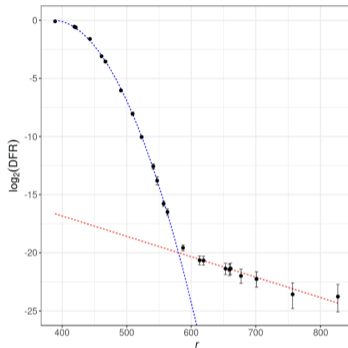


Fig. 1: Decoding failure rates as in Table 1 on a semi-log graph, with a quadratic best fit (blue) in the waterfall region  $r < 587$  and a linear best fit (red) in the error floor region  $r \geq 587$ .

## How can we get closer to an analysis of BIKE decoding failures?

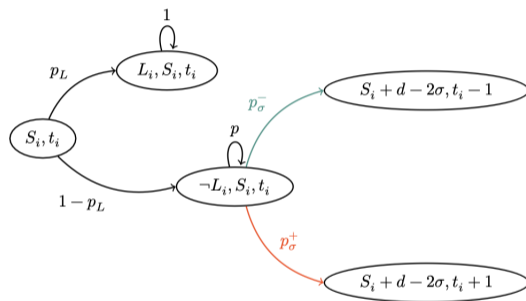
<sup>5</sup>A Study of Error Floor Behavior in QC-MDPC Codes A-, Tyler Raven Billingsley, Daniel Rayor Hast, Jun Bo Lau, Ray Perlner, and Angela Robinson (2022)

How can we model the DFR of BIKE with cryptographic parameters?

## Markov Approach: Previous work<sup>6</sup>

State space:  $(S, t)$  where  $S = |s|$  and  $t = |e|$  through the decoding process.

$L$ : blocked state.



- Does not account for the **effect of near codewords on DFR**.

<sup>6</sup>On the Decoding Failure Rate of QC-MDPC Bit-Flipping Decoders, Nicolas Sendrier and Valentin Vasseur (2018). Figure: *Post-quantum cryptography: a study of the decoding of QC-MDPC codes*, Valentin Vasseur PhD thesis (2021).

# Why are near codewords relevant here?

No near codeword:

- ▶ counter  $\sigma$  of a **correct bit**  $\sigma \sim \text{Bin}(d, \pi_0)$
- ▶ counter  $\sigma$  of a **normal bit in error**  $\sigma \sim \text{Bin}(d, \pi_1)$  with  $\pi_0 < \pi_1$

With  $u$  bits in error also in a near codeword:

- ▶  $u$  **bad bits**: in error + in near codeword
- ▶  $d - u$  **suspicious bits**: not in error + in near codeword
- ▶ counter  $\sigma$  of a **correct bit**  $\sigma \sim \text{Bin}(d, \pi_0)$
- ▶ counter  $\sigma$  of a **normal bit in error**  $\sigma \sim \text{Bin}(d, \pi_1)$  with  $\pi_0 < \pi_1$
- ▶ counter  $\sigma$  of a **bad bits**:  $\sigma \sim \text{Bin}(u - 1, \pi_0) + \text{Bin}(d - u + 1, \pi_1)$
- ▶ counter  $\sigma$  of a **suspicious bits**:  $\sigma \sim \text{Bin}(u, \pi_1) + \text{Bin}(d - u, \pi_0)$



In this work, we present two progressively more accurate Markov models to model the DFR of BIKE:

- (1) Model the effect of a **single, fixed near codeword** on the DFR and amplify the effect of this single near codeword on the decoding process in a post-computation to estimate the true DFR.
- (2) Assume that the nearest near codeword at the start is the only near codeword which affects the decoding process, and use the **structure of the parity-check matrix** to estimate the DFR.

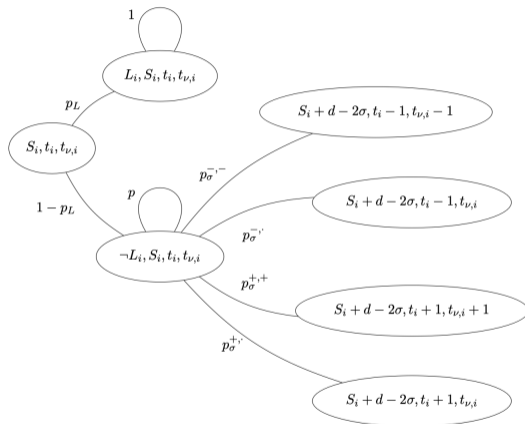
# Markov Approach 1: Adding the effect of near codewords<sup>7</sup>

Fix a near codeword  $\nu$ .

$(S_i, t_i, t_{\nu,i})$  = state at iteration  $i$  of decoder.

$t_{\nu}$  keeps track of overlaps with a near codeword  $\nu$ .

$L$  = blocked state.



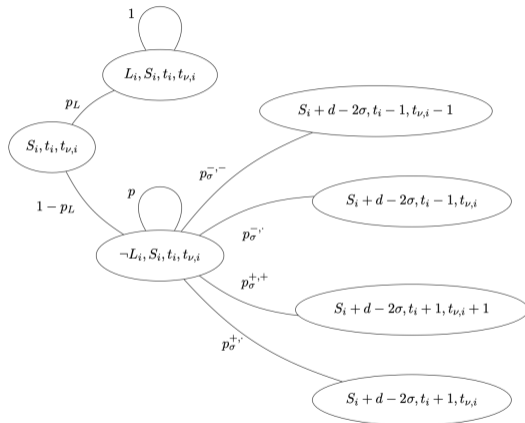
<sup>7</sup>Error floor prediction with Markov Models, A-, Jun Bo Lau, Ray Perlner, Angela Robinson, Jean-Pierre Tillich, and Valentin Vasseur. 2025\*.

# Markov Approach 1: Adding the effect of near codewords<sup>8</sup>

Counters  $\sigma$  model parity checks.

Bits are classified:

- ▶  $u$  bad bits: both error &  $\nu$
- ▶  $d - u$  sus bits: not error, in  $\nu$
- ▶  $t - u$  norm bits: error, not  $\nu$
- ▶ good bits: not error, not  $\nu$



<sup>8</sup>Error floor prediction with Markov Models, A-, Jun Bo Lau, Ray Perlner, Angela Robinson, Jean-Pierre Tillich, and Valentin Vasseur. 2025\*.

# Markov Approach 1: Adding the effect of near codewords

Model M1: With the effect of  $\nu$

Model estimating the DFR by following the values  $(S_i, t_i, t_{\nu,i})$  through the Markov model, where  $e$  is a random error pattern and  $\nu$  is a random near codeword.

Model M0: Without the effect of  $\nu$

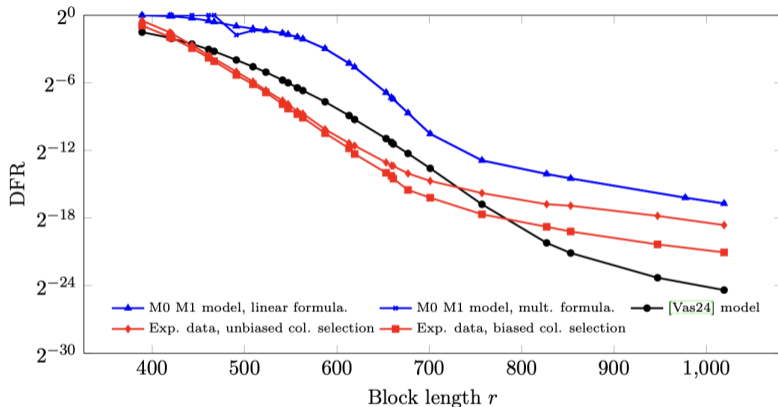
Model estimating the DFR by following the values  $(S_i, t_i, t_{\nu,i})$  through the Markov model, where  $e$  is a random error pattern and  $\nu$  is a random near codeword, and the value  $t_{\nu} = 0$  in the initial vector.

The DFRs estimated by  $M0, M1$  can be combined to estimate the true DFR, where there are  $|\mathcal{N}| = 2r$  near codewords:

DFR Formula

**Linear:**  $DFR = DFR(M0) + |\mathcal{N}|(DFR(M1) - DFR(M0))$

# Markov Model DFR<sup>9</sup>



<sup>9</sup>*Error floor prediction with Markov Models*, A-, Jun Bo Lau, Ray Perlner, Angela Robinson, Jean-Pierre Tillich, and Valentin Vasseur. 2025\*.

## Markov Approach 2: Accounting for key shape

### Assumption

Let  $\nu$  denote the near codeword with the largest **initial** overlap with the error. Assume that  $\nu$  remains the only near codeword of concern during the decoding process.

$$\text{State} = (s, t, u, b)$$

$$u = \text{size of the overlap with } \nu$$

$$b \in \{0, 1\} : \nu = x^s h_b(x)$$

Model uses the structure of the parity check matrix to compute transition probabilities.

### Benefits:

- ▶ With this model, we no longer need to isolate and amplify the effects of a single near codeword.
- ▶ We take into account the **structure of the key** : **very accurate predictions**.

## How the key structure is taken into account

**Degree distribution** of the two subgraphs of the Tanner graph generated by the two near codewords  $\nu_0 = h_0(x)$  and  $\nu_1 = h_1(x)$ .

- ▶ **variable nodes of  $\mathcal{G}_b$ :**  $j \in \nu_b$
- ▶ **check nodes of  $\mathcal{G}_b$**   $i$  such that  $H_{ij} = 1$  for variable node  $j$
- ▶ edge between check node  $i$  and variable node  $j$  iff  $H_{ij} = 1$ .

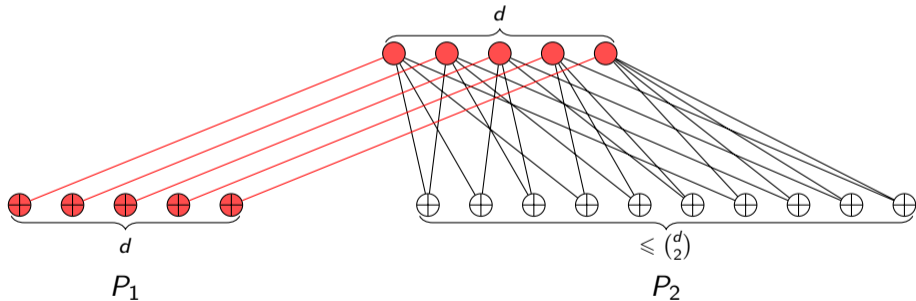
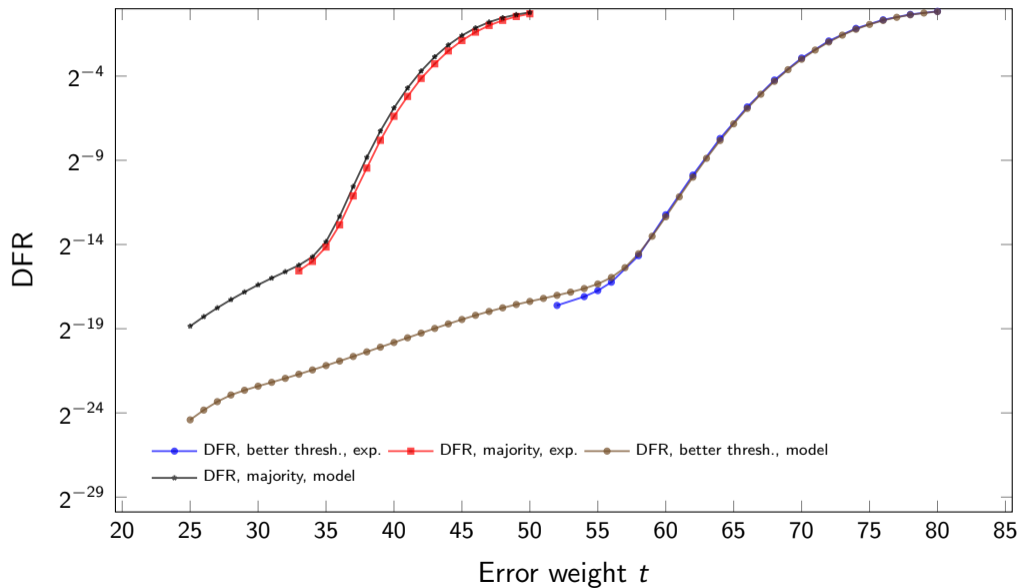
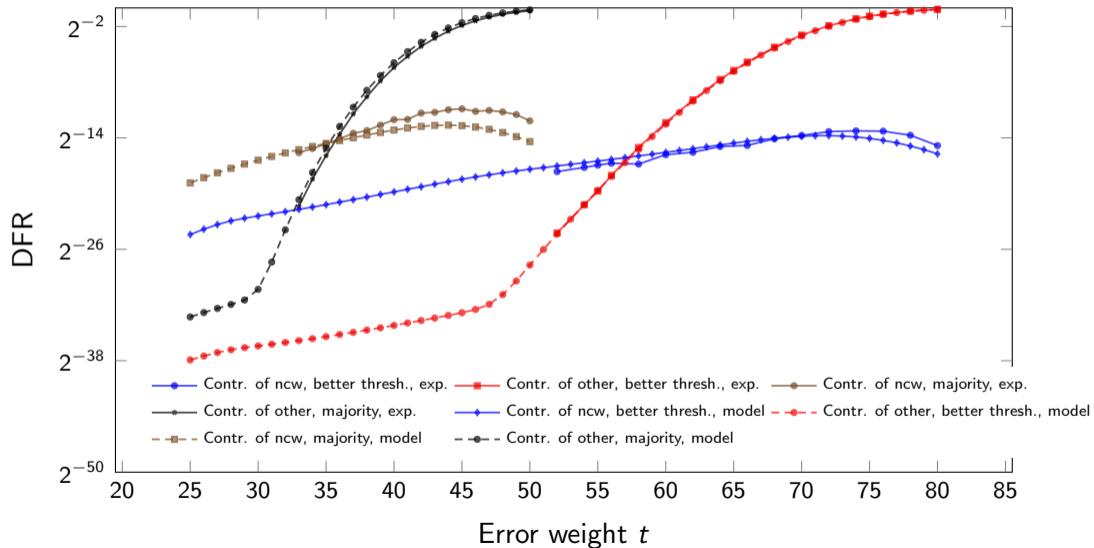


Figure: Illustration of the subgraph  $\mathcal{G}$  induced by the near codeword  $\nu$ .

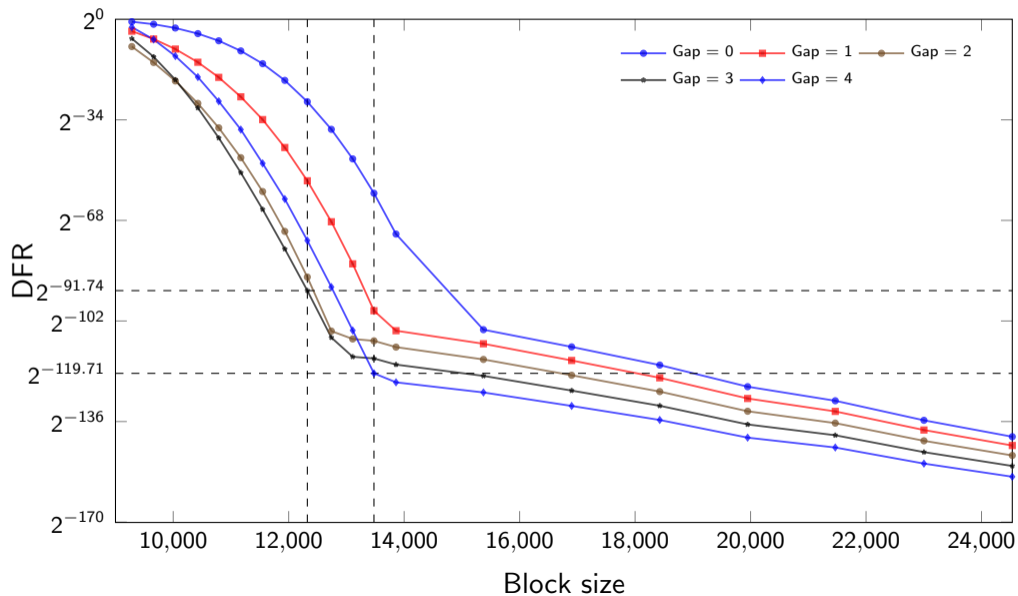
# Experiment vs. Model 2



# Experiment vs. Model 2 (II)



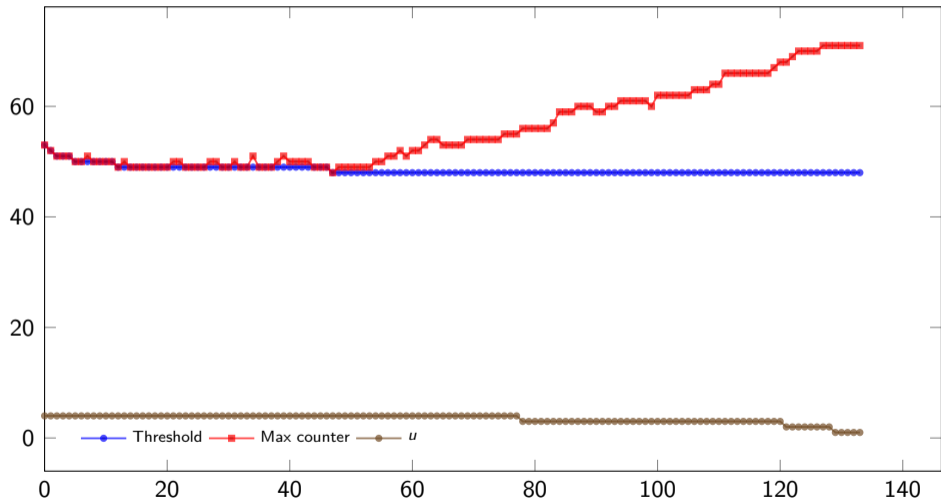
# BIKE parameter 1



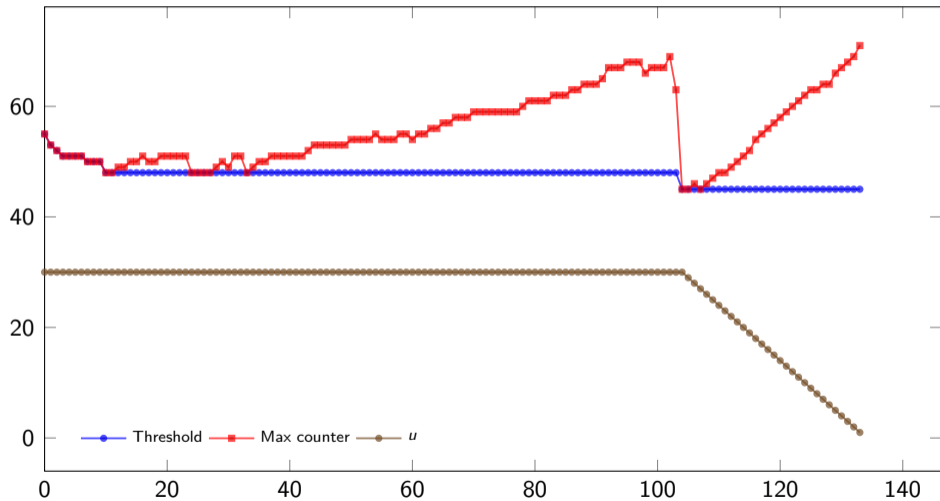
# Conclusion

- ▶ For a **single threshold accurate prediction** of the waterfall + error floor behavior by taking into account the key structure
- ▶ Typical key for BIKE 1 with a single conservative threshold,  $DFR \approx 2^{-90}$  and BIKE 1 + 10%  $DFR \approx 2^{-120}$
- ▶ Significant gain by choosing as in the latest BIKE decoder, thresholds that decrease during decoding.
- ▶ adapting Markov modeling to this case  $(s, t, u, b) \rightarrow (s, t, u, b, T)$  where  $T =$  max counter value.

# Maximum counter values $u = 4$



# Maximum counter values $u = 30$



# Different thresholds ( $r = 1723, d = 17$ )

