Decoding QC-MDPC

Valentin Vasseur

QC-MDPC scheme

$p, d, t \in \mathbb{N}$ parameters, p prime, d odd, $2d \sim t \sim \sqrt{2p}$
$\mathcal{R} = \mathbb{F}_2[X]/(X^p - 1)$

$h_0, h_1 \leftarrow \mathcal{R}$
$|h_0| = |h_1| = d$
h_0 invertible

$$\xrightarrow{h = h_1 h_0^{-1}}$$

$e_0, e_1 \in \mathcal{R}$
$|e_0| + |e_1| = t$

$$\xleftarrow{s = e_0 + e_1 h}$$

$sh_0 = e_0 h_0 + e_1 h_1$
$(e_0, e_1) \leftarrow \mathrm{Decode}(sh_0)$

Improving the decoding algorithm

Motivation :

- QC-MDPC are likely to be proposed for the NIST post-quantum cryptography project
- Current algorithm fails with a small probability
- An attacker could use the failures to get the secret key [GJS16]

Goals :

- Improve the algorithm
- Have a better understanding of the algorithm to estimate decoding failure probability

A decoding algorithm : Bit flip

procedure Bit-flipping(y, H)
  $\mathsf{y} \leftarrow y$
  while $H\mathsf{y}^\intercal \neq 0$ **do**
    $s \leftarrow H\mathsf{y}^\intercal$
    for $j = 1, \ldots, n$ **do**
      if $\sigma_j = \langle s, h_j \rangle \geq \text{threshold}$ **then**
        $\mathsf{y}_j \leftarrow 1 - \mathsf{y}_j$
  return y

$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathsf{y} - y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

A decoding algorithm : Bit flip

procedure Bit-flipping(y, H)
    $\mathsf{y} \leftarrow y$
    while $H\mathsf{y}^\intercal \neq 0$ **do**
        $s \leftarrow H\mathsf{y}^\intercal$
        for $j = 1, \ldots, n$ **do**
            if $\sigma_j = \langle s, h_j \rangle \geq \mathrm{threshold}$ **then**
                $\mathsf{y}_j \leftarrow 1 - \mathsf{y}_j$
    return y

$$\sigma = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 2 & 1 & 3 & 1 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \end{pmatrix}$$

$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathsf{y} - y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

A decoding algorithm : Bit flip

procedure Bit-flipping(y, H)
  $\mathbf{y} \leftarrow y$
  while $H\mathbf{y}^\intercal \neq 0$ **do**
    $s \leftarrow H\mathbf{y}^\intercal$
    for $j = 1, \dots, n$ **do**
      if $\sigma_j = \langle s, h_j \rangle \geq \text{threshold}$ **then**
        $\mathbf{y}_j \leftarrow 1 - \mathbf{y}_j$
  return $\mathbf{y}$

$$\sigma = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 2 & 1 & 3 & 1 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \end{pmatrix}$$

$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{y} - y = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

A decoding algorithm : Bit flip

procedure BIT-FLIPPING(y, H)
  $\mathbf{y} \leftarrow y$
  while $H\mathbf{y}^\intercal \neq 0$ **do**
    $s \leftarrow H\mathbf{y}^\intercal$
    for $j = 1, \ldots, n$ **do**
      if $\sigma_j = \langle s, h_j \rangle \geq \text{threshold}$ **then**
        $\mathbf{y}_j \leftarrow 1 - \mathbf{y}_j$
  return $\mathbf{y}$

$$\sigma = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 2 & 1 & 3 & 1 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \end{pmatrix}$$

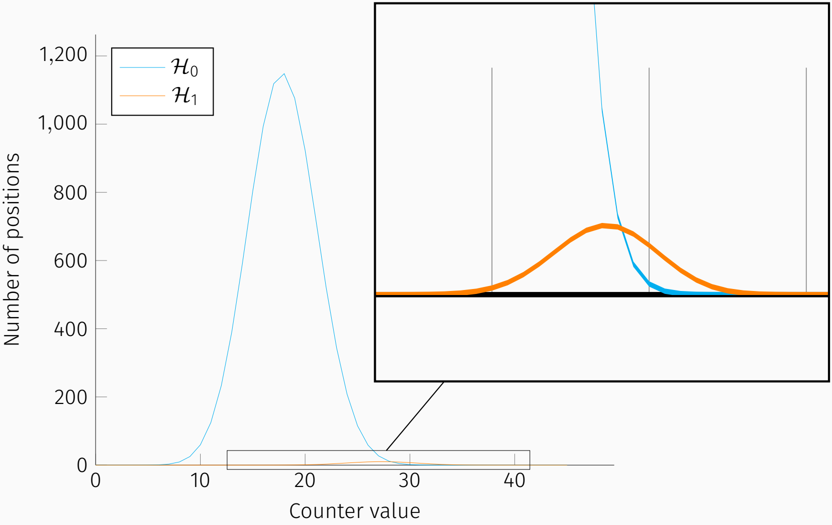
$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{y} - y = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

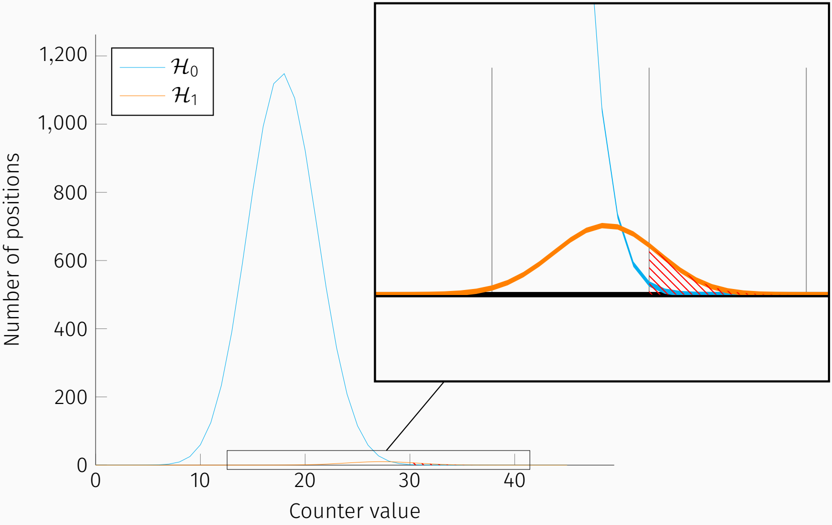
State of the art

- [MTSB13] : threshold $\max_i \sigma_i - \Delta$ ($\Delta \approx 5$ fixed) $\quad \mathrm{DFR} \approx 10^{-7}$

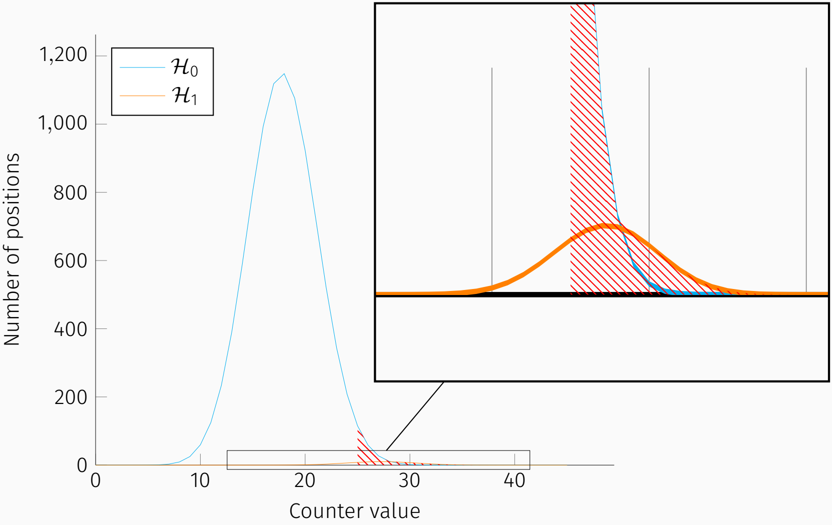
$\max_i \sigma_i - \Delta$



$\max_i \sigma_i - \Delta$



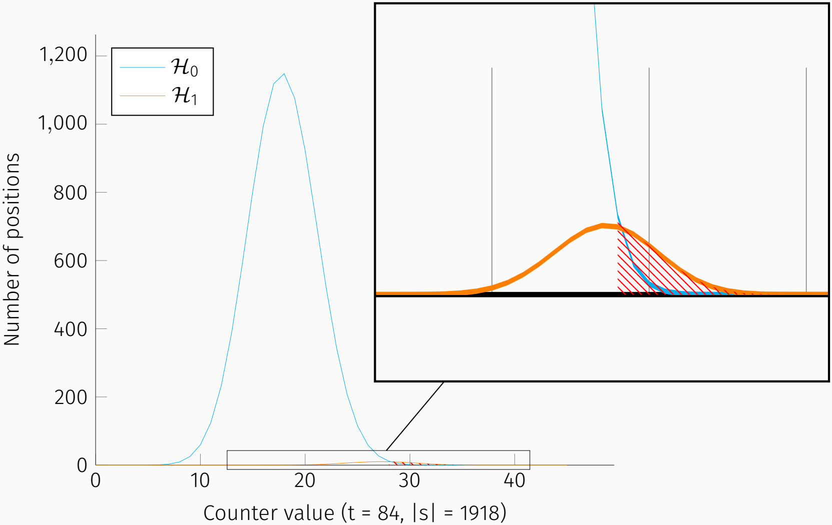
$\max_i \sigma_i - \Delta$



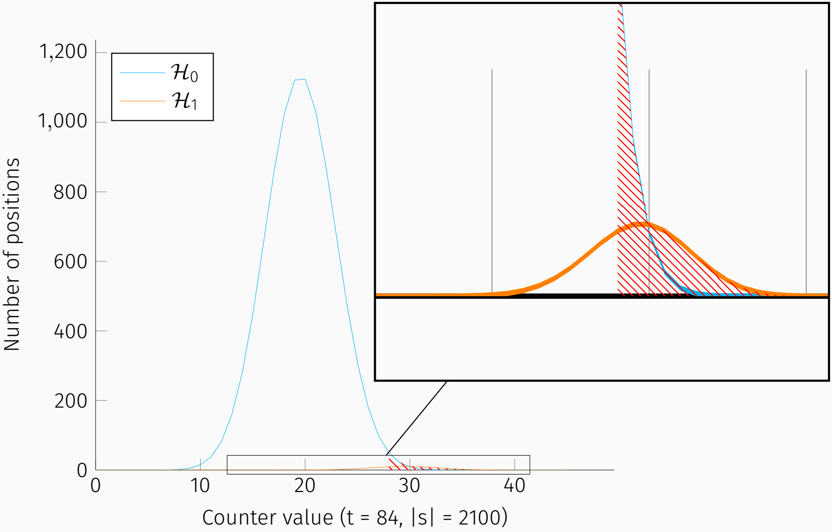
State of the art

- [MTSB13] : threshold $\max_i \sigma_i - \Delta$ ($\Delta \approx 5$ fixed) $\quad \mathrm{DFR} \approx 10^{-7}$
- [Cho16] : fixed precomputed thresholds for each iteration $\quad \mathrm{DFR} < 10^{-8}$

Fixed precomputed thresholds



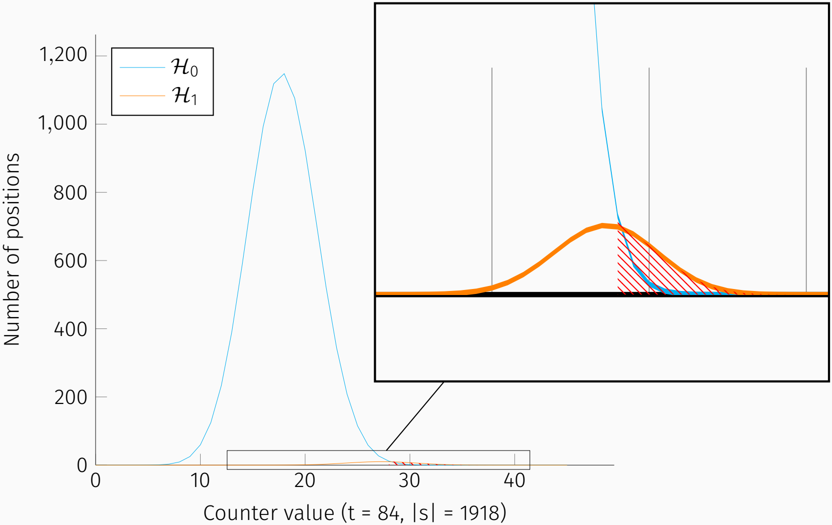
Fixed precomputed thresholds



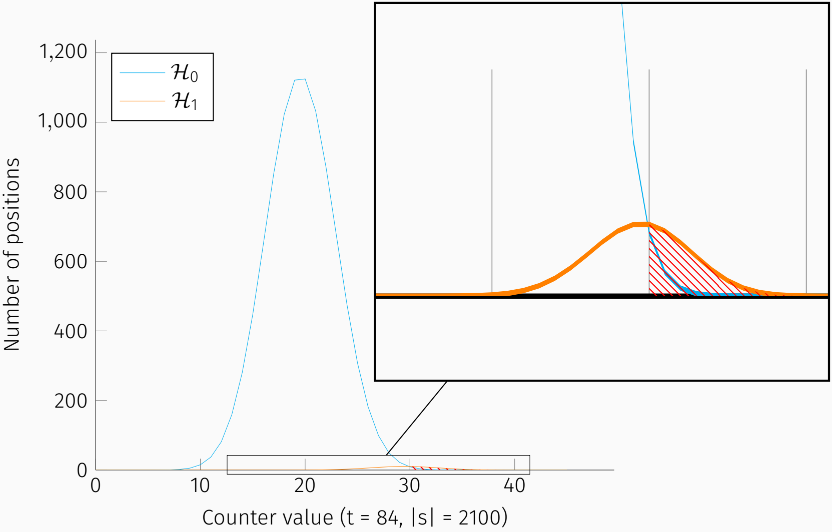
State of the art

- [MTSB13] : threshold $\max_i \sigma_i - \Delta$ ($\Delta \approx 5$ fixed) $\quad \mathrm{DFR} \approx 10^{-7}$
- [Cho16] : fixed precomputed thresholds for each iteration $\quad \mathrm{DFR} < 10^{-8}$
- [Cha17] : threshold dependent on the syndrome weight $\quad \mathrm{DFR} \approx 10^{-9}$

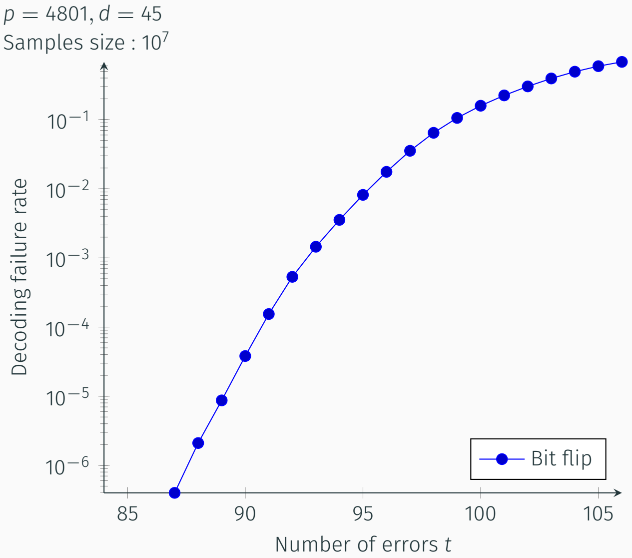
Threshold depending on the syndrome weight



Threshold depending on the syndrome weight



Decoding failure rate for QC-MDPC with oversized error weights



"Grey zones"

procedure GREY BIT-FLIPPING(y, H)
  $\mathbf{y} \leftarrow y$
  $G \leftarrow \{1, \ldots, n\}$
  while $H\mathbf{y}^\intercal \neq 0$ **do**
    $s \leftarrow H\mathbf{y}^\intercal$
    for $j = 1, \ldots, n$ **do**
      if $j \in G$ **and** $\sigma_j = \langle s, h_j \rangle \geq \mathrm{threshold}$ **then**
        $\mathbf{y}_j \leftarrow 1 - \mathbf{y}_j$
    $G \leftarrow \{j \mid \sigma_j \geq \mathrm{threshold}_G\}$
  return $\mathbf{y}$

$$\sigma = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 2 & 1 & 3 & 1 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \end{pmatrix}$$

$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{y} - y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

“Grey zones”

procedure Grey Bit-flipping(y, H)
    $\mathbf{y} \leftarrow y$
    $G \leftarrow \{1, \ldots, n\}$
    while $H\mathbf{y}^\intercal \neq 0$ do
        $s \leftarrow H\mathbf{y}^\intercal$
        for $j = 1, \ldots, n$ do
            if $j \in G$ and $\sigma_j = \langle s, h_j \rangle \geq \mathrm{threshold}$ then
                $\mathbf{y}_j \leftarrow 1 - \mathbf{y}_j$
        $G \leftarrow \{j \mid \sigma_j \geq \mathrm{threshold}_G\}$
    return $\mathbf{y}$

$$\sigma = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 2 & 1 & 3 & 1 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \end{pmatrix}$$

$$H = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{y} - y = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

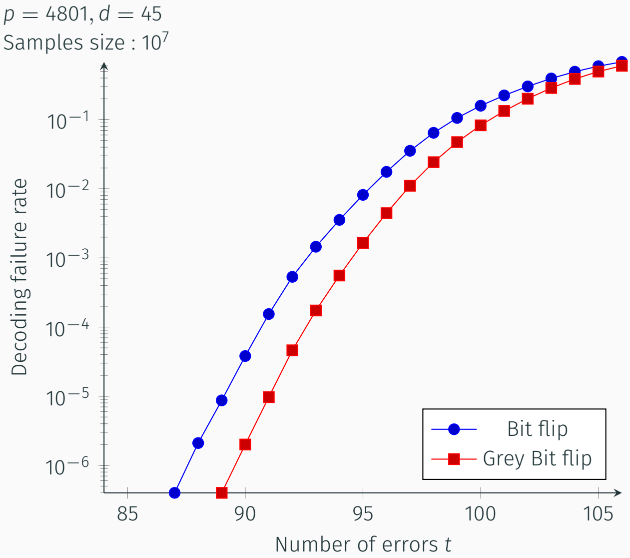
procedure GREY BIT-FLIPPING(y, H)
    $\mathbf{y} \leftarrow y$
    $G \leftarrow \{1, \dots, n\}$
    while $H\mathbf{y}^\intercal \neq 0$ **do**
        $s \leftarrow H\mathbf{y}^\intercal$
        for $j = 1, \dots, n$ **do**
            if $j \in G$ and $\sigma_j = \langle s, h_j \rangle \geq \mathrm{threshold}$ **then**
                $\mathbf{y}_j \leftarrow 1 - \mathbf{y}_j$
        $G \leftarrow \{j \mid \sigma_j \geq \mathrm{threshold}_G\}$
    return $\mathbf{y}$

$$\sigma = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 2 & 1 & 3 & 1 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \end{pmatrix}$$

$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{y} - y = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Decoding failure rate for QC-MDPC with oversized error weights



2 Bit flip

procedure 2 Bit-flipping(y, H)
  $\mathsf{y} \leftarrow y$
  while $H\mathsf{y}^\intercal \neq 0$ **do**
    $s \leftarrow H\mathsf{y}^\intercal$
    for $j = 1, \ldots, n$ **do**
      $\mathsf{y}_j \leftarrow F(\mathsf{y}_j, \sigma_j)$
  return y

$$\sigma = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 2 & 1 & 3 & 1 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \end{pmatrix}$$

$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathsf{y} - y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2 Bit flip

procedure 2 BIT-FLIPPING(y, H)
    $\mathbf{y} \leftarrow y$
    while $H\mathbf{y}^\intercal \neq 0$ **do**
        $s \leftarrow H\mathbf{y}^\intercal$
        for $j = 1, \ldots, n$ **do**
            $\mathbf{y}_j \leftarrow F(\mathbf{y}_j, \sigma_j)$
    return $\mathbf{y}$

$$\sigma = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 2 & 1 & 3 & 1 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \end{pmatrix}$$

$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{y} - y = \begin{pmatrix} \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} & 0 & \boxed{1} & 0 & 0 & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{0} & 0 & \boxed{0} & \boxed{0} \end{pmatrix}$$

procedure 2 BIT-FLIPPING(y, H)
  $\mathbf{y} \leftarrow y$
  while $H\mathbf{y}^\intercal \neq 0$ **do**
    $s \leftarrow H\mathbf{y}^\intercal$
    for $j = 1, \ldots, n$ **do**
      $\mathbf{y}_j \leftarrow F(\mathbf{y}_j, \sigma_j)$
  return $\mathbf{y}$

$$\sigma = \begin{pmatrix} 1 & 2 & 2 & 1 & 3 & 1 & 1 & 3 & 1 & 0 & 2 & 3 & 1 & 2 & 2 & 1 & 2 & 2 \end{pmatrix}$$

$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \qquad s = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{y} - y = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2 Bit flip

procedure 2 BIT-FLIPPING(y, H)

  $\mathbf{y} \leftarrow y$

  while $H\mathbf{y}^{\intercal} \neq 0$ **do**

    $s \leftarrow H\mathbf{y}^{\intercal}$

    for $j = 1, \ldots, n$ **do**

      $\mathbf{y}_j \leftarrow F(\mathbf{y}_j, \sigma_j)$

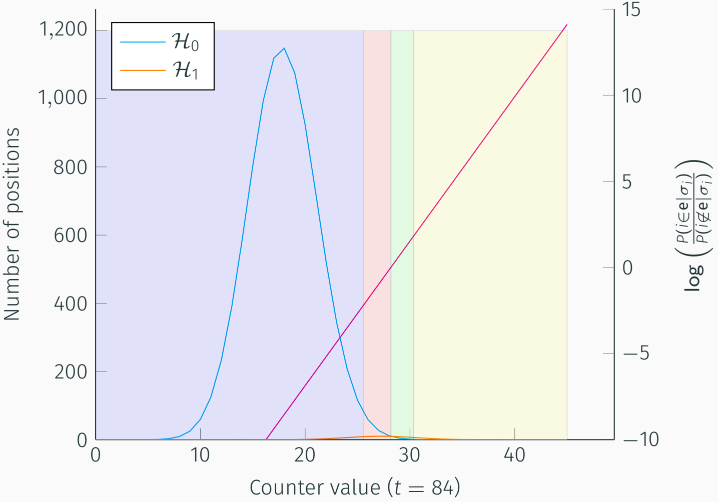
  return $\mathbf{y}$

$$\sigma = \begin{pmatrix} 1 & 2 & 2 & 1 & 3 & 1 & 1 & 3 & 1 & 0 & 2 & 3 & 1 & 2 & 2 & 1 & 2 & 2 \end{pmatrix}$$

$$H = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}
\qquad
s = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

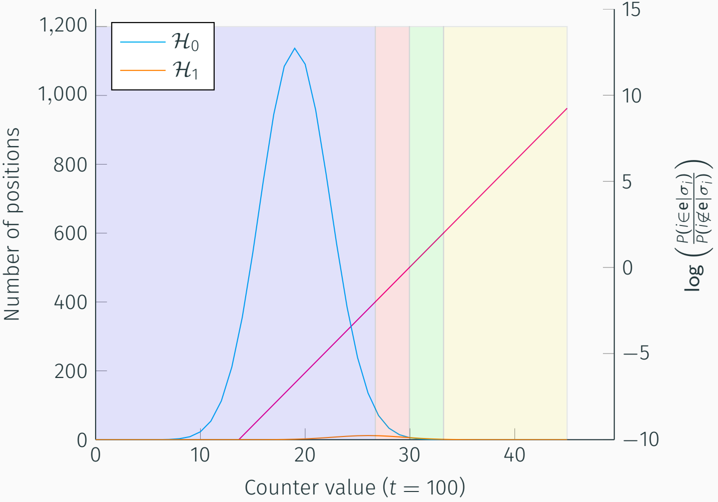
$$\mathbf{y} - y = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Defining F



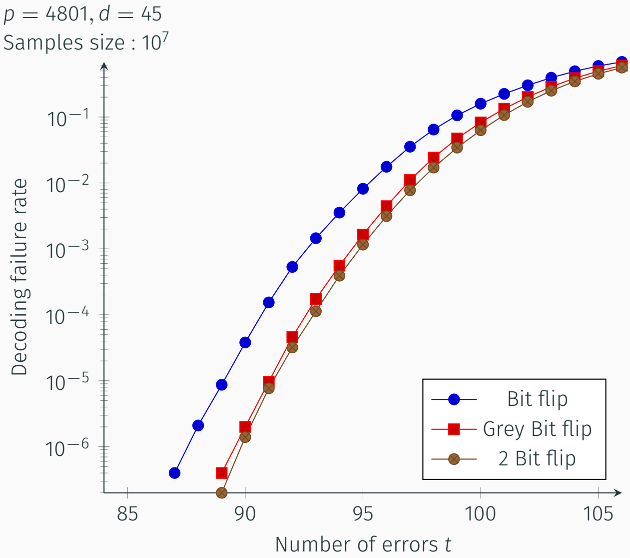
	Zone			
	0	1	2	3
0	0	0	0	1
1	1	1	1	0
0	0	0	1	1
1	1	1	0	0

Defining F

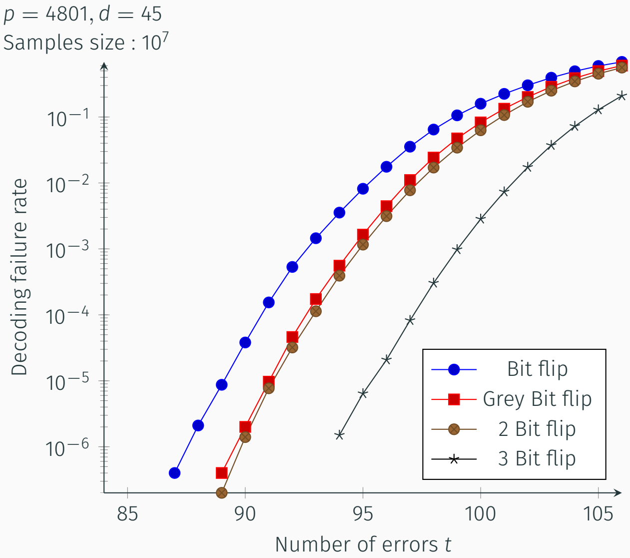


	Zone			
	0	1	2	3
0	0	0	0	1
1	1	1	1	0
0	0	0	1	1
1	1	1	0	0

Decoding failure rate for QC-MDPC with oversized error weights



Decoding failure rate for QC-MDPC with oversized error weights



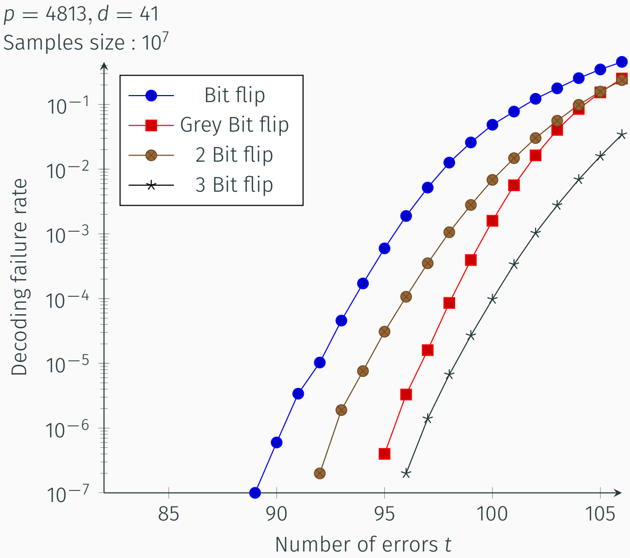
Ouroboros

$p, d \in \mathbb{N}$ parameters, p prime, d odd, $2d \sim \sqrt{2p}$
$\mathcal{R} = \mathbb{F}_2[X]/(X^p - 1)$

$$
\begin{array}{lcl}
h, x, y \leftarrow \mathcal{R} & \xrightarrow[]{h \quad s = x + yh} & \\
|x| = |y| = d & & \\
 & & e_0, e_1, e \in \mathcal{R} \\
 & & |e_0| = |e_1| = d,\ |e| = 3d \\
ys_0 + s_1 = ye_0 + xe_1 + e & \xleftarrow[]{s_0 = e_0 + e_1 h \quad s_1 = e + e_1 s} & \\
(e_0, e_1) \leftarrow \mathrm{Decode}(ys_0 + s_1) & & \\
e = s_1 - e_1 s & &
\end{array}
$$

Uses the same decoder as the MDPC with some minor adjustments

Decoding failure rate for Ouroboros with oversized error weights



Conclusion

In this presentation :

- Decreased DFR by using *soft* information

Further improvements :

- Find better thresholds for variants
- Find a theoretical bound on the decoding failure probability

Julia CHAULET. “Étude de cryptosystèmes à clé publique basés sur les codes MDPC quasi-cycliques”. Thèse de doct. University Pierre et Marie Curie, 2017.

Tung CHOU. “QcBits : Constant-Time Small-Key Code-Based Cryptography”. In : *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*. 2016, p. 280–300. DOI : 10.1007/978-3-662-53140-2_14. URL : https://doi.org/10.1007/978-3-662-53140-2_14.

Qian GUO, Thomas JOHANSSON et Paul STANKOVSKI. “A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors”. In : *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*. 2016, p. 789–815. DOI : 10.1007/978-3-662-53887-6_29. URL : http://dx.doi.org/10.1007/978-3-662-53887-6_29.

Rafael MISOCZKI et al. “MDPC-McEliece : New McEliece variants from Moderate Density Parity-Check codes”. In : *Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12, 2013*. 2013, p. 2069–2073. DOI : 10.1109/ISIT.2013.6620590. URL : http://dx.doi.org/10.1109/ISIT.2013.6620590.