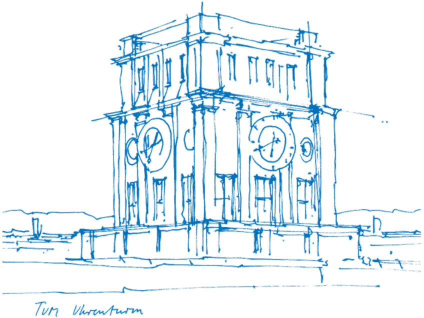
Tighter DFR Analysis & New Decoders for HQC

Marco Baldi, Sebastian Bitzer, Nicholas Lilla, Paolo Santini

Technical University of Munich
Università Politecnica delle Marche

CBC 2024

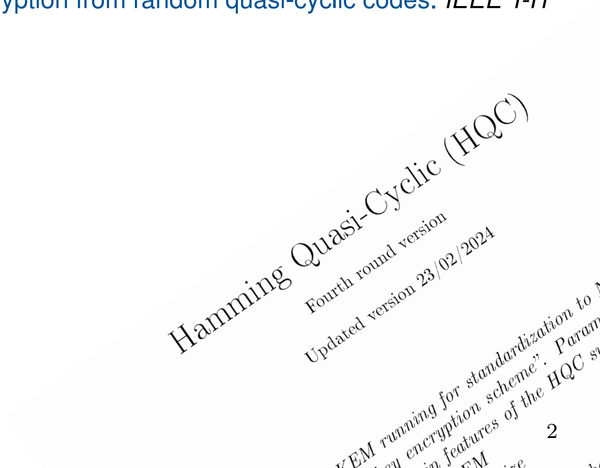


Hamming Quasi-Cyclic (HQC)

- Aguilar-Melchor, C., et al. (2017). Hamming quasi-cyclic (HQC). *NIST PQC*
- Aguilar-Melchor, C., et al. (2018). Efficient encryption from random quasi-cyclic codes. *IEEE T-IT*

Hamming Quasi-Cyclic (HQC)

- Aguilar-Melchor, C., et al. (2017). Hamming quasi-cyclic (HQC). *NIST PQC*
- Aguilar-Melchor, C., et al. (2018). Efficient encryption from random quasi-cyclic codes. *IEEE T-IT*


Hamming Quasi-Cyclic (HQC)

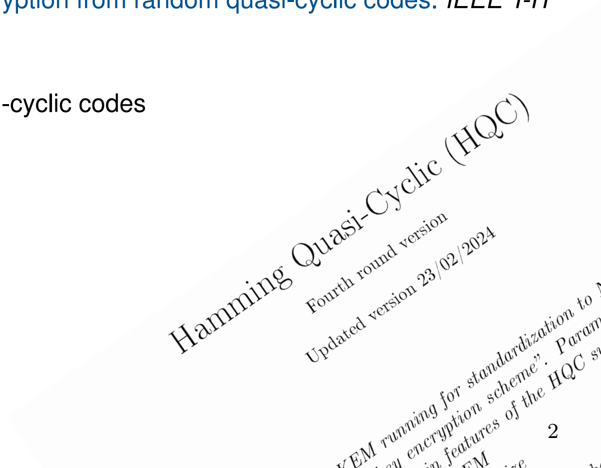
Fourth round version

Updated version 23/02/2024

Hamming Quasi-Cyclic (HQC)

- Aguilar-Melchor, C., et al. (2017). Hamming quasi-cyclic (HQC). *NIST PQC*
- Aguilar-Melchor, C., et al. (2018). Efficient encryption from random quasi-cyclic codes. *IEEE T-IT*

- Based on hardness of decoding random quasi-cyclic codes
- No hidden code structure
- Precise DFR analysis

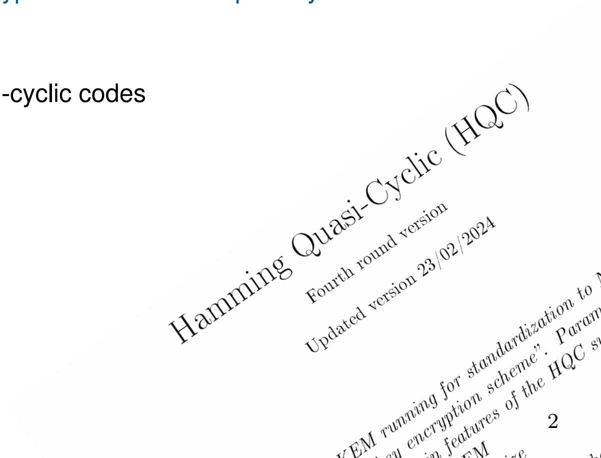

Hamming Quasi-Cyclic (HQC)
Fourth round version
Updated version 23/02/2024

Hamming Quasi-Cyclic (HQC)

Aguilar-Melchor, C., et al. (2017). Hamming quasi-cyclic (HQC). *NIST PQC*

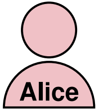
Aguilar-Melchor, C., et al. (2018). Efficient encryption from random quasi-cyclic codes. *IEEE T-IT*

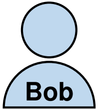
- Based on hardness of decoding random quasi-cyclic codes
- No hidden code structure
- Precise DFR analysis



HQC in a Nutshell

HQC in a Nutshell





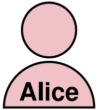
$\boldsymbol{h} \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$

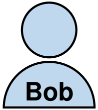
$\boldsymbol{u}_1, \boldsymbol{u}_2 \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$ of wt w_u

$\boldsymbol{s} \leftarrow \boldsymbol{u}_1 + \boldsymbol{h}\boldsymbol{u}_2$

$$\xrightarrow{(\boldsymbol{h}, \boldsymbol{s})}$$

HQC in a Nutshell





Alice:

$\boldsymbol{h} \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$

$\boldsymbol{u}_1, \boldsymbol{u}_2 \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$ of wt w_u

$\boldsymbol{s} \leftarrow \boldsymbol{u}_1 + \boldsymbol{h}\boldsymbol{u}_2$

$\xrightarrow{(\boldsymbol{h}, \boldsymbol{s})}$

Bob:

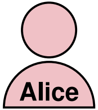
$\boldsymbol{c} \leftarrow \mathcal{C}.\textsc{Enc}(\boldsymbol{m})$

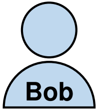
$\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3 \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$ of wt w_r

$(\boldsymbol{t}_1, \boldsymbol{t}_2) \leftarrow (\boldsymbol{c} + \boldsymbol{s}\boldsymbol{r}_2 + \boldsymbol{r}_3, \boldsymbol{r}_1 + \boldsymbol{h}\boldsymbol{r}_2)$

$\xleftarrow{(\boldsymbol{t}_1, \boldsymbol{t}_2)}$

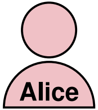
HQC in a Nutshell

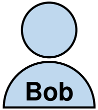




Alice		Bob
$\boldsymbol{h} \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$		
$\boldsymbol{u}_1, \boldsymbol{u}_2 \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$ of wt w_u		
$\boldsymbol{s} \leftarrow \boldsymbol{u}_1 + \boldsymbol{h}\boldsymbol{u}_2$	$\xrightarrow{(\boldsymbol{h}, \boldsymbol{s})}$	$\boldsymbol{c} \leftarrow \mathcal{C}.\textsc{Enc}(\boldsymbol{m})$
		$\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3 \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$ of wt w_r
$\hat{\boldsymbol{m}} \leftarrow \mathcal{C}.\textsc{Dec}(\boldsymbol{t}_1 - \boldsymbol{t}_2\boldsymbol{u}_2)$	$\xleftarrow{(\boldsymbol{t}_1, \boldsymbol{t}_2)}$	$(\boldsymbol{t}_1, \boldsymbol{t}_2) \leftarrow (\boldsymbol{c} + \boldsymbol{s}\boldsymbol{r}_2 + \boldsymbol{r}_3, \boldsymbol{r}_1 + \boldsymbol{h}\boldsymbol{r}_2)$

HQC in a Nutshell





Alice:

$\boldsymbol{h} \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$

$\boldsymbol{u}_1, \boldsymbol{u}_2 \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$ of wt w_u

$\boldsymbol{s} \leftarrow \boldsymbol{u}_1 + \boldsymbol{h}\boldsymbol{u}_2$

$\xrightarrow{(\boldsymbol{h}, \boldsymbol{s})}$

Bob:

$\boldsymbol{c} \leftarrow \mathcal{C}.\textsf{Enc}(\boldsymbol{m})$

$\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3 \xleftarrow{\$} \mathbb{F}_2[x]/(x^n - 1)$ of wt w_r

$(\boldsymbol{t}_1, \boldsymbol{t}_2) \leftarrow (\boldsymbol{c} + \boldsymbol{s}\boldsymbol{r}_2 + \boldsymbol{r}_3, \boldsymbol{r}_1 + \boldsymbol{h}\boldsymbol{r}_2)$

$\xleftarrow{(\boldsymbol{t}_1, \boldsymbol{t}_2)}$

Alice:

$\hat{\boldsymbol{m}} \leftarrow \mathcal{C}.\textsf{Dec}(\boldsymbol{t}_1 - \boldsymbol{t}_2\boldsymbol{u}_2)$

$\mathcal{C}$ needs to decode $\boldsymbol{t}_1 - \boldsymbol{t}_2\boldsymbol{u}_2 = \boldsymbol{c} + \underbrace{\boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3}_{\text{error } \boldsymbol{e}}$

A First Look at the Error

- $P(|\boldsymbol{e}| = w)$ difficult for $\boldsymbol{e} = \boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3$
- $\rho = P(e_i = 1)$ simple

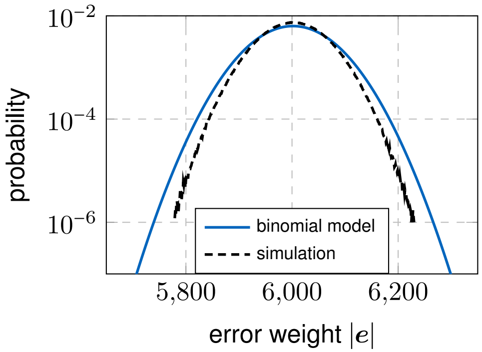
A First Look at the Error

- $P(|\boldsymbol{e}| = w)$ difficult for $\boldsymbol{e} = \boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3$
- $\rho = P(e_i = 1)$ simple

Binomial Approximation

Under the independence assumption,

$$P(|\boldsymbol{e}| = w) \approx \binom{n}{w}\rho^t(1-\rho)^{n-w}.$$



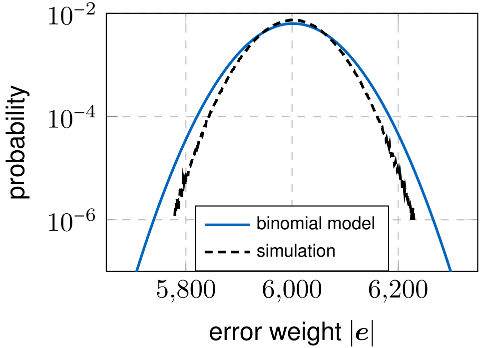
A First Look at the Error

- $P(|\boldsymbol{e}| = w)$ difficult for $\boldsymbol{e} = \boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3$
- $\rho = P(e_i = 1)$ simple

Binomial Approximation

Under the independence assumption,

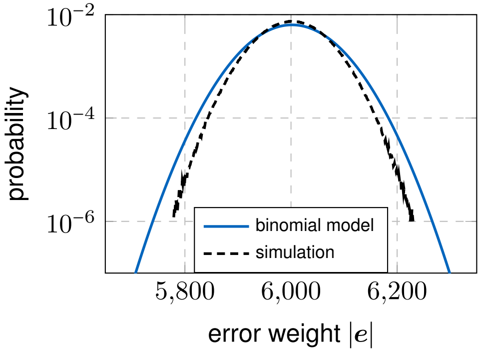
$$P(|\boldsymbol{e}| = w) \approx \binom{n}{w}\rho^t(1-\rho)^{n-w}.$$



Seems conservative but not precise!

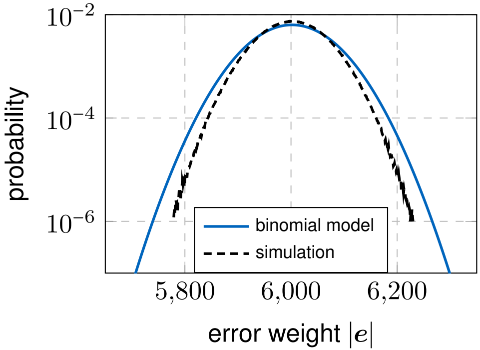
A Second Look at the Error

- Consider $\boldsymbol{a} = \boldsymbol{u} \cdot \boldsymbol{r} = \sum_{\ell \in \mathrm{supp}(\boldsymbol{u})} x^{\ell} \cdot \boldsymbol{r}(x)$



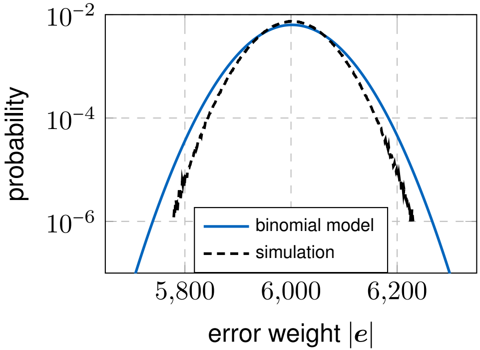
A Second Look at the Error

- Consider $\boldsymbol{a} = \boldsymbol{u} \cdot \boldsymbol{r} = \sum_{\ell \in \mathrm{supp}(\boldsymbol{u})} x^{\ell} \cdot \boldsymbol{r}(x)$
- b_i = # ones added in i-th position



A Second Look at the Error

- Consider $\boldsymbol{a} = \boldsymbol{u} \cdot \boldsymbol{r} = \sum_{\ell \in \mathrm{supp}(\boldsymbol{u})} x^{\ell} \cdot \boldsymbol{r}(x)$
- b_i = # ones added in i-th position
- $a_i = b_i \mod 2$
- $\sum_i b_i = |\boldsymbol{u}| \cdot |\boldsymbol{r}|$



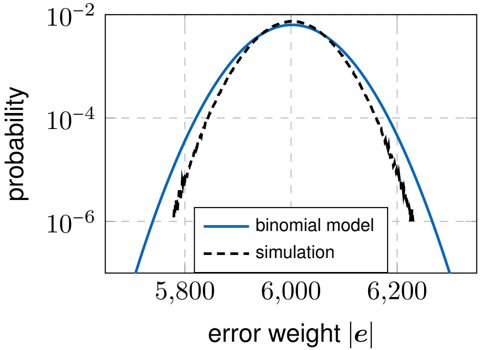
A Second Look at the Error

- Consider $\boldsymbol{a} = \boldsymbol{u} \cdot \boldsymbol{r} = \sum_{\ell \in \mathrm{supp}(\boldsymbol{u})} x^{\ell} \cdot \boldsymbol{r}(x)$
- b_i = # ones added in i-th position
- $a_i = b_i \mod 2$
- $\sum_i b_i = |\boldsymbol{u}| \cdot |\boldsymbol{r}|$

Proposed Approximation

Assume $b_0, \ldots, b_{n-1}$ indep. hypergeometric, let $a_i = b_i \mod 2$:

$$P(|\boldsymbol{u} \cdot \boldsymbol{r}| = w) \approx P\left(\sum_i a_i \mid \sum_i b_i = |\boldsymbol{u}| \cdot |\boldsymbol{r}|\right).$$



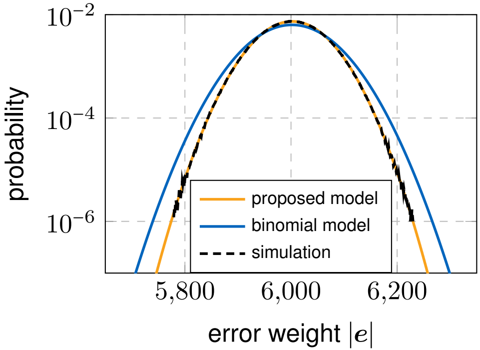
A Second Look at the Error

- Consider $\boldsymbol{a} = \boldsymbol{u} \cdot \boldsymbol{r} = \sum_{\ell \in \mathrm{supp}(\boldsymbol{u})} x^\ell \cdot \boldsymbol{r}(x)$
- b_i = # ones added in i-th position
- $a_i = b_i \mod 2$
- $\sum_i b_i = |\boldsymbol{u}| \cdot |\boldsymbol{r}|$

Proposed Approximation

Assume $b_0, \ldots, b_{n-1}$ indep. hypergeometric, let $a_i = b_i \mod 2$:

$$P(|\boldsymbol{u} \cdot \boldsymbol{r}| = w) \approx P\left(\sum_i a_i \mid \sum_i b_i = |\boldsymbol{u}| \cdot |\boldsymbol{r}|\right).$$

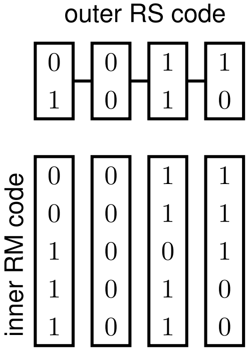


Impact on DFR estimation?

Tensor Product Code in HQC

Encoder

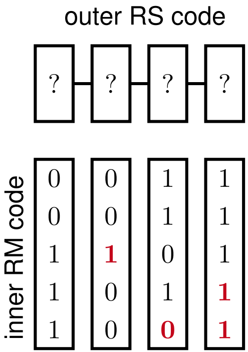
1. Encode outer RS code
2. Encode inner RM code



Decoder

1. Decode inner RM code
2. Decode outer RS code

Tensor Product Code in HQC



Encoder

1. Encode outer RS code
2. Encode inner RM code

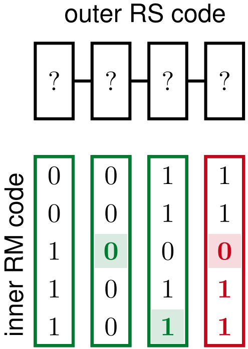
Decoder

1. Decode inner RM code
2. Decode outer RS code

Tensor Product Code in HQC

Encoder

1. Encode outer RS code
2. Encode inner RM code



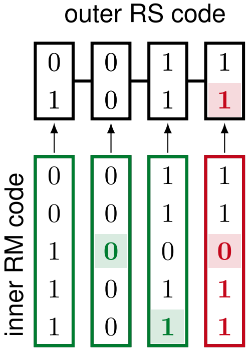
Decoder

1. Decode inner RM code
2. Decode outer RS code

Tensor Product Code in HQC

Encoder

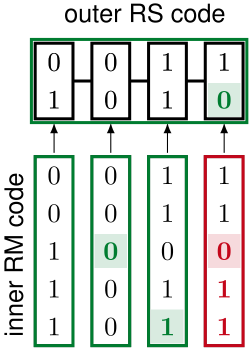
1. Encode outer RS code
2. Encode inner RM code



Decoder

1. Decode inner RM code
2. Decode outer RS code

Tensor Product Code in HQC



Encoder

1. Encode outer RS code
2. Encode inner RM code

Decoder

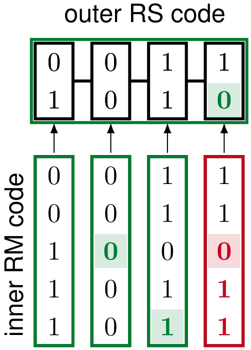
1. Decode inner RM code
2. Decode outer RS code

Simple DFR analysis under independence assumption ✓

Tensor Product Code in HQC

Encoder

1. Encode outer RS code
2. Encode inner RM code



Decoder

1. Decode inner RM code
2. Decode outer RS code

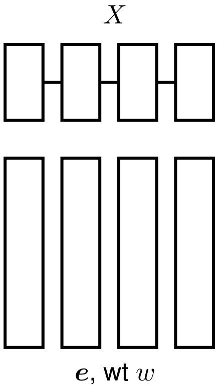
Simple DFR analysis under independence assumption ✓

But what about the proposed model?

DFR Analysis

Notation:

- X = # of erroneous outer symbols
- τ = correction capability outer code

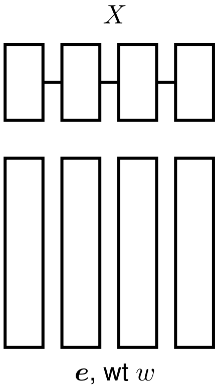


DFR Analysis

Notation:

- X = # of erroneous outer symbols
- τ = correction capability outer code

$\Rightarrow \mathrm{DFR} = \sum_w P(X > \tau \mid w) P(|\boldsymbol{e}| = w)$



DFR Analysis

Notation:

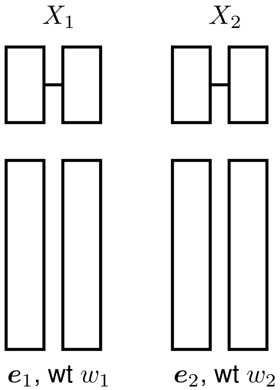
- X = # of erroneous outer symbols
- τ = correction capability outer code

$\Rightarrow \mathrm{DFR} = \sum_w P(X > \tau \mid w) P(|\boldsymbol{e}| = w)$

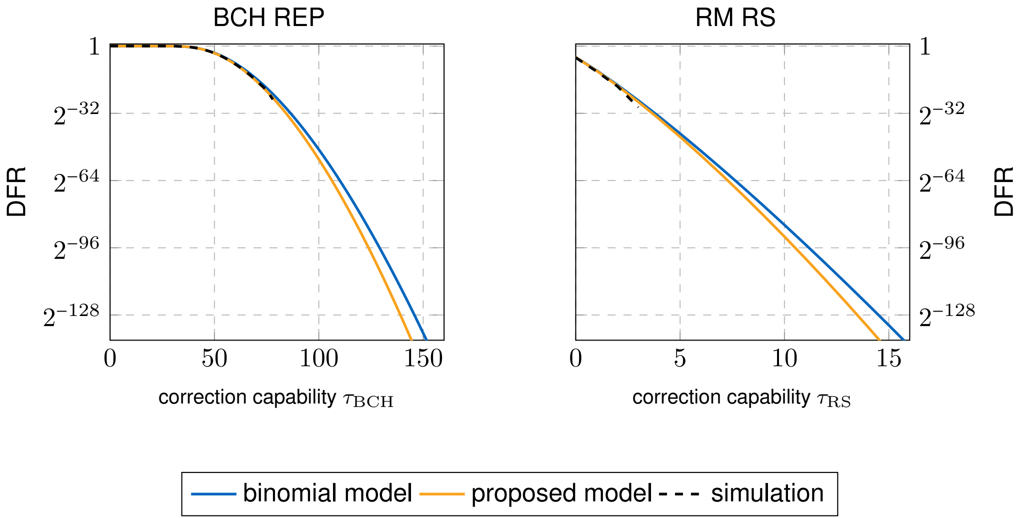
Divide and Analyze

Split $X = X_1 + X_2$ and $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$
with $|\boldsymbol{e}_1| = w_1, |\boldsymbol{e}_2| = w - w_1$:

$$P(X \mid w) = \sum_{w_1} P(X_1 \mid w_1) * P(X_2 \mid w - w_1) \cdot P(w_1 \mid w)$$



DFR Comparison



Using the Error Structure for Decoding

Remember: $\boldsymbol{e} = \boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3$

$$\boxed{\begin{matrix}0\\0\\1\\1\\1\end{matrix}} \quad \boxed{\begin{matrix}0\\0\\\mathbf{1}\\0\\0\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\0\\1\\\mathbf{0}\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\1\\\mathbf{1}\\\mathbf{1}\end{matrix}}$$

Using the Error Structure for Decoding

Remember: $\boldsymbol{e} = \boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3$

$$\boxed{\begin{matrix}0\\0\\1\\1\\1\end{matrix}} \quad \boxed{\begin{matrix}0\\0\\\mathbf{0}\\0\\0\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\0\\1\\\mathbf{1}\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\\mathbf{0}\\\mathbf{1}\\\mathbf{1}\end{matrix}}$$

Proposed Decoder

1. Decode inner codewords

Using the Error Structure for Decoding

Remember: $\boldsymbol{e} = \boldsymbol{u}_1 \boldsymbol{r}_2 + \boldsymbol{u}_2 \boldsymbol{r}_1 + \boldsymbol{r}_3$

Proposed Decoder

1. Decode inner codewords, get $\hat{e}$.

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ \mathbf{0} \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ \mathbf{1} \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}$$

$$\hat{\boldsymbol{e}} = \boxed{0 \ \cdots \ 0 \ \ \mathbf{1} \ \ 0 \ \ \mathbf{1} \ \ 0 \ \ \mathbf{1} \ \ \mathbf{0} \ \ \mathbf{0}}$$

Using the Error Structure for Decoding

Remember: $\boldsymbol{e} = \boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3$

Proposed Decoder

1. Decode inner codewords, get $\hat{e}$.
2. Estimate $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2$ using $\hat{e}, \boldsymbol{u}_1, \boldsymbol{u}_2$.

$$\boxed{\begin{matrix}0\\0\\1\\1\\1\end{matrix}} \quad \boxed{\begin{matrix}0\\0\\1\\0\\0\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\0\\1\\0\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\1\\1\\1\end{matrix}}$$

$$\hat{\boldsymbol{e}} = \boxed{0 \ \cdots \ 0 \ \ 1 \ \ 0 \ \ 1 \ \ 0 \ \ 1 \ \ 0 \ \ 0}$$

$$\hat{\boldsymbol{r}}_1 = \boxed{0 \ \cdots \ 0 \ \ 0 \ \ 1 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0}$$

$$\hat{\boldsymbol{r}}_2 = \boxed{0 \ \cdots \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 1 \ \ 0 \ \ 0 \ \ 0}$$

Using the Error Structure for Decoding

Remember: $\boldsymbol{e} = \boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3$

Proposed Decoder

1. Decode inner codewords, get $\hat{\boldsymbol{e}}$.
2. Estimate $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2$ using $\hat{\boldsymbol{e}}, \boldsymbol{u}_1, \boldsymbol{u}_2$.
3. Estimate error $\boldsymbol{e}^* = \boldsymbol{u}_1 \cdot \hat{\boldsymbol{r}}_2 + \boldsymbol{u}_2 \cdot \hat{\boldsymbol{r}}_1$.

$$\boxed{\begin{matrix}0\\0\\1\\1\\1\end{matrix}} \quad \boxed{\begin{matrix}0\\0\\1\\0\\0\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\0\\1\\0\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\1\\1\\1\end{matrix}}$$

$$\begin{aligned}
\hat{\boldsymbol{e}} &= \boxed{0 \ \cdots \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0} \\
\hat{\boldsymbol{r}}_1 &= \boxed{0 \ \cdots \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0} \\
\hat{\boldsymbol{r}}_2 &= \boxed{0 \ \cdots \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0} \\
\boldsymbol{e}^* &= \boxed{0 \ \cdots \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0}
\end{aligned}$$

Using the Error Structure for Decoding

Remember: $\boldsymbol{e} = \boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3$

Proposed Decoder

1. Decode inner codewords, get $\hat{e}$.
2. Estimate $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2$ using $\hat{e}, \boldsymbol{u}_1, \boldsymbol{u}_2$.
3. Estimate error $\boldsymbol{e}^* = \boldsymbol{u}_1 \cdot \hat{\boldsymbol{r}}_2 + \boldsymbol{u}_2 \cdot \hat{\boldsymbol{r}}_1$.
4. Decode $\boldsymbol{t}_1 + \boldsymbol{t}_2\boldsymbol{u}_2 - \boldsymbol{e}^* = \boldsymbol{c} + \boldsymbol{e} - \boldsymbol{e}^*$.

$$\boxed{\begin{array}{c}0\\1\\1\\1\\1\end{array}} \quad \boxed{\begin{array}{c}0\\0\\0\\0\\0\end{array}} \quad \boxed{\begin{array}{c}1\\1\\0\\1\\0\end{array}} \quad \boxed{\begin{array}{c}1\\1\\1\\0\\1\end{array}}$$

$$\hat{\boldsymbol{e}} = \boxed{0 \ \cdots \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0}$$

$$\hat{\boldsymbol{r}}_1 = \boxed{0 \ \cdots \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0}$$

$$\hat{\boldsymbol{r}}_2 = \boxed{0 \ \cdots \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0}$$

$$\boldsymbol{e}^* = \boxed{0 \ \cdots \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0}$$

Using the Error Structure for Decoding

Remember: $\boldsymbol{e} = \boldsymbol{u}_1\boldsymbol{r}_2 + \boldsymbol{u}_2\boldsymbol{r}_1 + \boldsymbol{r}_3$

Proposed Decoder

1. Decode inner codewords, get $\hat{e}$.
2. Estimate $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2$ using $\hat{e}, \boldsymbol{u}_1, \boldsymbol{u}_2$.
3. Estimate error $\boldsymbol{e}^* = \boldsymbol{u}_1 \cdot \hat{\boldsymbol{r}}_2 + \boldsymbol{u}_2 \cdot \hat{\boldsymbol{r}}_1$.
4. Decode $\boldsymbol{t}_1 + \boldsymbol{t}_2\boldsymbol{u}_2 - \boldsymbol{e}^* = \boldsymbol{c} + \boldsymbol{e} - \boldsymbol{e}^*$.

$$\boldsymbol{e} - \boldsymbol{e}^* = (\boldsymbol{r}_1 - \hat{\boldsymbol{r}}_1)\boldsymbol{u}_2 + (\boldsymbol{r}_2 - \hat{\boldsymbol{r}}_2)\boldsymbol{u}_1 + \boldsymbol{r}_3$$

$\Rightarrow$ error weight reduced if $\hat{\boldsymbol{r}}_1 \approx \boldsymbol{r}_1$ and $\hat{\boldsymbol{r}}_2 \approx \boldsymbol{r}_2$

$$\boxed{\begin{matrix}0\\1\\1\\1\\1\end{matrix}} \quad \boxed{\begin{matrix}0\\0\\0\\0\\0\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\0\\1\\0\end{matrix}} \quad \boxed{\begin{matrix}1\\1\\1\\0\\1\end{matrix}}$$

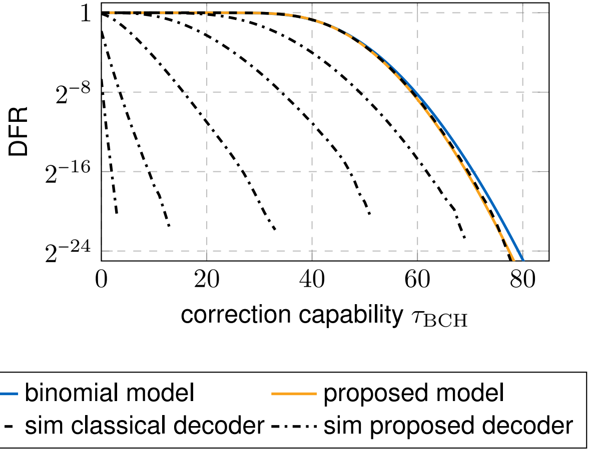
$$\hat{\boldsymbol{e}} = \boxed{0 \;\cdots\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 0}$$

$$\hat{\boldsymbol{r}}_1 = \boxed{0 \;\cdots\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0}$$

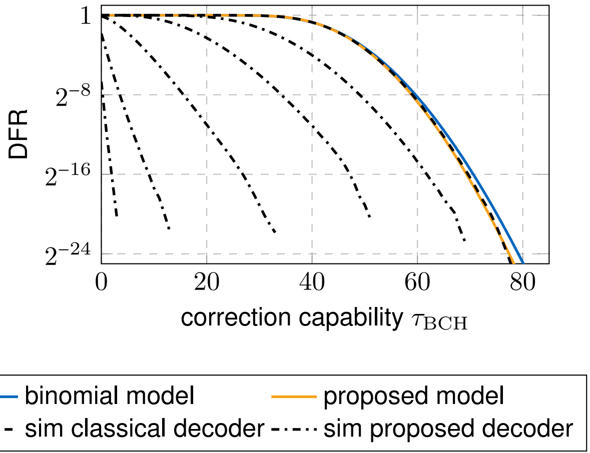
$$\hat{\boldsymbol{r}}_2 = \boxed{0 \;\cdots\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0}$$

$$\boldsymbol{e}^* = \boxed{0 \;\cdots\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0}$$

Decoding Performance Results



Decoding Performance Results



Considerable improvements conceivable ✓

Conclusion

The structure of the HQC error enables

- ☺ tighter DFR estimates
- ☺ improved decoding performance

Conclusion

The structure of the HQC error enables

- ☺ tighter DFR estimates
- ☺ improved decoding performance

Can one

- ? refine the ML bound for the RM code?
- ? provide DFR analysis for the Correlation Decoder?

Conclusion

The structure of the HQC error enables

- ☺ tighter DFR estimates
- ☺ improved decoding performance

Can one

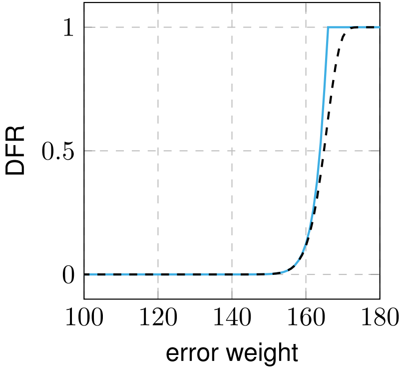
- ? refine the ML bound for the RM code?
- ? provide DFR analysis for the Correlation Decoder?

Thank you!

Questions?

A Word about ML Decoding of First-Order RM Codes

- efficient ML decoding:
  Fast Walsh-Hadamard transform
- bound on ML performance



How do we get $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2$?

Approximate MDPC-like equation: $\hat{e} \approx \boldsymbol{u}_1 \boldsymbol{r}_2 + \boldsymbol{u}_2 \boldsymbol{r}_1$

Correlation Estimate

WLOG, consider $\boldsymbol{r}_1$. Score computation:

$$\sigma_i = \sum_{j \in \mathrm{supp}(\boldsymbol{u}_2)} \mathbb{Z}(\hat{e}_{j+i \mod n}).$$

Threshold decision:

$$\hat{r}_{1,i} = \begin{cases} 1 & \text{if } \sigma_i \geq T, \\ 0 & \text{otherwise,} \end{cases}$$