A new Key Recovery Side-Channel Attack on HQC with Chosen Ciphertext

Guillaume Goy & Antoine Loiseau & Phillipe Gaborit

CEA Grenoble & University of Limoges

PQcrypto 2022

Introduction

- HQC is a code-based candidate (NIST PQC 4th round)
- SCA protection could be a criteria for standardization
- Decoders are knonwn to be vulnerable against SCA

State of the Art, SCA against HQC

	Ref	Year	Type	Nb. of Traces	Target
BCH	[PT19]	2019	TA	400.000.000	sk
	[WTBB+20]	2020	TA	6.000	sk
	[SRSWZ20]	2020	CCA	20.000	sk
RMRS	[Sch22, GHJ+21]	2021	TA	$\approx$ 850.000	sk
	[GLG22]	2022	Horizontal	1	message
	[SHR+22]	2022	CCA	50.000	sk
	Our attack	2022	CCA	20.000	sk

- Requierement: Static Secret Key Use

Notations I

Support of a vector

The support $\mathrm{Supp}(\mathbf{x})$ of a vector $\mathbf{x} = (x_0, \cdots, x_{n-1})$ is the location of its non-zero coordinates.

$$\mathrm{Supp}(\mathbf{x}) = \{i \in \mathbb{Z} \mid x_i \neq 0\}$$

Hamming weight

The hamming weight $\mathrm{wt}(\mathbf{x})$ of a vector $\mathbf{x} = (x_0, \cdots, x_{n-1})$ is the number of non-zero coordinates in $\mathbf{x}$.

$$\mathrm{wt}(\mathbf{x}) = \#\{i \in \mathbb{Z} \mid x_i \neq 0\} = \#\,\mathrm{Supp}(\mathbf{x})$$

Hamming Quasi-Cyclic (HQC) Framework

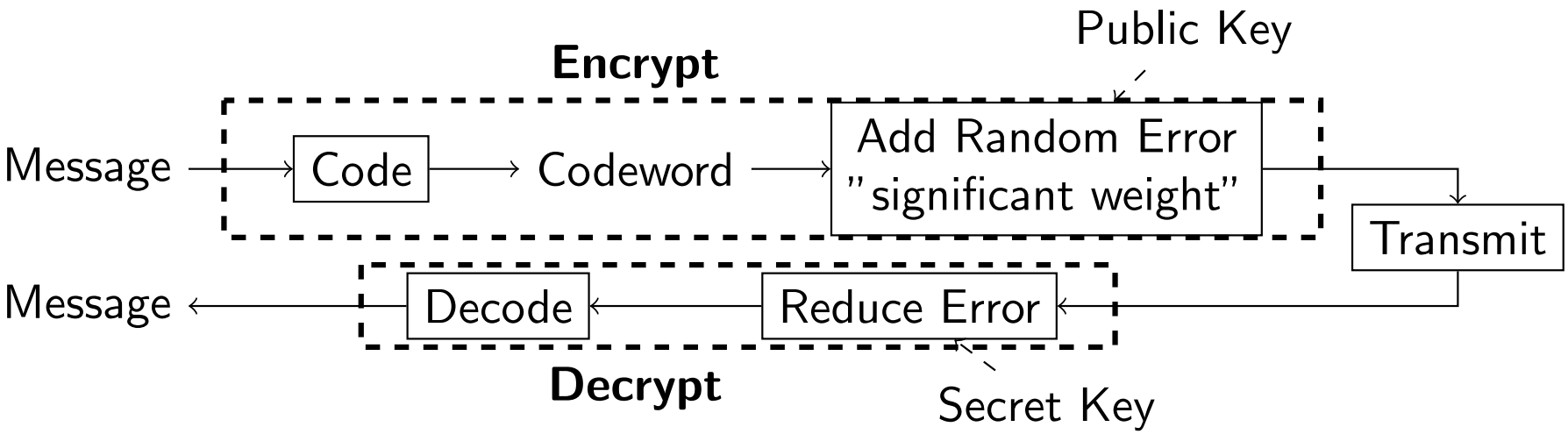


Figure: HQC PKE Framework

HQC-PKE Algorithms

$$\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1),\ \mathcal{R}_\omega = \{\mathbf{x} \in \mathcal{R} \mid \mathrm{wt}(\mathbf{x}) = \omega\}$$

Algorithm Keygen

Input: param
Output: $(\mathsf{pk}, \mathsf{sk})$

1: $\mathbf{h} \xleftarrow{\$} \mathcal{R}$
2: $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{R}_\omega^2$
3: $\mathbf{s} = \mathbf{x} + \mathbf{hy}$
4: $\mathsf{pk} = (\mathbf{h}, \mathbf{s})$
5: $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$

Algorithm Encrypt

Input: $(\mathsf{pk}, \mathbf{m})$, param
Output: ciphertext $\mathbf{c}$

1: $\mathbf{e} \xleftarrow{\$} \mathcal{R}_{\omega_e}$
2: $(\mathbf{r}_1, \mathbf{r}_2) \xleftarrow{\$} \mathcal{R}_{\omega_r}^2$
3: $\mathbf{u} = \mathbf{r}_1 + \mathbf{hr}_2$
4: $\mathbf{v} = \mathbf{m}G + \mathbf{sr}_2 + \mathbf{e}$
5: $\mathbf{c} = (\mathbf{u}, \mathbf{v})$

Figure: HQC-PKE Algorithms [AMAB+, AMAB+17]

HQC-PKE Decryption Algorithm

Algorithm Decrypt

Input: $(\mathsf{sk}, \mathbf{c})$
Output: $\mathbf{m}$

1: $\mathbf{m} = \mathcal{C}.\mathsf{Decode}(\mathbf{v} - \mathbf{uy})$

- Security does not rely on the choice of $\mathcal{C}$
- Authors of HQC proprose a concatenated Reed-Muller and Reed-Solomon Codes for $\mathcal{C}$
- Hofheinz-Hövelmanns-Kiltz [HHK17] (Fujisaki-Okamoto like [FO99, FO13]) transformation is applied PKE $\rightarrow$ KEM

Simplified HQC codes Framework

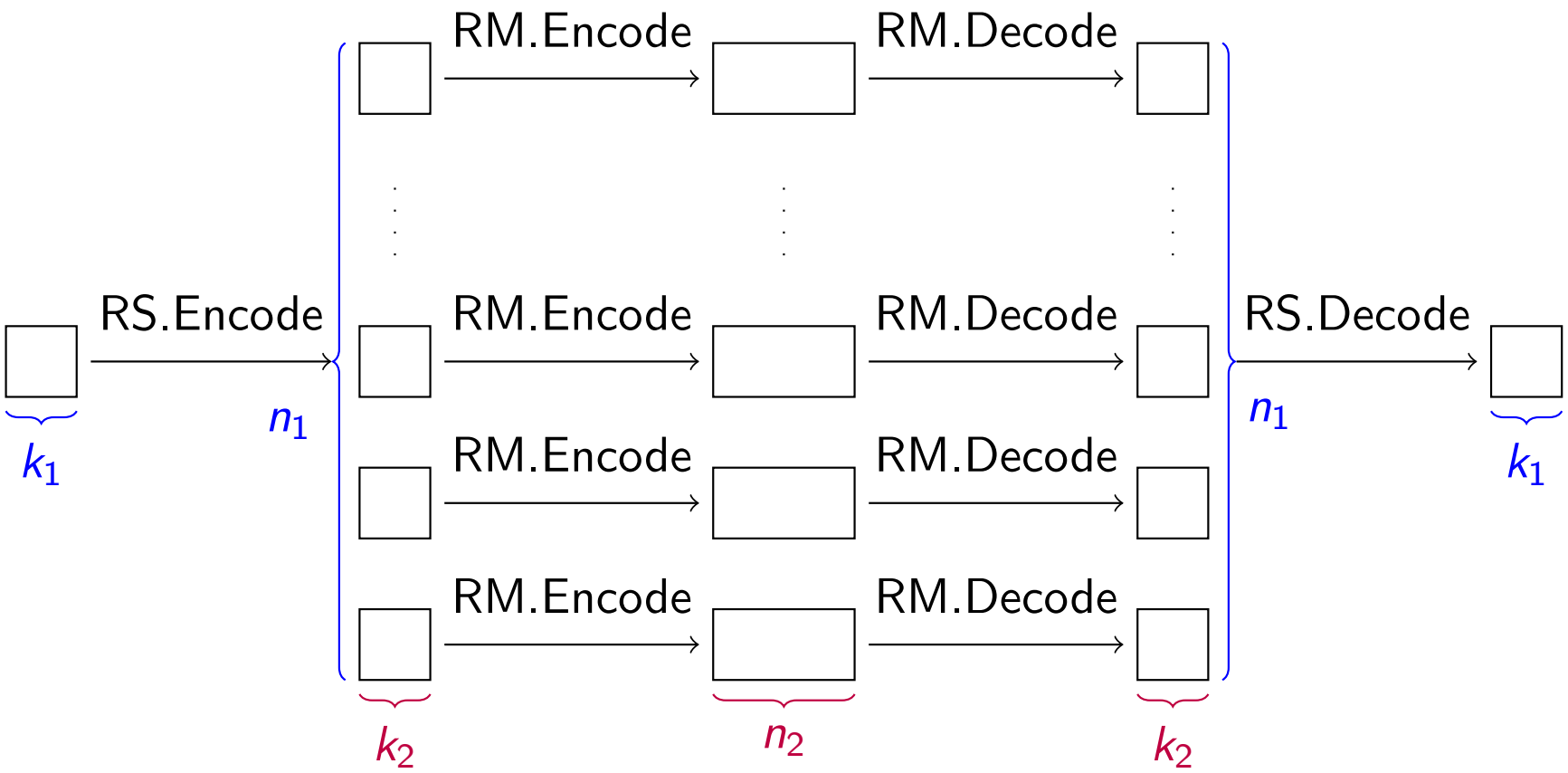


Figure: Simplified HQC Concatenated RMRS Codes Framework

Reed-Muller Codes

- RM(1, 7) is used for all security levels.

$$[128, 8, 64] \text{ code over } \mathbb{F}_2$$

- The encoding is done by the multiplication with the generator matrix.
- They are duplicated 3 (HQC-128) or 5 (HQC-192 and HQC-256) times to obtain:

$$[\underbrace{384}_{n_2}, \underbrace{8}_{k_2}, 64] \text{ or } [\underbrace{640}_{n_2}, \underbrace{8}_{k_2}, 64] \text{ codes over } \mathbb{F}_2$$

Decoding First Order Reed-Muller Codes

- First Order RM are seen as Hadamard Codes.

1. Remove the multiplicity of codewords with the expand and sum function.
2. Apply the Fast Hadamard Transform (FHT).
3. Recover the message with the find peaks function.

Expand and Sum

Algorithm Expand and sum

Input: codeword $\mathbf{c}$ and the multiplicity mul.
Output: expanded codeword $\mathbf{c}'$

1: $\mathbf{c}' = \mathbf{0} \in \mathbb{N}^{128}$
2: **for** $i \in [\![0, \text{mul}]\!]$ **do**
3: **for** $j \in [\![0, 128]\!]$ **do**
4: $\mathbf{c}'[j] \mathrel{+}= \mathbf{c}[128 \times i + j]$
5: **end for**
6: **end for**
7: **return** $\mathbf{c}'$

Fast Hadamard Transform (FHT) Theory

- RM cosets distribution = Hadamard Transform application [MS77]
- Decoding with maximum likelihood strategy.
- Decoding RM$(1, m)$ code $\rightarrow$ vector matrix multiplication with Hadamard matrix H_{2^m}

$$H_{2m} = \begin{pmatrix} H_m & H_m \\ H_m & -H_m \end{pmatrix}, \; H_1 = 1$$

- $2^m \times 2^m$ additions and subtractions.
- $H_{2^m} \rightarrow$ product of m $2^m \times 2^m$ sparse matrices.

$$H_{2^m} = M_{2^m}^{(1)} M_{2^m}^{(2)} \cdots M_{2^m}^{(m)}, \; M_{2^m}^{(i)} = I_{2^{m-i}} \otimes H_2 \otimes I_{2^{i-1}}, \; 1 \leq i \leq m$$

- I_n the identity matrix of size $n \times n$.
- $\otimes$ the Kronecker product.

Fast Hadamard Transform (FHT) Algortihm

Algorithm Fast Hadamard Transform (FHT)

Input: expanded codeword $\mathbf{c}$ and the multiplicity mul.
Output: expanded codeword transformed structure $\mathbf{c}$

1: **for** $pass \in [\![0, 6]\!]$ **do** ▷ $\mathbf{m} = 7$
2: **for** $i \in [\![0, 63]\!]$ **do**
3: $\mathbf{d}[i] = \mathbf{c}[2i] + \mathbf{c}[2i + 1]$
4: $\mathbf{d}[i + 64] = \mathbf{c}[2i] - \mathbf{c}[2i + 1]$
5: **end for**
6: swapp$(\mathbf{d}, \mathbf{c})$ ▷ copy $\mathbf{d}$ in $\mathbf{c}$ and $\mathbf{c}$ in $\mathbf{d}$
7: **end for**
8: $\mathbf{c}'[0] -= 64 * \text{mul}$
9: **return c**

HQC-KEM Algorithms – IND CCA2 Security

Algorithm Encaps

Input: $\mathsf{pk} = (\mathbf{h}, \mathbf{s})$
Output: $(\mathbf{c}, \mathbf{d})$

1: $\mathbf{m} \xleftarrow{\$} \mathbb{F}_2^k$
2: $\theta = \mathcal{G}(\mathbf{m})$ ▷ seed
3: $\mathbf{c} = \mathsf{Encrypt}(\mathbf{m}, \mathsf{pk}, \theta)$
4: $K = \mathcal{K}(\mathbf{m}, \mathbf{c})$
5: $\mathbf{d} = \mathcal{H}(\mathbf{m})$
6: **return** $(\mathbf{c}, \mathbf{d})$

Algorithm Decaps

Input: $\mathbf{c}, \mathbf{d}, \mathsf{sk}, \mathsf{pk}$
Output: shared key K or $\perp$

1: $\mathbf{m'} = \mathsf{Decrypt}(\mathbf{c}, \mathsf{sk})$
2: $\theta' = \mathcal{G}(\mathbf{m}')$ ▷ seed
3: $\mathbf{c}' = \mathsf{Encrypt}(\mathbf{m}', \mathsf{pk}, \theta')$
4: **if** $\mathbf{c} \neq \mathsf{c}$ or $\mathbf{d} \neq \mathcal{H}(\mathbf{m}')$ **then**
5: **return** $\perp$
6: **else**
7: **return** $K = \mathcal{K}(\mathbf{m}, \mathbf{c})$
8: **end if**

Figure: HQC-KEM Algorithms [AMAB+, AMAB+17] (Same key gen. as PKE)

Attack Scenario

- Goal: Recover the static secret key $\mathbf{y} \in \mathbb{F}_2^n$ with $\mathrm{wt}(\mathbf{y}) = \omega$
- Focus on the function

$$\mathcal{C}.\mathrm{Decode}(\mathbf{v} - \mathbf{uy})$$

- Choosing $(\mathbf{u}, \mathbf{v}) = (1,0)$ leads to compute $\mathcal{C}.\mathrm{Decode}(\mathbf{y})$
  - IND-CCA2 Security $\rightarrow$ Invalid ciphertext
- Choose the $\mathbf{v}$ value in order to find collisions with $\mathbf{y}$
- Oracle $\mathcal{O}_b^{\mathrm{RM}}$ that determine the number of corrected error by the RM decoder.
  - Goal : Reach $\widetilde{\mathbf{v}} = \mathbf{y}$ and then decode

$$\mathcal{C}.\mathrm{Decode}(\widetilde{\mathbf{v}} - \mathbf{y}) = \mathcal{C}.\mathrm{Decode}(0)$$

Support Probability Distribution between $\mathbf{y}'$ and $\mathbf{y}''$

- The secret key $\mathbf{y}$ has n bits but only $n_1 n_2$ are manipulated by the decoder. $l = n - n_1 n_2$ bits are truncated.
- $\mathbf{y} = (\mathbf{y}', \mathbf{y}'')$ with $\mathbf{y}' \in \mathbb{F}_2^{n_1 n_2}$ and $\mathbf{y}'' \in \mathbb{F}_2^{l}$.

$$Q_k = \mathbb{P}\left(\mathrm{wt}(\mathbf{y}'') = k\right) \cong \binom{\omega}{k} p^k (1-p)^{\omega - k}, \; p = \frac{|\mathbf{y}'|}{|\mathbf{y}|}$$

λ	n	$n_1 n_2$	ω	Q_0	Q_1	Q_2	$Q_{\geq 2}$
128	17.669	17.664	66	$98,15\%$	$1,83\%$	$0,02\%$	$\leq 10^{-3}\%$
192	35.851	35.840	100	$96,98\%$	$2,98\%$	$0,05\%$	$\leq 10^{-3}\%$
256	57.637	57.600	131	$91,93\%$	$7,73\%$	$0,32\%$	$\leq 10^{-2}\%$

Support Probability Distribution among the blocs of $\mathbf{y}'$

- RM decoder manipulates n_1 blocs of size n_2

λ	n_1	n_2	ω
128	46	384	66
192	56	640	100
256	90	640	131

$$\mathbf{y}' = \left(\mathbf{y}'_0, \mathbf{y}'_1, \cdots, \mathbf{y}'_{n_1-1}\right) \text{ and for all } i,\ \mathbf{y}'_i \in \mathbb{F}_2^{n_2}$$

$$P_k = \mathbb{P}\left(\mathrm{wt}(\mathbf{y}'_i) = k \;\middle|\; \mathbf{y} \xleftarrow{\$} \mathcal{R}_\omega,\ i \xleftarrow{\$} [\![0, n_1 - 1]\!]\right)$$

λ	P_0	P_1	P_2	P_3	P_4	$P_{\geq 5}$
128	23,44%	34,38%	24,83%	11,77%	4,12%	1,45%
192	16,50%	30,00%	27,00%	16,04%	7,07%	3.40%
256	23,14%	34,06%	24,87%	12,02%	4,32%	1,59%

Fast Hadamard Transform (FHT) Behavior

$$\begin{pmatrix}0\\\vdots\\0\\0\\0\\\vdots\\0\\0\\0\\\vdots\\0\end{pmatrix} \xrightarrow{\text{FHT}} \begin{pmatrix}0\\\vdots\\0\\0\\0\\\vdots\\0\\0\\0\\\vdots\\0\end{pmatrix} \qquad \begin{pmatrix}0\\\vdots\\0\\1\\0\\\vdots\\0\\0\\0\\\vdots\\0\end{pmatrix} \xrightarrow{\text{FHT}} \begin{pmatrix}0\\\vdots\\1\\1\\1\\\vdots\\-1\\-1\\-1\\\vdots\\0\end{pmatrix} \qquad \begin{pmatrix}0\\\vdots\\0\\1\\0\\\vdots\\0\\1\\0\\\vdots\\0\end{pmatrix} \xrightarrow{\text{FHT}} \begin{pmatrix}0\\\vdots\\2\\2\\2\\\vdots\\-2\\-2\\-2\\\vdots\\0\end{pmatrix} \qquad \begin{pmatrix}0\\\vdots\\1\\1\\0\\\vdots\\0\\1\\0\\\vdots\\0\end{pmatrix} \xrightarrow{\text{FHT}} \begin{pmatrix}0\\\vdots\\3\\3\\3\\\vdots\\-3\\-3\\-3\\\vdots\\0\end{pmatrix}$$

Fast Hadamard Transform (FHT) Behavior with HME

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \xrightarrow{\text{FHT}} \begin{pmatrix} 0 \\ \vdots \\ 2 \\ 2 \\ 2 \\ \vdots \\ -2 \\ -2 \\ -2 \\ \vdots \\ 0 \end{pmatrix} \qquad \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 2 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \xrightarrow{\text{FHT}} \begin{pmatrix} 0 \\ \vdots \\ 2 \\ 2 \\ 2 \\ \vdots \\ -2 \\ -2 \\ -2 \\ \vdots \\ 0 \end{pmatrix}$$

Higher Magnitude Errors (HME)

- Expand and sum algorithm can create a collision

$$\begin{aligned}\mathbb{P}(\text{HME}) &= \sum_{k=0}^{n_2} \mathbb{P}(\text{wt}(\mathbf{y}'_i) = k) \times \mathbb{P}(\text{HME} \mid \text{wt}(\mathbf{y}'_i) = k) \\ &= \sum_{k=0}^{n_2} P_k \times \left(1 - \mathbb{P}\left(\overline{\text{HME}} \mid \text{wt}(\mathbf{y}'_i) = k\right)\right) \\ &= \sum_{k=0}^{n_2} P_k \times \left(1 - \prod_{i=0}^{k} \frac{n_2 - (\text{mul} - 1) \times i}{n_2}\right)\end{aligned}$$

- An HME happens in vector $\mathbf{y}'$ with probabilities $0,53\%$, $0,97\%$ and $0,65\%$ for respectively HQC-128, HQC-192 and HQC-256.

Attack Description

- n_1 decoding Oracles $\mathcal{O}^{\text{RM}}_{i,b}$ able to determine the number of errors corrected by the RM decoder in the ith bloc $\mathbf{y}'_i$ for $i \in [\![0, n_1 - 1]\!]$

$$(1, 0) \longrightarrow \mathcal{C}.\text{Decode}(\mathbf{y})$$

$$(1, 0) \xrightarrow{\mathcal{O}^{\text{RM}}_b} (b_0, b_1, \cdots, b_{n_1-1}) \text{ [reference values]}$$

- select $\mathbf{v}$ and call again the Oracle

$$(1, \mathbf{v}) \longrightarrow \mathcal{C}.\text{Decode}(\mathbf{v} - \mathbf{y})$$

$$(1, \mathbf{v}) \xrightarrow{\mathcal{O}^{\text{RM}}_b} (b'_0, b'_1, \cdots, b'_{n_1-1}) \text{ [attack values]}$$

- GOAL: find $\widetilde{\mathbf{v}} = \mathbf{y}$

$$(1, \widetilde{\mathbf{v}}) \xrightarrow{\mathcal{O}^{\text{RM}}_b} (0, 0, \cdots, 0)$$

Attack Description II

- Focus on a given bloc $\mathbf{y}'_j$
- Try all $\mathbf{v}_j$ of hamming weight 1
  1. $\mathrm{Supp}(\mathbf{y}'_j) \cap \mathrm{Supp}(\mathbf{v}_j) = \mathrm{Supp}(\mathbf{v}_j)$. Then $\mathrm{wt}(\mathbf{v}_j - \mathbf{y}'_j) = \mathrm{wt}(\mathbf{y}'_j) - 1$, the decoder will correct one error less than the reference decoding of $\mathbf{y}'_j$.

$$\mathcal{O}^{\mathrm{RM}}_{j,b}(\mathbf{v} - \mathbf{y}) = O^{\mathrm{RM}}_{j,b}(\mathbf{y}) - 1$$

  2. $\mathrm{Supp}(\mathbf{y}'_j) \cap \mathrm{Supp}(\mathbf{v}_j) = \varnothing$. Then $\mathrm{wt}(\mathbf{v}_j - \mathbf{y}_j) = \mathrm{wt}(\mathbf{y}_j) + 1$, the decoder will correct one error more than the reference decoding of $\mathbf{y}$.

$$\mathcal{O}^{\mathrm{RM}}_{j,b}(\mathbf{v} - \mathbf{y}) = O^{\mathrm{RM}}_{j,b}(\mathbf{y}) + 1$$

- Remember the locations where the Oracle outputs 1 less than the reference value

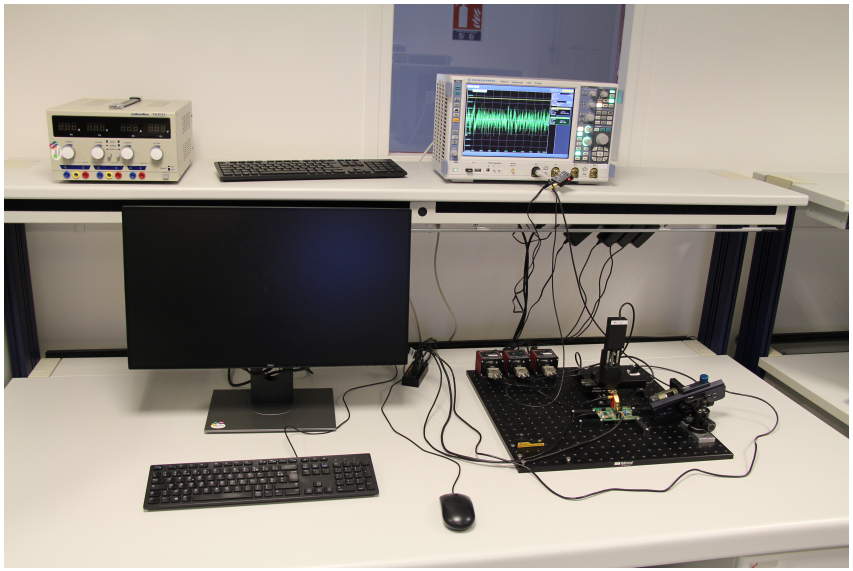
Divide and Conquer Strategy

- Blocs decoding are independent
- Queries on each bloc can be done at the same time
- Number of queries = number of bits in a single blocs $\mathbf{y}'_i = n_2$

λ	$n_1 n_2$	n_1	n_2	ω
128	17.664	46	384	66
192	35.840	56	640	100
256	57.600	90	640	131

- Remark Total number of attack traces = number of queries * number of traces requiered to classify with accuracy 1.

Building 6 Classes



$$\text{Classe } i = \left\{ EM\left[\text{FHT}(\mathbf{x})\right]) \middle| \mathbf{x} = EAS(\mathbf{X}),\ \mathbf{X} \xleftarrow{\$} \mathbb{F}_2^{3\times n_2},\ \text{wt}(\mathbf{X}) = i \right\}\ 0 \leq i \leq 5$$

- 50.000 electromagnetic measurement per class.
- Randomness provided by the microcontroller
- Same HME distribution as in real case

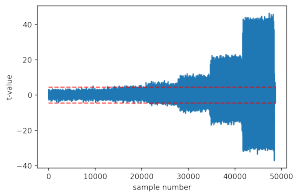
T-test Leakage Assessment

- Two sets S_0 and S_1 with Cardinalities n_0 and n_1, means μ_0 and μ_1 and variances σ_0 and σ_1

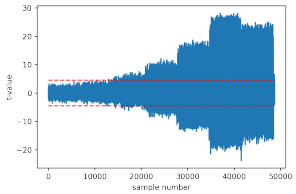
$$t\text{-value} = \frac{\mu_0 - \mu_1}{\sqrt{\left(\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1}\right)}}$$

- Threshold $|t| \geq 4.5$
- if $|t| \geq 4.5 \rightarrow$ Statistical difference with confidence 99.9999%
- if $|t| < 4.5 \rightarrow$ No (first order) leakage with confidence 99.9999%

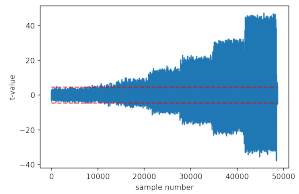
T-test Leakage Assessment



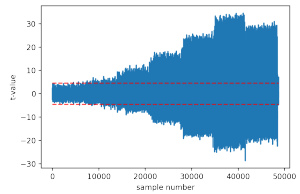
(a) Cl. 0 and 1



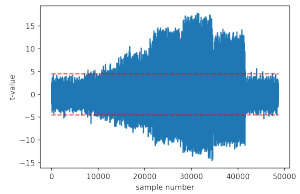
(b) Cl. 0 and 2



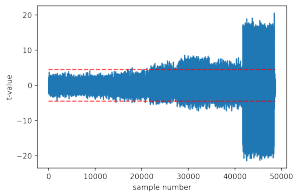
(c) Cl. 0 and 3



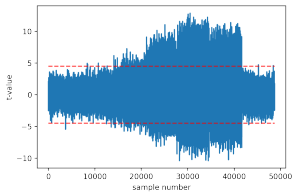
(d) Cl. 0 and 4



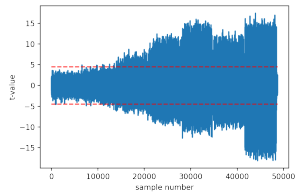
(e) Cl. 0 and 5



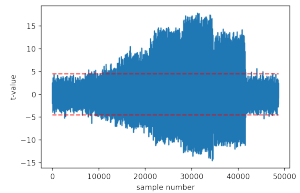
(f) Cl. 1 and 2



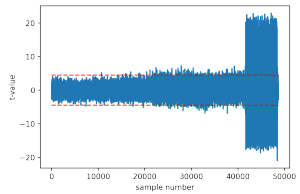
(g) Cl. 1 and 3



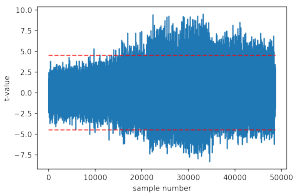
(h) Cl. 1 and 4



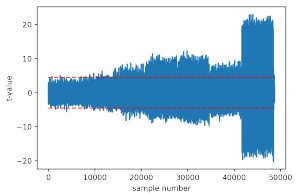
(i) Cl. 1 and 5



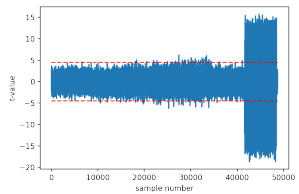
(j) Cl. 2 and 3



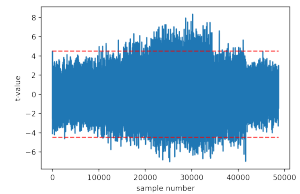
(k) Cl. 2 and 4



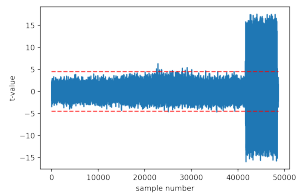
(l) Cl. 2 and 5



(m) Cl. 3 and 4

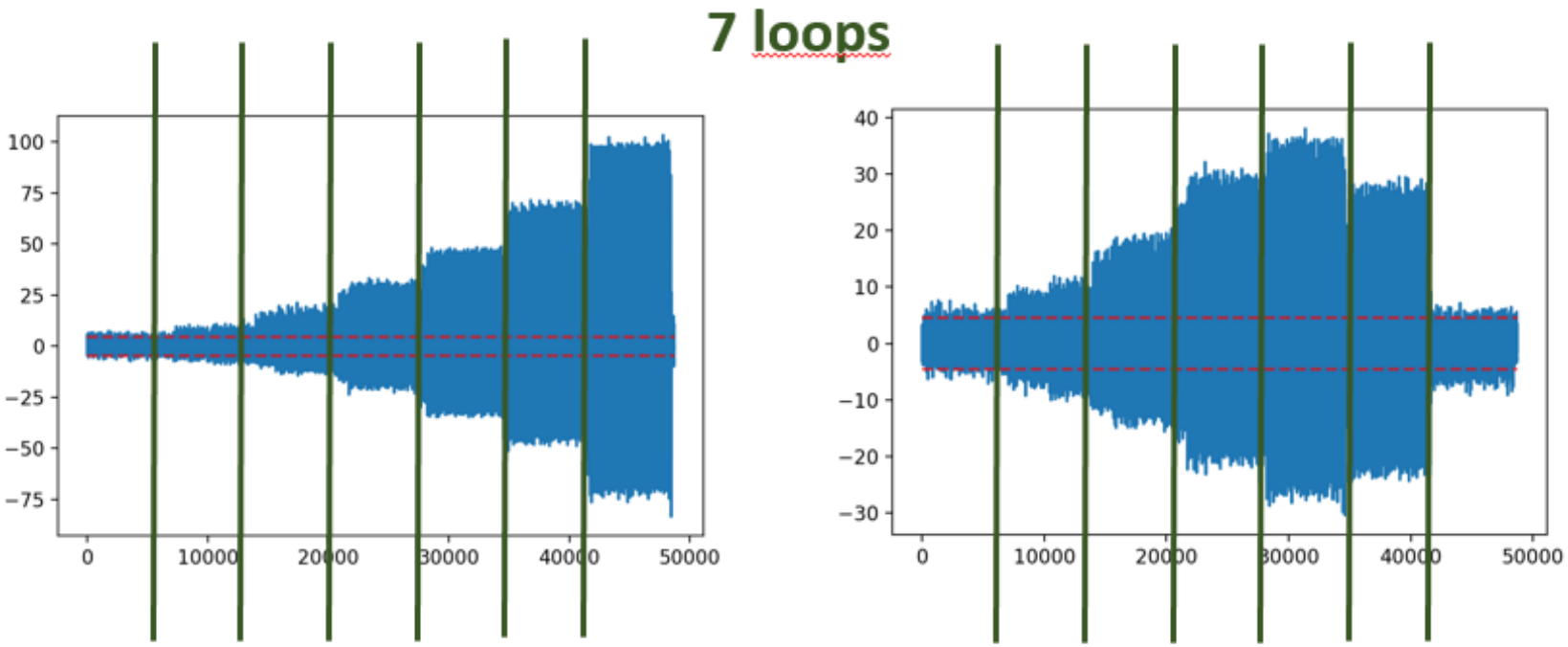


(n) Cl. 3 and 5

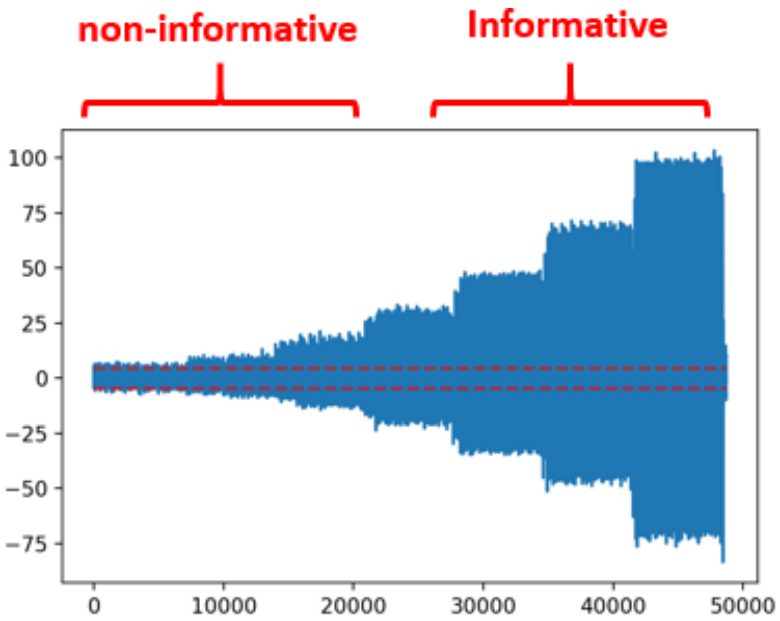


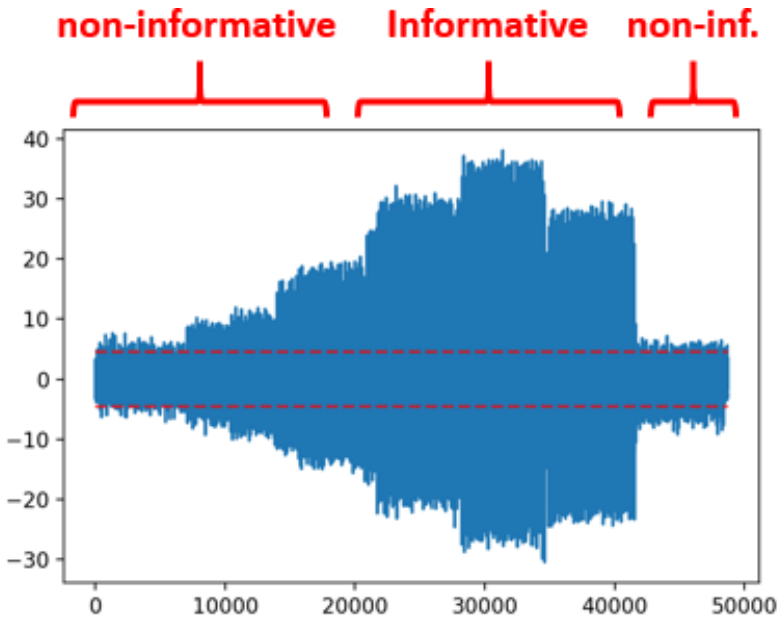
(o) Cl. 4 and 5

Results


7 loops
100
75
50
25
0
−25
−50
−75
0
10000
20000
30000
40000
50000
40
30
20
10
0
−10
−20
−30
0
10000
20000
30000
40000
50000

Results


non-informative
Informative
100
75
50
25
0
−25
−50
−75
0
10000
20000
30000
40000
50000

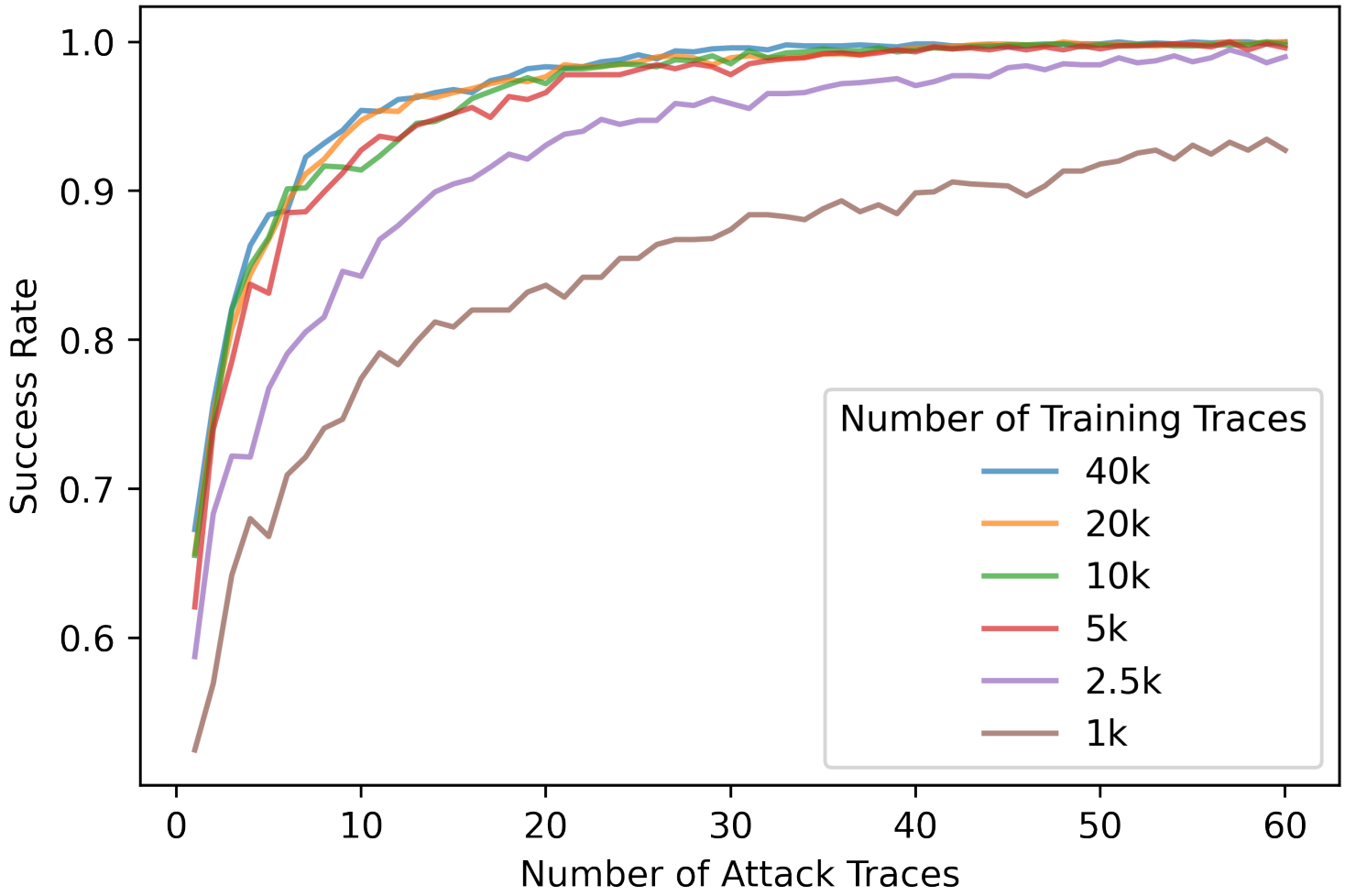

non-informative
Informative
non-inf.
40
30
20
10
0
−10
−20
−30
0
10000
20000
30000
40000
50000

LDA

- Use of the Linear Discriminant Analisys (LDA) [Linear Classifier]
- We set the number training traces (1k to 40k per classes).
- To increase the accuracy, we send k traces from the same class to the Oracle.
- We reconcile the results with a soft-max strategy (with $\tau = 6$ the number of classes):

$$\text{argmax}(p_1, \cdots, p_\tau) = \text{argmax}(\sum_{i=1}^{k}(p_{1,i}, \cdots, p_{\tau,i}))$$

Results


1.0
0.9
0.8
0.7
0.6
Success Rate
0
10
20
30
40
50
60
Number of Attack Traces
Number of Training Traces
40k
20k
10k
5k
2.5k
1k

Cost

- 50 attack traces are enough to obtain 100% accuracy
- Each bloc of $\mathbf{y}$ can be recover indendently and at the same time.
- Total Number of attack traces: $50 \times n_2 = 50 \times 384 = 19.200$ for HQC-128.

Masking-based Countermeasure

- Using a Mask
- Hide Sensitive data by dividing the knwoledge in n shares

$$\mathbf{c} = \sum_{i=0}^{n} \mathbf{c}_i$$

- Linearity of the Hadamard Transform.

$$\mathrm{FHT}(\mathbf{c}) = \sum_{i=0}^{n} \mathrm{FHT}(\mathbf{c}_i)$$

- $n-1$ first $\mathbf{c}_i$ must be sampled uniformly at random
- $\mathbf{c}_n$ is chosen to satisfy the relation.
- **COST:** Compute $n+1$ times the FHT.

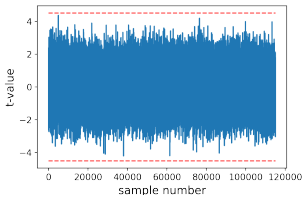
Countermeasure Algorithm

Algorithm Hadamard Transform with first order mask

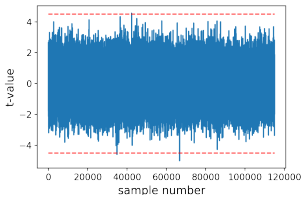
Input: expanded codeword $\mathbf{c}$ and the multiplicity mul.
Output: expanded codeword transformed structure $\mathbf{c}$

1: $\mathbf{c}_0 \xleftarrow{\$}$ expanded codeword
2: $\mathbf{c}_1 = c - c_0$
3: $\mathbf{c}_0 = \mathrm{FHT}(\mathbf{c}_0)$
4: $\mathbf{c}_1 = \mathrm{FHT}(\mathbf{c}_1)$
5: $\mathbf{c} = \mathbf{c}_0 + \mathbf{c}_1$
6: **return** $\mathbf{c}$

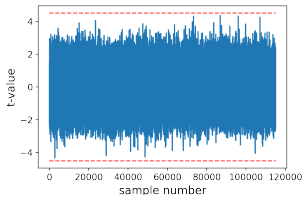
Countermeasure T-test Leakage Assessment



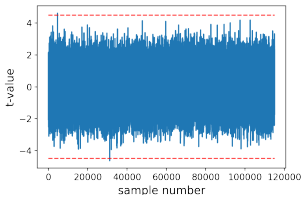
(a) Cl. 0 and 1



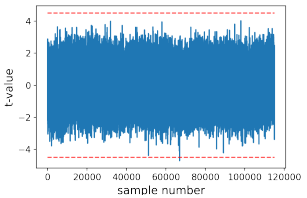
(b) Cl. 0 and 2



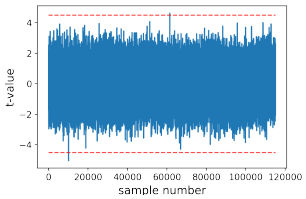
(c) Cl. 0 and 3



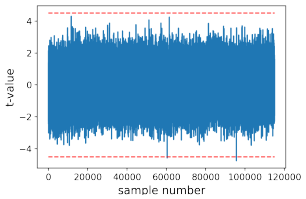
(d) Cl. 0 and 4



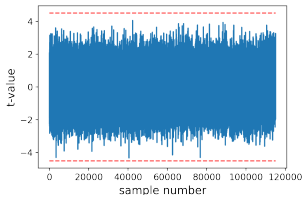
(e) Cl. 0 and 5



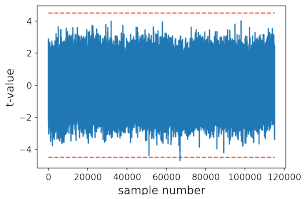
(f) Cl. 1 and 2



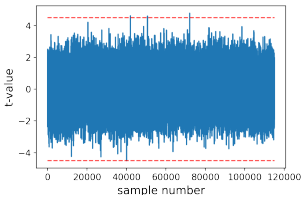
(g) Cl. 1 and 3



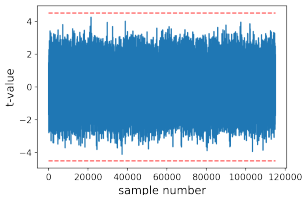
(h) Cl. 1 and 4



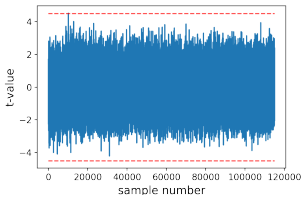
(i) Cl. 1 and 5



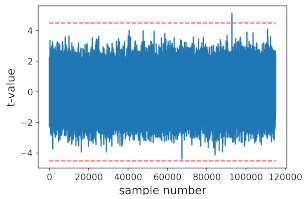
(j) Cl. 2 and 3



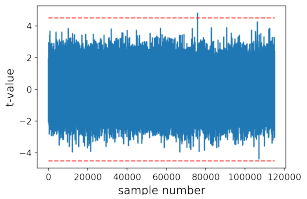
(k) Cl. 2 and 4



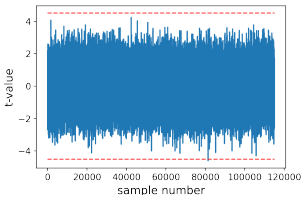
(l) Cl. 2 and 5



(m) Cl. 3 and 4



(n) Cl. 3 and 5



(o) Cl. 4 and 5

Conclusion

- I presented:
  - New SCA against RMRS HQC
  - Simple Strategy to recover the secret key
  - Less than 20.000 attack traces
  - Threat for HQC $\rightarrow$ Countermeasure (local)
- Future works:
  - Reduce the number of requiered attack traces
  - Target other HQC function for a new Oracle

Thank you for your attention

Bibliography I


Carlos Aguilar-Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, and Gilles Zémor.
HQC reference implementation.


Carlos Aguilar-Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, and Gilles Zémor.
Hamming Quasi-Cyclic (HQC), 2017.


Eiichiro Fujisaki and Tatsuaki Okamoto.
Secure integration of asymmetric and symmetric encryption schemes.

In *Annual International Cryptology Conference*, pages 537–554. Springer, 1999.

Bibliography II


Eiichiro Fujisaki and Tatsuaki Okamoto.
Secure integration of asymmetric and symmetric encryption schemes.

Journal of cryptology, 26(1):80–101, 2013.


Qian Guo, Clemens Hlauschek, Thomas Johansson, Norman Lahr, Alexander Nilsson, and Robin Leander Schröder.
Don't reject this: Key-recovery timing attacks due to rejection-sampling in hqc and bike.
Cryptology ePrint Archive, 2021.


Guillaume Goy, Antoine Loiseau, and Philippe Gaborit.
Estimating the strength of horizontal correlation attacks in the hamming weight leakage model: A side-channel analysis on hqc kem.

2022.

Bibliography III


Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz.
A modular analysis of the fujisaki-okamoto transformation.
In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography*, pages 341–371, Cham, 2017. Springer International Publishing.


Florence Jessie MacWilliams and Neil James Alexander Sloane.
The theory of error correcting codes, volume 16.
Elsevier, 1977.


Thales Bandiera Paiva and Routo Terada.
A timing attack on the HQC encryption scheme.
In *International Conference on Selected Areas in Cryptography*, pages 551–573. Springer, 2019.


Leander Schröder.
A novel timing side-channel assisted key-recovery attack against HQC.
PhD thesis, Wien, 2022.

Bibliography IV

Thomas Schamberger, Lukas Holzbaur, Julian Renner, Antonia Wachter-Zeh, and Georg Sigl.
A power side-channel attack on the reed-muller reed-solomon version of the hqc cryptosystem.
Cryptology ePrint Archive, 2022.

Thomas Schamberger, Julian Renner, Georg Sigl, and Antonia Wachter-Zeh.
A power side-channel attack on the CCA2-Secure HQC KEM.
In *19th Smart Card Research and Advanced Application Conference (CARDIS2020)*, 2020.

Guillaume Wafo-Tapa, Slim Bettaieb, Loïc Bidoux, Philippe Gaborit, and Etienne Marcatel.
A practicable timing attack against HQC and its countermeasure.
Advances in Mathematics of Communications, 2020.