

A Reaction Attack against Cryptosystems based on LRPC Codes

Gustavo Banegas

joint work with

Simona Samardjiska, Paolo Santini and Edoardo Persichetti

TU/e



CHALMERS
UNIVERSITY OF TECHNOLOGY



Radboud Universiteit Nijmegen



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FAU
FLORIDA ATLANTIC
UNIVERSITY

LatinCrypt 2019
October 3rd, 2019

Outline

Introduction

Reaction Attack

Our Result

Conclusion

Post-quantum cryptography

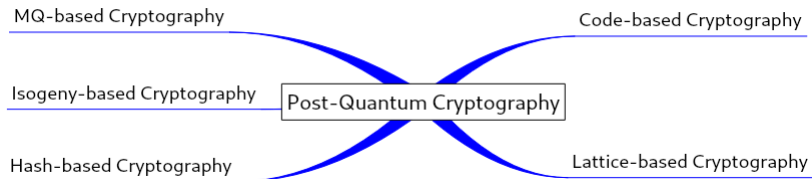
Why do we need post-quantum cryptography?

Shor's Algorithm solves in polynomial time:

- ▶ Integer factorization; RSA is dead.
- ▶ The discrete-logarithm problem in finite fields; DSA is dead.
- ▶ The discrete-logarithm problem on elliptic curves; ECDSA is dead.

Post-quantum cryptography

What is post-quantum cryptography?



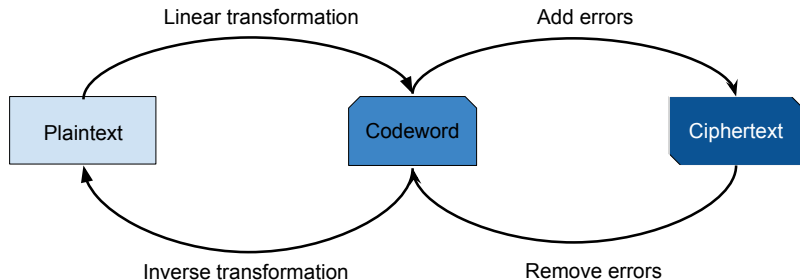
Post-quantum cryptography

Timeline

- ▶ 2016: NIST calls for submissions to “Post-Quantum Cryptography Standardization Project”.
- ▶ 2017: NIST receives 69 proper submissions.
- ▶ 2018-19: NIST 2nd round of proposals with 26 proposals.
- ▶ 17 code-based in the 1st round; 7 code-based in 2nd round.

Code-based Cryptography

Code-based cryptography in a nutshell



- ▶ Originally proposed by McEliece in 1978;
- ▶ It uses a linear code:
 - ▶ Goppa codes;
 - ▶ LDPC/MDPC;
 - ▶ Rank Metric (LRPC);
 - ▶ Several others.

Rank Metric Codes

A Low-Rank Parity-Check (LRPC) code

A LRPC \mathcal{C} over \mathbb{F}_{q^m} of length n , dimension k and rank d is described by an $(n - k) \times n$ parity-check matrix

$$\mathbf{H} = \{h_{i,j}\} \in \mathbb{F}_{q^m}^{(n-k) \times n},$$

- ▶ Each coefficient $h_{i,j}$ can be written as

$$h_{i,j} = \sum_{l=1}^d h_{i,j,l} F_l, \quad h_{i,j,l} \in \mathbb{F}_q,$$

each $F_i \in \mathbb{F}_{q^m}$, and $F = \langle F_1, F_2, \dots, F_d \rangle$ is a \mathbb{F}_q subspace of \mathbb{F}_{q^m} .

Decoding LRPC codes

How to decode LRPC codes?

Let $\mathbf{s} = (s_1, \dots, s_{n-k}) \in \mathbb{F}_{q^m}^{n-k}$ be the *syndrome* of \mathbf{e} , i.e. $\mathbf{H}\mathbf{e}^T = \mathbf{s}$.

Decoding: Recover \mathbf{e} from the knowledge of \mathbf{s} .

Crucial facts:

- ▶ If $h_{i,j} \in F = \langle F_1, F_2, \dots, F_d \rangle$ and $e \in E = \langle E_1, E_2, \dots, E_r \rangle$ then

$$s_i \in \langle F_1 E_1, F_1 E_2, \dots, F_d E_r \rangle$$

- ▶ Assume $S = \langle s_1, s_2, \dots, s_{n-k} \rangle = \langle F_1 E_1, F_1 E_2, \dots, F_d E_r \rangle$ then:

1. Set $S_i = F_i^{-1} \cdot S$. Then

$$S_i = F_i^{-1} \cdot \langle \dots F_i E_1, F_i E_2, \dots, F_i E_r \dots \rangle \Rightarrow E = \langle E_1, E_2, \dots, E_r \rangle \subset S_i$$

2. Find $E = S_1 \cap S_2 \cap \dots \cap S_d$
3. Find \mathbf{e} by solving $\mathbf{H}\mathbf{e}^T = \mathbf{s}$

Decoding of LRPC codes

When do decoding failures happen?

1. When $\text{Dim}(\langle EF \rangle) < rd$: this happens with probability
$$P_1 = \frac{d}{q^{m-rd}}$$
 2. When $E \neq \bigcap_{i=1}^d S_i$: when $m > rd + 8$, this happens with probability $P_2 \ll 2^{-30}$
 3. **When $\text{Dim}(S) < rd$ this happens with probability**
$$P_3 = \frac{1}{q^{n-k+1-rd}}$$
- In practice usually $P_1, P_2 \ll P_3$.

LRPC cryptosystems

What is a LRPC cryptosystem?

Basically any cryptosystem that

- ▶ uses LRPC codes (low rank of \mathbf{H}_{secret})
- ▶ uses $\mathbf{RH}_{secret} = \mathbf{H}$ to hide the secret \mathbf{H}_{secret}
- ▶ relies on the **Rank syndrome decoding problem**:

Find \mathbf{e} such that $\mathbf{He}^T = \mathbf{s}$ and $|\mathbf{e}| \leq r$.

- ▶ LRPC cryptosystem [Gaborit et al.'13]
- ▶ McNie [Kim et al.'17] (NIST 1st round candidate)
- ▶ ROLLO (Rank-Ouroboros, LAKE and LOCKER) [Aguilar Melchor et al. '17] (NIST 2nd round candidate)
- ▶ Durandal [Aragon et al.'19]

Reaction attack



Reaction attack



$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1 \mathbf{G} + \mathbf{e}_1$$

Reaction attack



$$m_1, e_1, c_1 = m_1 G + e_1 \xrightarrow{c_1}$$

Reaction attack



$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1 \mathbf{G} + \mathbf{e}_1 \xrightarrow{\mathbf{c}_1} \checkmark \leftarrow \text{Decode}(\mathbf{c}_1)$$

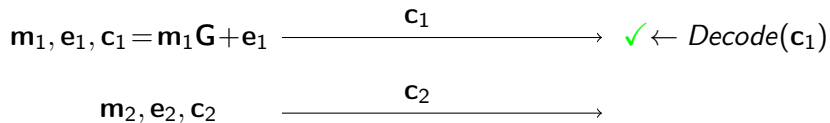
Reaction attack



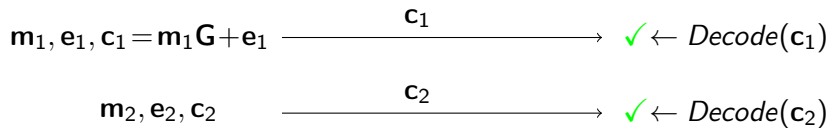
$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1 \mathbf{G} + \mathbf{e}_1 \xrightarrow{\mathbf{c}_1} \checkmark \leftarrow \text{Decode}(\mathbf{c}_1)$$

$\mathbf{m}_2, \mathbf{e}_2, \mathbf{c}_2$

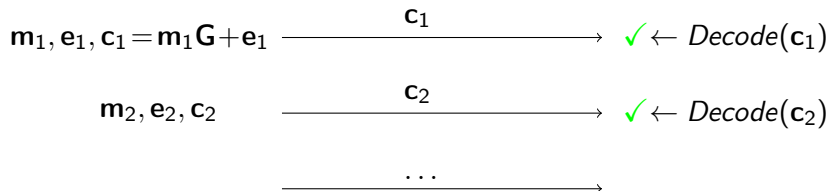
Reaction attack



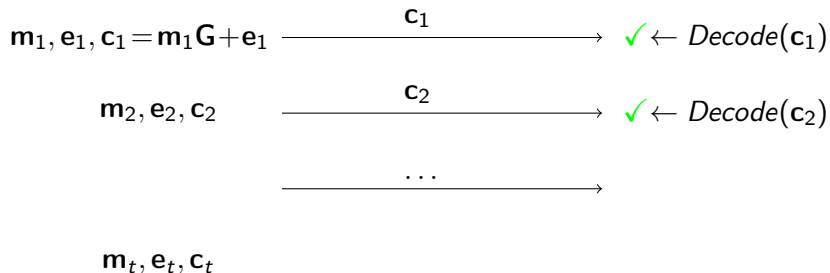
Reaction attack



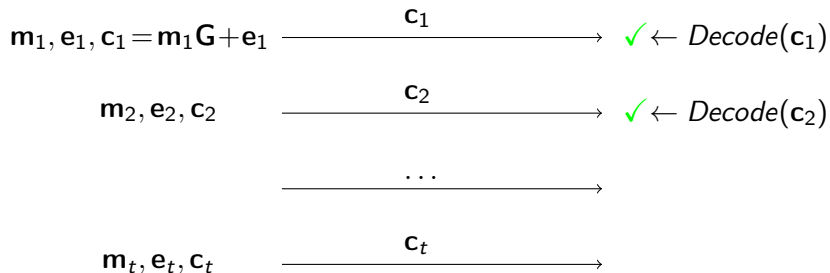
Reaction attack



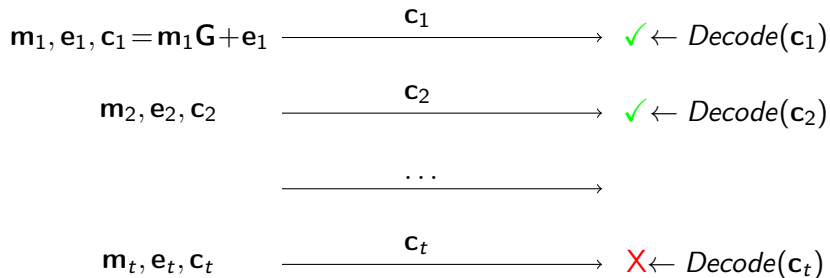
Reaction attack



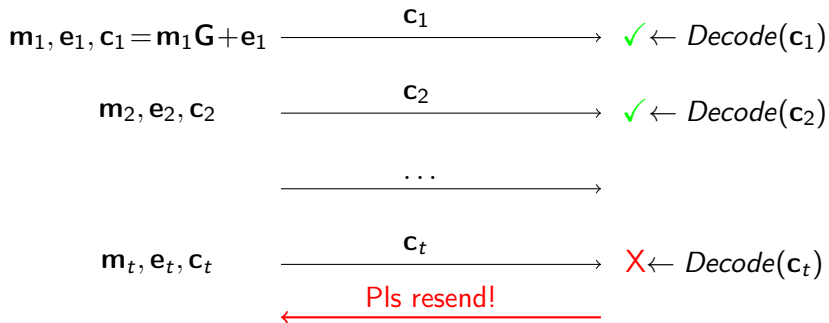
Reaction attack



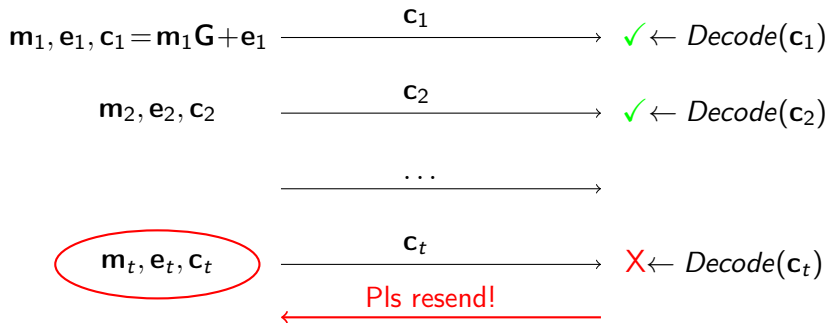
Reaction attack



Reaction attack



Reaction attack



Key recovery attack

When does a decoding failure happen?

(A closer look at) the syndrome equation for LRPC:

$$\mathbf{H}_{secret} \mathbf{e}^T = \mathbf{s}$$

$$\begin{aligned} s_i &= \sum_{j=1}^n h_{i,j} e_j = \sum_{j=1}^n \left(\sum_{l=1}^d h_{i,j,l} F_l \right) \left(\sum_{v=1}^r e_{j,u} E_u \right) \\ &= \sum_{l=1}^d \sum_{u=1}^r F_l E_u \left(\sum_{j=1}^n h_{i,j,l} e_{j,u} \right), \quad \forall i \in \{1, \dots, n - k\}. \end{aligned}$$

In matrix form:

$$\mathbf{s} = (F_1 E_1, F_1 E_2, \dots, F_d E_r) \cdot \bar{\mathbf{A}}_{\mathbf{h}, \mathbf{e}}$$

Recall: Decoding fails when $\text{Dim}(S) < rd$

$\bar{\mathbf{A}}_{\mathbf{h}, \mathbf{e}}$ is not of full rank

Our Attack

What to do with the errors?

$$\left\{ \begin{array}{l} \mathbf{v}_{e_1} \cdot \bar{\mathbf{A}}_{e_1}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \mathbf{v}_{e_2} \cdot \bar{\mathbf{A}}_{e_2}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \dots \\ \mathbf{v}_{e_t} \cdot \bar{\mathbf{A}}_{e_t}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \end{array} \right.$$

Our Attack

What to do with the errors?

$$\left\{ \begin{array}{l} \mathbf{v}_{\mathbf{e}_1} \cdot \bar{\mathbf{A}}_{\mathbf{e}_1}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \mathbf{v}_{\mathbf{e}_2} \cdot \bar{\mathbf{A}}_{\mathbf{e}_2}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \dots \\ \mathbf{v}_{\mathbf{e}_t} \cdot \bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \end{array} \right.$$

High level attack idea:

0: *Collect errors $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_t$ from decryption failures*

0: **repeat**

0: $\mathbf{h} \leftarrow \text{SolveSystem}(\mathbf{v}_{\mathbf{e}_1}, \mathbf{v}_{\mathbf{e}_2}, \dots, \mathbf{v}_{\mathbf{e}_t}, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_t)$

0: **if** $\mathbf{h} \neq \perp$ **then**

0: *Collect ℓ messages, errors, ciphertexts $(\mathbf{m}_i, \mathbf{e}_i, \mathbf{c}_i)$*

0: $F, \text{success} \leftarrow \text{FindBasis}(\mathbf{h}, \{(\mathbf{m}_i, \mathbf{e}_i, \mathbf{c}_i)\}_{i=1}^{\ell})$

0: **else** $\text{success} \leftarrow \perp$

0: **until** success

0: $\mathbf{H} \leftarrow \text{ReconstructMatrix}(\mathbf{h}, F)$

return \mathbf{H} of small rank $d = 0$

Our Attack

How to solve the system?

$$\left\{ \begin{array}{l} \mathbf{v}_{\mathbf{e}_1} \cdot \bar{\mathbf{A}}_{\mathbf{e}_1}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \mathbf{v}_{\mathbf{e}_2} \cdot \bar{\mathbf{A}}_{\mathbf{e}_2}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \dots \\ \mathbf{v}_{\mathbf{e}_t} \cdot \bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \end{array} \right.$$

▶ Kernel method

- ▶ $n - k$ equations for each error \mathbf{e}_i
- ▶ nd unknown coefficients in \mathbf{h}
- ▶ guess $\mathbf{v}_{\mathbf{e}_i}$ in kernel of $\bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h})$
- ▶ \Rightarrow linear system only in the nd \mathbf{h} -variables
- ▶ need to collect $t \geq \frac{nd}{n-k}$ errors from DF

Our Attack

How to solve the system?

$$\begin{cases} \mathbf{v}_{\mathbf{e}_1} \cdot \bar{\mathbf{A}}_{\mathbf{e}_1}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \mathbf{v}_{\mathbf{e}_2} \cdot \bar{\mathbf{A}}_{\mathbf{e}_2}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \dots \\ \mathbf{v}_{\mathbf{e}_t} \cdot \bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \end{cases}$$

► Kernel method

- $n - k$ equations for each error \mathbf{e}_i
- nd unknown coefficients in \mathbf{h}
- guess $\mathbf{v}_{\mathbf{e}_i}$ in kernel of $\bar{\mathbf{A}}_{\mathbf{e}_i}(\mathbf{h})$
- \Rightarrow linear system only in the nd \mathbf{h} -variables
- need to collect $t \geq \frac{nd}{n-k}$ errors from DF
- Probability of guessing $\mathbf{v}_{\mathbf{e}_i}$ correctly: $P_{\mathbf{e}_i} = \frac{q^{K_{\mathbf{e}_i}}}{q^{rd}}$, where $q^{K_{\mathbf{e}_i}} = |\text{Ker}(\bar{\mathbf{A}}_{\mathbf{e}_i}(h))|$.

Our Attack

How to solve the system?

$$\begin{cases} \mathbf{v}_{e_1} \cdot \bar{\mathbf{A}}_{e_1}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \mathbf{v}_{e_2} \cdot \bar{\mathbf{A}}_{e_2}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \\ \dots \\ \mathbf{v}_{e_t} \cdot \bar{\mathbf{A}}_{e_t}(\mathbf{h}) = \mathbf{0}_{1 \times n-k} \end{cases}$$

► Kernel method

- $n - k$ equations for each error \mathbf{e}_i
- nd unknown coefficients in \mathbf{h}
- guess \mathbf{v}_{e_i} in kernel of $\bar{\mathbf{A}}_{e_t}(\mathbf{h})$
- \Rightarrow linear system only in the nd \mathbf{h} -variables
- need to collect $t \geq \frac{nd}{n-k}$ errors from DF
- Probability of guessing \mathbf{v}_{e_i} correctly: $P_{e_i} = \frac{q^{K_{e_i}}}{q^{rd}}$, where $q^{K_{e_i}} = |\text{Ker}(\bar{\mathbf{A}}_{e_i}(h))|$.
- Probability of guessing all $\mathbf{v}_{e_1}, \dots, \mathbf{v}_{e_t}$: $P_t = P_{e_i}^t = q^{-(rd-1)t}$.

Our Attack

Can it be improved?

An LRPC cryptosystem, with a secret key $sk = (\mathbf{H}, \cdot)$ has an *equivalent key* $sk' = (\mathbf{H}', \cdot')$, if $sk' \neq sk$ and sk' can be used as a secret key with equal efficiency as sk . In particular, \mathbf{H}' is of the same rank as \mathbf{H} .

- ▶ If $\mathbf{W} \in GL_n(\mathbb{F}_q)$, $sk' = (\mathbf{W}\mathbf{H}', \cdot')$ is an equivalent key.
- ▶ Decryption failures are invariant with respect to equivalent keys
- ▶ We can rewrite \mathbf{H} as

$$\mathbf{H} = \sum_{i=1}^d \hat{\mathbf{H}}_i \cdot F_i = \sum_{i=1}^d [\hat{\mathbf{H}}_{i1} | \hat{\mathbf{H}}_{i2}] \cdot F_i$$

$$\Rightarrow \mathbf{H}' = [\mathbf{I}_{n-k} | \hat{\mathbf{H}}'_{12}] \cdot F_1 + \sum_{i=2}^d [\hat{\mathbf{H}}'_{i1} | \hat{\mathbf{H}}'_{i2}] \cdot F_i \text{ is an equivalent key.}$$

- ▶ **We reduce the number of variables to $nd - (n - k)$.**

Our Attack

Our attack on McNie

- ▶ We evaluated the attack on the 1st round submission parameters

n	k	d	r	q	m	Dec. Failure	Security (bits)	Classical Attack (bits)	Quantum Attack (bits)	t
93	62	3	5	2	37	2^{-17}	128	138	82.8	8
105	70	3	5	2	37	2^{-20}	128	140	83.7	8
111	74	3	7	2	41	2^{-17}	192	188	108	8
123	82	3	7	2	41	2^{-20}	192	189	109	8
111	74	3	7	2	59	2^{-17}	256	188	108	8
141	94	3	9	2	47	2^{-20}	256	238	134	8

- ▶ Better attacks exist -
We do not take advantage of any additional structure of McNie
- ▶ We do not take full advantage of the high decryption failure

Conclusion

Final notes

- ▶ Reaction attacks in the Hamming metric - Guo, Johansson, Stankovski '16;
- ▶ A concurrent work in the Rank metric - Aragon, Gaborit '19;

Differences to our attack

- ▶ We need only a handful of observed decryption failures
- ▶ We do not rely on any statistical tests
- ▶ We do not rely on any specific decoder
- ▶ We assume “randomly generated” errors

Questions

Thank you for your attention.

Questions?

gustavo@cryptme.in

