Algebraic techniques for cryptanalysis of rank-based cryptosystems

Simona Samardjiska
(joint work with
Paolo Santini, Edoardo Persichetti and Gustavo Banegas)

Radboud University, Nijmegen, The Netherlands

2019-07-09
AC19

Ongoing NIST competition for Post Quantum Cryptography

- Second round $\mathcal{MQ}$-based candidates
  - LUOV
  - Rainbow
  - GeMSS
  - MQDSS
- Second round Rank based candidates
  - Rollo (merge of LAKE, LOCKER, Ouroboros-R)
  - RQC

Ongoing NIST competition for Post Quantum Cryptography

- Second round $\mathcal{MQ}$-based candidates
  - LUOV
  - Rainbow
  - GeMSS
  - MQDSS
- Second round Rank based candidates
  - Rollo (merge of LAKE, LOCKER, Ouroboros-R)
  - RQC

- Best attacks - guessing + solving systems of equations
- Except for MQDSS and LUOV, all are instances of a MinRank problem
  - Decoding in the rank metric is essentially structured MinRank

Ongoing NIST competition for Post Quantum Cryptography

- Second round $\mathcal{MQ}$-based candidates
  - LUOV
  - Rainbow
  - GeMSS
  - MQDSS
- Second round Rank based candidates
  - Rollo (merge of LAKE, LOCKER, Ouroboros-R)
  - RQC

- Best attacks - guessing + solving systems of equations
- Except for MQDSS and LUOV, all are instances of a MinRank problem
  - Decoding in the rank metric is essentially structured MinRank

- **In this talk:** Using MinRank to mount
  Decryption failure attack for Rank based cryptosystems

MinRank $MR(n, m, r, M_1, \ldots, M_m)$

Input: $n, m, r \in \mathbb{N}$, and $M_1, \ldots, M_m \in \mathcal{M}_n(\mathbb{F}_q)$.
Question: Find – if any – a nonzero m-tuple $(\lambda_1, \ldots, \lambda_m) \in \mathbb{F}_q^m$ s.t.:

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i M_i\right) \leqslant r.$$

[Courtois '01], [Buss & Shallit '99]

MinRank $MR(n, m, r, M_1, \ldots, M_m)$

Input: $n, m, r \in \mathbb{N}$, and $M_1, \ldots, M_m \in \mathcal{M}_n(\mathbb{F}_q)$.
Question: Find – if any – a nonzero m-tuple $(\lambda_1, \ldots, \lambda_m) \in \mathbb{F}_q^m$ s.t.:

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i\, M_i\right) \leqslant r.$$

[Courtois '01], [Buss & Shallit '99]

- **Solving MinRank**
  - Kernel method [Goubin-Courtois'00]
  - Kipnis-Shamir method [Kipnis-Shamir'99]
  - Minors method [Faugère et al.'08]

Solving MinRank - Kernel method

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i \, M_i\right) \leq r \;\Leftrightarrow\; \text{Dim}\left(\text{Ker}\left(\sum_{i=1}^{m} \lambda_i \, M_i\right)\right) \geqslant n - k$$

Solving MinRank - Kernel method

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i M_i\right) \leq r \;\Leftrightarrow\; \text{Dim}\left(\text{Ker}\left(\sum_{i=1}^{m} \lambda_i M_i\right)\right) \geqslant n - k$$

- Guess $\lceil \frac{m}{n} \rceil$ vectors $v_i \in \text{Ker}\left(\sum_{i=1}^{m} \lambda_i M_i\right)$
- Form linear equations in the λ_i variables

$$v_i \cdot \left(\sum_{i=1}^{m} \lambda_i M_i\right) = \mathbf{0}_{1\times n}.$$

Solving MinRank - Kernel method

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i M_i\right) \leq r \;\Leftrightarrow\; \text{Dim}\left(\text{Ker}\left(\sum_{i=1}^{m} \lambda_i M_i\right)\right) \geqslant n - k$$

- Guess $\lceil \frac{m}{n} \rceil$ vectors $v_i \in \text{Ker}\left(\sum_{i=1}^{m} \lambda_i M_i\right)$
- Form linear equations in the λ_i variables

$$v_i \cdot \left(\sum_{i=1}^{m} \lambda_i M_i\right) = \mathbf{0}_{1\times n}.$$

- Complexity: $\mathcal{O}\left(q^{\lceil \frac{m}{n} \rceil \cdot r} m^3\right)$

Solving MinRank - Kipnis-Shamir modeling

$$\mathrm{Rank}\left(\sum_{i=1}^{m} \lambda_i M_i\right) \leq r \Leftrightarrow \exists\, x^{(1)}, \ldots, x^{(n-r)} \in \mathrm{Ker}\left(\sum_{i=1}^{m} \lambda_i M_i\right)$$

$$\begin{pmatrix} 1 & & & x_1^{1} & \ldots & x_r^{(1)} \\ & \ddots & & \vdots & & \vdots \\ & & 1 & x_1^{(n-r)} & \ldots & x_r^{(n-r)} \end{pmatrix} \cdot \left(\sum_{i=1}^{m} \lambda_i M_i\right) = \mathbf{0}_{n\times n}.$$

$n\,(n-r)$ quadratic (bilinear) equations in $r\,(n-r) + m$ variables

Solving MinRank - Kipnis-Shamir modeling

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i M_i\right) \leq r \;\Leftrightarrow\; \exists\, x^{(1)}, \dots, x^{(n-r)} \in \text{Ker}\left(\sum_{i=1}^{m} \lambda_i M_i\right)$$

$$\begin{pmatrix} 1 & & & x_1^{1} & \dots & x_r^{(1)} \\ & \ddots & & \vdots & & \vdots \\ & & 1 & x_1^{(n-r)} & \dots & x_r^{(n-r)} \end{pmatrix} \cdot \left(\sum_{i=1}^{m} \lambda_i M_i\right) = \mathbf{0}_{n\times n}.$$

$n\,(n-r)$ quadratic (bilinear) equations in $r\,(n-r) + m$ variables

- Relinearization [Kipnis & Shamir '99]

Solving MinRank - Kipnis-Shamir modeling

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i\, M_i\right) \leq r \;\Leftrightarrow\; \exists\; x^{(1)}, \ldots, x^{(n-r)} \in \text{Ker}\left(\sum_{i=1}^{m} \lambda_i\, M_i\right)$$

$$\begin{pmatrix} 1 & & & x_1^1 & \ldots & x_r^{(1)} \\ & \ddots & & \vdots & & \vdots \\ & & 1 & x_1^{(n-r)} & \ldots & x_r^{(n-r)} \end{pmatrix} \cdot \left(\sum_{i=1}^{m} \lambda_i\, M_i\right) = \mathbf{0}_{n\times n}.$$

$n\,(n-r)$ quadratic (bilinear) equations in $r\,(n-r)+m$ variables

- Relinearization [Kipnis & Shamir '99]
- Gröbner bases [Faugère & Levy-dit-Vehel & Perret '08]
  - Complexity: $\mathcal{O}\left(\binom{n+d_{\mathrm{reg}}}{d_{\mathrm{reg}}}^{\omega}\right)$ [Faugère '02]

$$d_{reg} \leqslant \min(n_X, n_Y) + 1,$$

for bilinear system in X, Y blocks of variables of sizes n_X, n_Y.

Solving MinRank - Minors modeling

$$\mathrm{Rank}\left(\sum_{i=1}^{m} \lambda_i M_i\right) \leq r \;\Leftrightarrow\; \text{all minors of size } r+1 \text{ of } \left(\sum_{i=1}^{k} \lambda_i M_i\right) \text{ vanish.}$$

$$\binom{n}{r+1}^2 \text{ equations in } m \text{ variables}$$

- [Faugère & Levy-dit-Vehel & Perret '08], [Faugère & Safey El Din & Spaenlehauer '10]

Solving MinRank - Minors modeling

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i M_i\right) \leq r \;\Leftrightarrow\; \text{all minors of size } r+1 \text{ of } \left(\sum_{i=1}^{k} \lambda_i M_i\right) \text{ vanish.}$$

$$\binom{n}{r+1}^2 \text{ equations in } m \text{ variables}$$

- [Faugère & Levy-dit-Vehel & Perret '08],
  [Faugère & Safey El Din & Spaenlehauer '10]
- **Less variables than the Kipnis-Shamir modeling**
  but equations of **degree $r+1$.**
- Complexity: $\mathcal{O}\left(\binom{m}{r+1}^{\omega}\right)$ if fully linearizable [Faugère '02]
- Can be **more efficient** than Kipnis-Shamir method
  (depends on parameters)

Rank metric essentials

- $B = \{B_1, \cdots, B_m\}$ - basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$

$$v \in \mathbb{F}_{q^m} \leftrightarrow v = \sum_{i=1}^{m} \mathcal{F}_i(v) B_i$$

Rank metric essentials

- $B = \{B_1, \cdots, B_m\}$ - basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$

$$v \in \mathbb{F}_{q^m} \leftrightarrow v = \sum_{i=1}^{m} \mathcal{F}_i(v) B_i$$

$$\mathbf{v} \in \mathbb{F}_{q^m}^n = [v_1, \cdots, v_n] \leftrightarrow \bar{\mathbf{V}} = \begin{bmatrix} \mathcal{F}_1(v_1) & \mathcal{F}_1(v_2) & \cdots & \mathcal{F}_1(v_n) \\ \mathcal{F}_2(v_1) & \mathcal{F}_2(v_2) & \cdots & \mathcal{F}_2(v_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{F}_m(v_1) & \mathcal{F}_m(v_2) & \cdots & \mathcal{F}_m(v_n) \end{bmatrix} \in \mathbb{F}_q^{m \times n}.$$

Rank metric essentials

- $B = \{B_1, \cdots, B_m\}$ - basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$

$$v \in \mathbb{F}_{q^m} \leftrightarrow v = \sum_{i=1}^{m} \mathcal{F}_i(v) B_i$$

$$\mathbf{v} \in \mathbb{F}_{q^m}^n = [v_1, \cdots, v_n] \leftrightarrow \bar{\mathbf{V}} = \begin{bmatrix} \mathcal{F}_1(v_1) & \mathcal{F}_1(v_2) & \cdots & \mathcal{F}_1(v_n) \\ \mathcal{F}_2(v_1) & \mathcal{F}_2(v_2) & \cdots & \mathcal{F}_2(v_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{F}_m(v_1) & \mathcal{F}_m(v_2) & \cdots & \mathcal{F}_m(v_n) \end{bmatrix} \in \mathbb{F}_q^{m \times n}.$$

- *Rank distance* between $\mathbf{a}, \mathbf{b} \in \mathbb{F}_{q^m}^n$

$$\mathrm{rd}(\mathbf{a}, \mathbf{b}) = \left| \bar{\mathbf{A}} - \bar{\mathbf{B}} \right|.$$

Rank metric essentials

- $B = \{B_1, \cdots, B_m\}$ - basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$

$$v \in \mathbb{F}_{q^m} \leftrightarrow v = \sum_{i=1}^{m} \mathcal{F}_i(v) B_i$$

$$\mathbf{v} \in \mathbb{F}_{q^m}^n = [v_1, \cdots, v_n] \leftrightarrow \bar{\mathbf{V}} = \begin{bmatrix} \mathcal{F}_1(v_1) & \mathcal{F}_1(v_2) & \cdots & \mathcal{F}_1(v_n) \\ \mathcal{F}_2(v_1) & \mathcal{F}_2(v_2) & \cdots & \mathcal{F}_2(v_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{F}_m(v_1) & \mathcal{F}_m(v_2) & \cdots & \mathcal{F}_m(v_n) \end{bmatrix} \in \mathbb{F}_q^{m \times n}.$$

- *Rank distance* between $\mathbf{a}, \mathbf{b} \in \mathbb{F}_{q^m}^n$

$$\mathrm{rd}(\mathbf{a}, \mathbf{b}) = \left| \bar{\mathbf{A}} - \bar{\mathbf{B}} \right| .$$

- *Support* of $\mathbf{v} \in V_n$ - $\langle \mathbf{v} \rangle$ - the subspace generated by $v_1, \cdots, v_n$

Rank metric essentials

- $B = \{B_1, \cdots, B_m\}$ - basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$

$$v \in \mathbb{F}_{q^m} \leftrightarrow v = \sum_{i=1}^{m} \mathcal{F}_i(v) B_i$$

$$\mathbf{v} \in \mathbb{F}_{q^m}^n = [v_1, \cdots, v_n] \leftrightarrow \bar{\mathbf{V}} = \begin{bmatrix} \mathcal{F}_1(v_1) & \mathcal{F}_1(v_2) & \cdots & \mathcal{F}_1(v_n) \\ \mathcal{F}_2(v_1) & \mathcal{F}_2(v_2) & \cdots & \mathcal{F}_2(v_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{F}_m(v_1) & \mathcal{F}_m(v_2) & \cdots & \mathcal{F}_m(v_n) \end{bmatrix} \in \mathbb{F}_q^{m \times n}.$$

- *Rank distance* between $\mathbf{a}, \mathbf{b} \in \mathbb{F}_{q^m}^n$

$$\mathrm{rd}(\mathbf{a}, \mathbf{b}) = \left| \bar{\mathbf{A}} - \bar{\mathbf{B}} \right| .$$

- *Support* of $\mathbf{v} \in V_n$ - $\langle \mathbf{v} \rangle$ - the subspace generated by $v_1, \cdots, v_n$
- Isometry: If $\mathbf{W} \in GL_n(\mathbb{F}_q)$

$$|\mathbf{b}| = |\mathbf{b} \cdot \mathbf{W}|.$$

LRPC codes

A *Low-Rank Parity-Check (LRPC) code* $\mathcal{C}$ 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 $F = \langle F_1, F_2, \cdots, F_d \rangle$ is a $\mathbb{F}_q$ subspace of $\mathbb{F}_{q^m}$.

Decoding of LRPC codes

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

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

Decoding of LRPC codes

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

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

Crucial facts:

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

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

Decoding of LRPC codes

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

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

Crucial facts:

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

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

- Assume $S = \langle s_1, s_2, \ldots, s_{n-k} \rangle = \langle F_1 E_1, F_1 E_2, \ldots, F_d E_r \rangle$ then:
  1. Set $S_i = F_i^{-1}.S$. Then

$$S_i = F_i^{-1}.\langle \ldots F_i E_1, F_i E_2, \ldots, F_i E_r \ldots \rangle \Rightarrow E = \langle E_1, E_2, \cdots, E_r \rangle \subset S_i$$

  2. Find $E = S_1 \cap S_2 \cap \cdots \cap S_d$
  3. Find $\mathbf{e}$ by solving $\mathbf{He}^\top = \mathbf{s}$

Decoding of LRPC codes

Decoding failures:

1. When $\mathrm{Dim}(\langle EF \rangle) < rd$: 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** $\mathrm{Dim}(S) < rd$ **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

Basically any cryptosystem that

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

$$\text{Find } \mathbf{e} \text{ such that } \mathbf{H}\mathbf{e}^\top = \mathbf{s} \text{ and } |\mathbf{e}| \leqslant r.$$

LRPC cryptosystems

Basically any cryptosystem that

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

$$\text{Find } \mathbf{e} \text{ such that } \mathbf{H}\mathbf{e}^{\top} = \mathbf{s} \text{ and } |\mathbf{e}| \leqslant r.$$

Some examples:

- 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]

Direct attack

Rank syndrome decoding problem:

Given $\mathbf{s}$, find $\mathbf{e}$ such that $\mathbf{He}^\top = \mathbf{s}$ and $|\mathbf{e}| \leqslant r$.

Direct attack

Rank syndrome decoding problem:

Given $\mathbf{s}$, find $\mathbf{e}$ such that $\mathbf{H}\mathbf{e}^\top = \mathbf{s}$ and $|\mathbf{e}| \leqslant r$.

Write the dual (using the generator matrix $\mathbf{G}$):

Given $\mathbf{c}$, find $\mathbf{e}$ (or $\mathbf{m}$) such that $\mathbf{m}\mathbf{G} + \mathbf{e} = \mathbf{c}$ and $|\mathbf{e}| \leqslant r$.

Or as: Given $\mathbf{c}$, find $\mathbf{m}$ such that:

$$|\mathbf{c} - \mathbf{m}\mathbf{G}| \leqslant r$$

Direct attack

Rank syndrome decoding problem:

Given $\mathbf{s}$, find $\mathbf{e}$ such that $\mathbf{He}^\top = \mathbf{s}$ and $|\mathbf{e}| \leqslant r$.

Write the dual (using the generator matrix $\mathbf{G}$):

Given $\mathbf{c}$, find $\mathbf{e}$ (or $\mathbf{m}$) such that $\mathbf{mG} + \mathbf{e} = \mathbf{c}$ and $|\mathbf{e}| \leqslant r$.

Or as: Given $\mathbf{c}$, find $\mathbf{m}$ such that:

$$|\mathbf{c} - \mathbf{mG}| \leqslant r$$

$$|\mathbf{c} - \sum_{i=1}^{k} (\mu_1^i b_1 + \cdots + \mu_m^i b_m)\mathbf{g}_i| \leqslant r$$

Direct attack

Rank syndrome decoding problem:

Given $\mathbf{s}$, find $\mathbf{e}$ such that $\mathbf{H}\mathbf{e}^\top = \mathbf{s}$ and $|\mathbf{e}| \leqslant r$.

Write the dual (using the generator matrix $\mathbf{G}$):

Given $\mathbf{c}$, find $\mathbf{e}$ (or $\mathbf{m}$) such that $\mathbf{mG} + \mathbf{e} = \mathbf{c}$ and $|\mathbf{e}| \leqslant r$.

Or as: Given $\mathbf{c}$, find $\mathbf{m}$ such that:

$$|\mathbf{c} - \mathbf{mG}| \leqslant r$$

$$|\mathbf{c} - \sum_{i=1}^{k} (\mu_1^i b_1 + \cdots + \mu_m^i b_m)\mathbf{g}_i| \leqslant r$$

$$|\mathbf{c} - \sum_{i=1}^{k} \sum_{j=1}^{m} \mu_j^i (b_j \mathbf{g}_i)| \leqslant r$$

Direct attack

Rank syndrome decoding problem:

Given $\mathbf{s}$, find $\mathbf{e}$ such that $\mathbf{H}\mathbf{e}^\top = \mathbf{s}$ and $|\mathbf{e}| \leqslant r$.

Write the dual (using the generator matrix $\mathbf{G}$):

Given $\mathbf{c}$, find $\mathbf{e}$ (or $\mathbf{m}$) such that $\mathbf{m}\mathbf{G} + \mathbf{e} = \mathbf{c}$ and $|\mathbf{e}| \leqslant r$.

Or as: Given $\mathbf{c}$, find $\mathbf{m}$ such that:

$$|\mathbf{c} - \mathbf{m}\mathbf{G}| \leqslant r$$

$$|\mathbf{c} - \sum_{i=1}^{k} (\mu_1^i b_1 + \cdots + \mu_m^i b_m)\mathbf{g}_i| \leqslant r$$

$$|\mathbf{c} - \sum_{i=1}^{k} \sum_{j=1}^{m} \mu_j^i (b_j \mathbf{g}_i)| \leqslant r$$

MinRank $MR(n, mk, r, M_1, \ldots, M_{mk})$.

And a key recovery attack?

And a key recovery attack?

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

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

And a key recovery attack?

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

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

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

And a key recovery attack?

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

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

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

In matrix form:

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

And a key recovery attack?

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

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

$$s_i = \sum_{j=1}^{n} h_{i,j} e_j = \sum_{j=1}^{n} \left(\sum_{l=1}^{d} h_{i,j,l} F_l \right) \left(\sum_{v=1}^{r} e_{j,u} E_u \right)$$

$$= \sum_{l=1}^{d} \sum_{u=1}^{r} F_l E_u \left(\sum_{j=1}^{n} h_{i,j,l} e_{j,u} \right), \quad \forall i \in \{1, \ldots, n-k\}.$$

In matrix form:

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

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

And a key recovery attack?

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

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

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

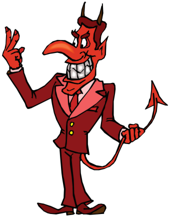
In matrix form:

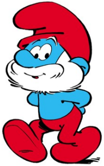
$$\mathbf{s} = (F_1 E_1, F_1 E_2 \ldots, F_d E_r) \cdot \bar{\mathbf{A}}_{\mathbf{h},\mathbf{e}}$$

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

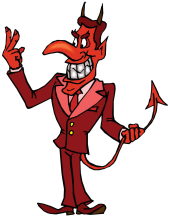
I.e. $\bar{\mathbf{A}}_{\mathbf{h},\mathbf{e}}$ is not of full rank

Reaction attack

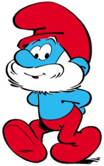




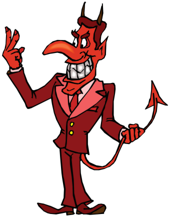
Reaction attack

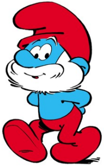


$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1$



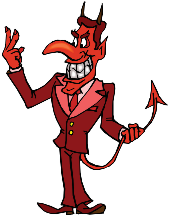
Reaction attack

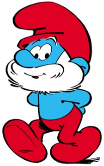




$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1 \xrightarrow{\quad \mathbf{c}_1 \quad}$$

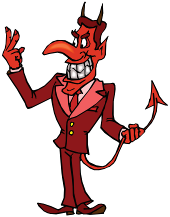
Reaction attack

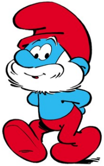




$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1 \xrightarrow{\quad \mathbf{c}_1 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_1)$$

Reaction attack

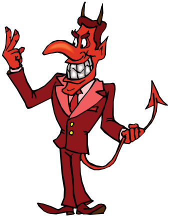


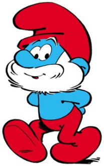


$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1 \xrightarrow{\quad \mathbf{c}_1 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_1)$$

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

Reaction attack

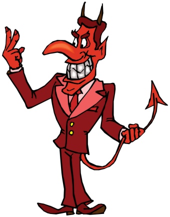


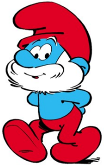


$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1 \xrightarrow{\quad \mathbf{c}_1 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_1)$$

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

Reaction attack

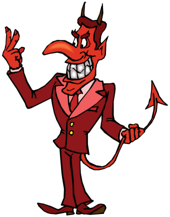


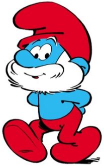


$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1 \xrightarrow{\qquad\mathbf{c}_1\qquad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_1)$$

$$\mathbf{m}_2, \mathbf{e}_2, \mathbf{c}_2 \xrightarrow{\qquad\mathbf{c}_2\qquad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_2)$$

Reaction attack



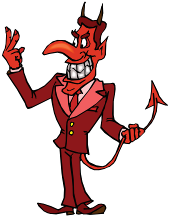


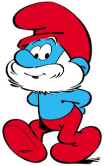
$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1 \xrightarrow{\quad \mathbf{c}_1 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_1)$$

$$\mathbf{m}_2, \mathbf{e}_2, \mathbf{c}_2 \xrightarrow{\quad \mathbf{c}_2 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_2)$$

$$\xrightarrow{\quad \ldots \quad}$$

Reaction attack





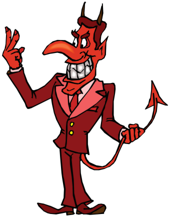
$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1 \xrightarrow{\quad \mathbf{c}_1 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_1)$$

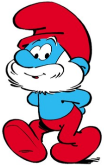
$$\mathbf{m}_2, \mathbf{e}_2, \mathbf{c}_2 \xrightarrow{\quad \mathbf{c}_2 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_2)$$

$$\xrightarrow{\quad \dots \quad}$$

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

Reaction attack





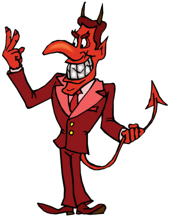
$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1 \xrightarrow{\mathbf{c}_1} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_1)$$

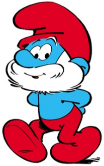
$$\mathbf{m}_2, \mathbf{e}_2, \mathbf{c}_2 \xrightarrow{\mathbf{c}_2} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_2)$$

$$\xrightarrow{\dots}$$

$$\mathbf{m}_t, \mathbf{e}_t, \mathbf{c}_t \xrightarrow{\mathbf{c}_t}$$

Reaction attack





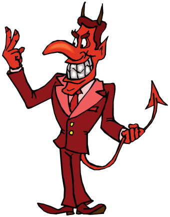
$$\mathbf{m}_1, \mathbf{e}_1, \mathbf{c}_1 = \mathbf{m}_1\mathbf{G} + \mathbf{e}_1 \xrightarrow{\quad \mathbf{c}_1 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_1)$$

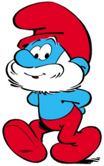
$$\mathbf{m}_2, \mathbf{e}_2, \mathbf{c}_2 \xrightarrow{\quad \mathbf{c}_2 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_2)$$

$$\xrightarrow{\quad \dots \quad}$$

$$\mathbf{m}_t, \mathbf{e}_t, \mathbf{c}_t \xrightarrow{\quad \mathbf{c}_t \quad} \text{X} \leftarrow \mathit{Decode}(\mathbf{c}_t)$$

Reaction attack





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

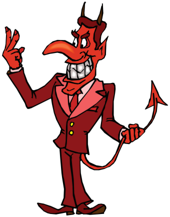
$\mathbf{m}_2, \mathbf{e}_2, \mathbf{c}_2 \xrightarrow{\quad \mathbf{c}_2 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_2)$

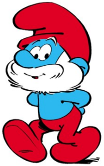
$\xrightarrow{\quad \ldots \quad}$

$\mathbf{m}_t, \mathbf{e}_t, \mathbf{c}_t \xrightarrow{\quad \mathbf{c}_t \quad} \text{X} \leftarrow \mathit{Decode}(\mathbf{c}_t)$

$\xleftarrow{\quad \text{Pls resend!} \quad}$

Reaction attack





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

$\mathbf{m}_2, \mathbf{e}_2, \mathbf{c}_2 \xrightarrow{\quad \mathbf{c}_2 \quad} \checkmark \leftarrow \mathit{Decode}(\mathbf{c}_2)$

$\xrightarrow{\quad \dots \quad}$

$\mathbf{m}_t, \mathbf{e}_t, \mathbf{c}_t \xrightarrow{\quad \mathbf{c}_t \quad} \text{X} \leftarrow \mathit{Decode}(\mathbf{c}_t)$

$\xleftarrow{\quad \text{Pls resend!} \quad}$

Our attack

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

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

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

In matrix form:

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

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

$\bar{\mathbf{A}}_{\mathbf{e}_1}(\mathbf{h})$

$$\mathbf{v}_{\mathbf{e}_1} \cdot \bar{\mathbf{A}}_{\mathbf{e}_1}(\mathbf{h}) = \mathbf{0}_{1 \times n-k}$$

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

$$\mathbf{v}_{\mathbf{e}_1} \cdot \bar{\mathbf{A}}_{\mathbf{e}_1}(\mathbf{h}) = \mathbf{0}_{1\times n-k}$$
$$\mathbf{v}_{\mathbf{e}_2} \cdot \bar{\mathbf{A}}_{\mathbf{e}_2}(\mathbf{h}) = \mathbf{0}_{1\times n-k}$$
$$\cdots$$

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

Form a system and try to solve it

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

Form a system and try to solve it

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

High level attack idea:

Collect errors $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_t$ *from decryption failures*
repeat
    $\mathbf{h} \leftarrow \text{SolveSystem}(\mathbf{v}_{\mathbf{e}_1}, \mathbf{v}_{\mathbf{e}_2}, \dots, \mathbf{v}_{\mathbf{e}_t}, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_t)$
    if $\mathbf{h} \neq \perp$ **then**
        Collect ℓ *messages, errors, ciphertexts* $(\mathbf{m}_i, \mathbf{e}_i, \mathbf{c}_i)$
        $F, \text{success} \leftarrow \text{FindBasis}(\mathbf{h}, \{(\mathbf{m}_i, \mathbf{e}_i, \mathbf{c}_i)\}_{i=1}^{\ell})$
    else success $\leftarrow \perp$
    end if
until success
$\mathbf{H} \leftarrow \text{ReconstructMatrix}(\mathbf{h}, F)$
return H of small rank d

How to solve the system?

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

- Kipnis-Shamir method

How to solve the system?

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

- Kipnis-Shamir method
  - $n-k$ equations for each error $\mathbf{e}_i$
  - typically, nd unknown coefficients in $\mathbf{h}$
  - rd new variables from each $\mathbf{v}_{\mathbf{e}_i}$

How to solve the system?

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

- Kipnis-Shamir method
  - $n-k$ equations for each error $\mathbf{e}_i$
  - typically, nd unknown coefficients in $\mathbf{h}$
  - rd new variables from each $\mathbf{v}_{\mathbf{e}_i}$
  - bilinear system in $trd + nd$ variables

How to solve the system?

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

- Kipnis-Shamir method
  - $n-k$ equations for each error $\mathbf{e}_i$
  - typically, nd unknown coefficients in $\mathbf{h}$
  - rd new variables from each $\mathbf{v}_{\mathbf{e}_i}$
  - bilinear system in $trd + nd$ variables
  - need to collect $t \geqslant \frac{nd}{n-k-rd}$ errors from DF

How to solve the system?

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

- Kipnis-Shamir method
  - $n-k$ equations for each error $\mathbf{e}_i$
  - typically, nd unknown coefficients in $\mathbf{h}$
  - rd new variables from each $\mathbf{v}_{\mathbf{e}_i}$
  - bilinear system in $trd + nd$ variables
  - need to collect $t \geqslant \frac{nd}{n-k-rd}$ errors from DF
  - $\Rightarrow$ Problem - too many variables

How to solve the system?

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

- Kipnis-Shamir method
  - $n-k$ equations for each error $\mathbf{e}_i$
  - typically, nd unknown coefficients in $\mathbf{h}$
  - rd new variables from each $\mathbf{v}_{\mathbf{e}_i}$
  - bilinear system in $trd + nd$ variables
  - need to collect $t \geqslant \frac{nd}{n-k-rd}$ errors from DF
  - $\Rightarrow$ Problem - too many variables

- Minors method

How to solve the system?

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

- Kipnis-Shamir method
  - $n-k$ equations for each error $\mathbf{e}_i$
  - typically, nd unknown coefficients in $\mathbf{h}$
  - rd new variables from each $\mathbf{v}_{\mathbf{e}_i}$
  - bilinear system in $trd + nd$ variables
  - need to collect $t \geqslant \frac{nd}{n-k-rd}$ errors from DF
  - $\Rightarrow$ Problem - too many variables
- Minors method
  - $\binom{n-k}{rd}$ equations for each error $\mathbf{e}_i$ in degree rd
  - only the nd unknown coefficients from $\mathbf{h}$

How to solve the system?

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

- Kipnis-Shamir method
  - $n-k$ equations for each error $\mathbf{e}_i$
  - typically, nd unknown coefficients in $\mathbf{h}$
  - rd new variables from each $\mathbf{v}_{\mathbf{e}_i}$
  - bilinear system in $trd + nd$ variables
  - need to collect $t \geqslant \frac{nd}{n-k-rd}$ errors from DF
  - $\Rightarrow$ Problem - too many variables
- Minors method
  - $\binom{n-k}{rd}$ equations for each error $\mathbf{e}_i$ in degree rd
  - only the nd unknown coefficients from $\mathbf{h}$
  - need to collect $t \geqslant \frac{\binom{nd}{rd}}{\binom{n-k}{rd}}$ errors from DF for full linearization

How to solve the system?

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

- Kipnis-Shamir method
  - $n-k$ equations for each error $\mathbf{e}_i$
  - typically, nd unknown coefficients in $\mathbf{h}$
  - rd new variables from each $\mathbf{v}_{\mathbf{e}_i}$
  - bilinear system in $trd + nd$ variables
  - need to collect $t \geqslant \frac{nd}{n-k-rd}$ errors from DF
  - $\Rightarrow$ Problem - too many variables
- Minors method
  - $\binom{n-k}{rd}$ equations for each error $\mathbf{e}_i$ in degree rd
  - only the nd unknown coefficients from $\mathbf{h}$
  - need to collect $t \geqslant \frac{\binom{nd}{rd}}{\binom{n-k}{rd}}$ errors from DF for full linearization
  - $\Rightarrow$ Problem - too big final system + t might be too big

How to solve the system?

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

- Kernel method

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 $\bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h})$

How to solve the system?

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

- 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 $\bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h})$
  - $\Rightarrow$ linear system only in the nd $\mathbf{h}$-variables

How to solve the system?

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

- 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 $\bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h})$
  - $\Rightarrow$ linear system only in the nd $\mathbf{h}$-variables
  - need to collect $t \geqslant \frac{nd}{n-k}$ errors from DF

How to solve the system?

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

- 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 $\bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h})$
  - $\Rightarrow$ linear system only in the nd $\mathbf{h}$-variables
  - need to collect $t \geqslant \frac{nd}{n-k}$ errors from DF
  - Probability of guessing $\mathbf{v}_{\mathbf{e}_i}$ correctly: $P_{\mathbf{e}_i} = \dfrac{q^{K_{\mathbf{e}_i}}}{q^{rd}}$.

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 $\bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h})$
  - $\Rightarrow$ linear system only in the nd $\mathbf{h}$-variables
  - need to collect $t \geqslant \frac{nd}{n-k}$ errors from DF
  - Probability of guessing $\mathbf{v}_{\mathbf{e}_i}$ correctly: $P_{\mathbf{e}_i} = \dfrac{q^{K_{\mathbf{e}_i}}}{q^{rd}}$.
  - No way of detecting a larger kernel - effectively $dim = 1$

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 $\bar{\mathbf{A}}_{\mathbf{e}_t}(\mathbf{h})$
  - $\Rightarrow$ linear system only in the nd $\mathbf{h}$-variables
  - need to collect $t \geqslant \frac{nd}{n-k}$ errors from DF
  - Probability of guessing $\mathbf{v}_{\mathbf{e}_i}$ correctly: $P_{\mathbf{e}_i} = \dfrac{q^{K_{\mathbf{e}_i}}}{q^{rd}}$.
  - No way of detecting a larger kernel - effectively $dim = 1$
  - Probability of guessing all $\mathbf{v}_{\mathbf{e}_1}, \dots, \mathbf{v}_{\mathbf{e}_t}$
    $$P_t = P_{\mathbf{e}_i}^t = q^{-(rd-1)t}$$

Equivalent keys

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

Equivalent keys

An LRPC cryptosystem, with a secret key $\mathrm{sk} = (\mathbf{H}, \cdot)$ has an *equivalent key* $\mathrm{sk}' = (\mathbf{H}', \cdot')$, if $\mathrm{sk}' \neq \mathrm{sk}$ 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 $\mathbf{W} \in GL_n(\mathbb{F}_q)$, $\mathrm{sk}' = (\mathbf{W}\mathbf{H}', \cdot')$ is an equivalent key

Equivalent keys

An LRPC cryptosystem, with a secret key $\mathrm{sk} = (\mathbf{H}, \cdot)$ has an *equivalent key* $\mathrm{sk}' = (\mathbf{H}', \cdot')$, if $\mathrm{sk}' \neq \mathrm{sk}$ 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 $\mathbf{W} \in GL_n(\mathbb{F}_q)$, $\mathrm{sk}' = (\mathbf{WH}', \cdot')$ is an equivalent key
- Decryption failures are invariant with respect to equivalent keys

Equivalent keys

An LRPC cryptosystem, with a secret key $\mathrm{sk} = (\mathbf{H}, \cdot)$ has an *equivalent key* $\mathrm{sk}' = (\mathbf{H}', \cdot')$, if $\mathrm{sk}' \neq \mathrm{sk}$ 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 $\mathbf{W} \in GL_n(\mathbb{F}_q)$, $\mathrm{sk}' = (\mathbf{W}\mathbf{H}', \cdot')$ is an equivalent key
- Decryption failures are invariant with respect to equivalent keys
- We can rewrite $\mathbf{H}$ as

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

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

Equivalent keys

An LRPC cryptosystem, with a secret key $\mathrm{sk} = (\mathbf{H}, \cdot)$ has an *equivalent key* $\mathrm{sk}' = (\mathbf{H}', \cdot')$, if $\mathrm{sk}' \neq \mathrm{sk}$ 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 $\mathbf{W} \in GL_n(\mathbb{F}_q)$, $\mathrm{sk}' = (\mathbf{WH}', \cdot')$ 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 $nd - (n-k)$.**
- **Typically, we reduce t by 1**

Case study - McNie

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

n	k	d	r	q	m	Dec. Failure	Security (bits)	Attack (Classical)	Attack (Quantum)	t
93	62	3	5	2	37	2^{-17}	128	128.8	82.8	8
105	70	3	5	2	37	2^{-20}	128	139.7	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

Case study - McNie

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

n	k	d	r	q	m	Dec. Failure	Security (bits)	Attack (Classical)	Attack (Quantum)	t
93	62	3	5	2	37	2^{-17}	128	128.8	82.8	8
105	70	3	5	2	37	2^{-20}	128	139.7	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

Final words

- Reaction attacks in the Hamming metric - Guo, Johansson, Stankovski '16

Final words

- Reaction attacks in the Hamming metric - Guo, Johansson, Stankovski '16
- A concurent work in the Rank metric - Aragon, Gaborit '19
  - similar in nature to Guo et al.'s, but not efficient

Final words

- Reaction attacks in the Hamming metric - Guo, Johansson, Stankovski '16
- A concurent work in the Rank metric - Aragon, Gaborit '19
  - similar in nature to Guo et al.'s, but not efficient

Differences in our attack

- We need only a handful of observed decryption failures
  - Is there a trade-off?

Final words

- Reaction attacks in the Hamming metric - Guo, Johansson, Stankovski '16
- A concurent work in the Rank metric - Aragon, Gaborit '19
  - similar in nature to Guo et al.'s, but not efficient

Differences in our attack

- We need only a handful of observed decryption failures
  - Is there a trade-off?
- We don't rely on any statistical tests

Final words

- Reaction attacks in the Hamming metric - Guo, Johansson, Stankovski '16
- A concurent work in the Rank metric - Aragon, Gaborit '19
  - similar in nature to Guo et al.'s, but not efficient

Differences in our attack

- We need only a handful of observed decryption failures
  - Is there a trade-off?
- We don't rely on any statistical tests
- We don't rely on any specific decoder

Final words

- Reaction attacks in the Hamming metric - Guo, Johansson, Stankovski '16
- A concurent work in the Rank metric - Aragon, Gaborit '19
  - similar in nature to Guo et al.'s, but not efficient

Differences in our attack

- We need only a handful of observed decryption failures
  - Is there a trade-off?
- We don't rely on any statistical tests
- We don't rely on any specific decoder
- The attack works even in a CCA setting
  - We assume "randomly generated" errors

Final words

- Reaction attacks in the Hamming metric - Guo, Johansson, Stankovski '16
- A concurent work in the Rank metric - Aragon, Gaborit '19
  - similar in nature to Guo et al.'s, but not efficient

Differences in our attack

- We need only a handful of observed decryption failures
  - Is there a trade-off?
- We don't rely on any statistical tests
- We don't rely on any specific decoder
- The attack works even in a CCA setting
  - We assume "randomly generated" errors
- Plenty of room for improvement!

Thank you for listening!