Radbound University

Multivariate cryptography – Cryptanalysis techniques

SLMath summer school:
Introduction to Quantum-Safe Cryptography (IBM Zurich)

Simona Samardjiska

July, 2024

Institute for Computing and Information Sciences
Radboud University

The MinRank problem

The MinRank problem

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]

- **NP-hard**, however...
- instances in MQ crypto can be much easier
- polynomial complexity when $n - r$ is constant
- **Solving MinRank**
  - Kernel method [Goubin-Courtois'00]
  - Kipnis-Shamir method [Kipnis-Shamir'99]
  - Minors method [Faugère et al.'08]

The MinRank problem

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]

- **NP-hard**, however...
- instances in MQ crypto can be much easier
- polynomial complexity when $n - r$ is constant
- **Solving MinRank**
  - Kernel method [Goubin-Courtois'00]
  - Kipnis-Shamir method [Kipnis-Shamir'99]
  - Minors method [Faugère et al.'08]

The MinRank problem

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.:

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

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

- **NP-hard**, however...
- instances in MQ crypto can be much easier
- polynomial complexity when $n - r$ is constant
- **Solving MinRank**
  - Kernel method [Goubin-Courtois'00]
  - Kipnis-Shamir method [Kipnis-Shamir'99]
  - Minors method [Faugère et al.'08]

The MinRank problem

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]

- **NP-hard**, however...
- instances in MQ crypto can be much easier
- polynomial complexity when $n - r$ is constant
- Solving MinRank
  - Kernel method [Goubin-Courtois'00]
  - Kipnis-Shamir method [Kipnis-Shamir'99]
  - Minors method [Faugère et al.'08]

The MinRank problem

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.:

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

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

- **NP-hard**, however...
- instances in MQ crypto can be much easier
- polynomial complexity when $n - r$ is constant
- **Solving MinRank**
  - Kernel method [Goubin-Courtois'00]
  - Kipnis-Shamir method [Kipnis-Shamir'99]
  - Minors method [Faugère et al.'08]

Solving MinRank - Kernel method

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

- Guess a vector $v \in \text{Ker}\left(\sum_{i=1}^{m} \lambda_i M_i\right)$
- Form n linear equations in the λ_i variables

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

- It is enough to guess $\lceil \frac{m}{n} \rceil$ vectors
- Complexity: $\mathcal{O}\left(q^{\lceil \frac{m}{n} \rceil \cdot r} m^3\right)$

Solving MinRank - Kernel method

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

- Guess a vector $v \in \text{Ker}\left(\sum_{i=1}^{m} \lambda_i M_i\right)$
- Form n linear equations in the λ_i variables

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

- It is enough to guess $\lceil \frac{m}{n} \rceil$ vectors
- Complexity: $\mathcal{O}\left(q^{\lceil \frac{m}{n} \rceil \cdot r} m^3\right)$

Solving MinRank - Kernel method

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

- Guess a vector $v \in \text{Ker}\left(\sum_{i=1}^{m} \lambda_i M_i\right)$
- Form n linear equations in the λ_i variables

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

- It is enough to guess $\lceil \frac{m}{n} \rceil$ vectors
- Complexity: $\mathcal{O}\left(q^{\lceil \frac{m}{n} \rceil \cdot r} m^3\right)$

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

Solving MinRank - Kipnis-Shamir modeling

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

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

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

- Relinearization [Kipnis & Shamir '99]
- Gröbner bases [Faugère & Levy-dit-Vehel & Perret '08]

Solving MinRank - Kipnis-Shamir modeling

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

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

$n\,(n-r)$ 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.

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

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

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

Solving MinRank - Minors modeling

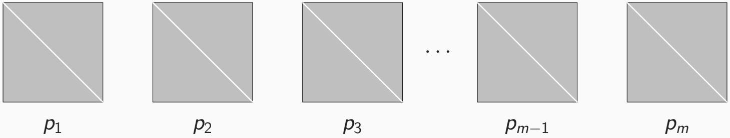
$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i M_i\right) \leq r \iff \text{all minors of size } r+1 \text{ of } \left(\sum_{i=1}^{k} \lambda_i M_i\right) \text{ vanish.}$$

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

- [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)

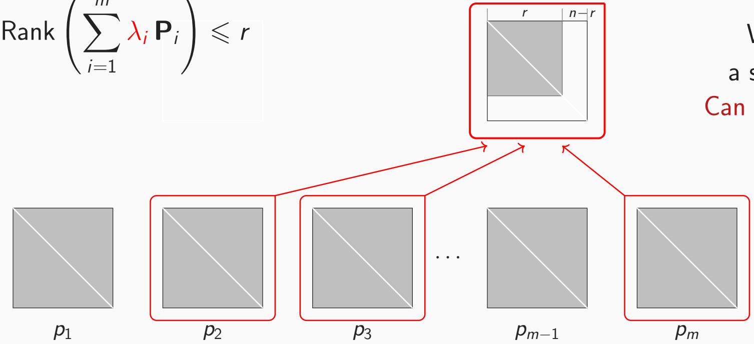
How do we use MinRank in cryptanalysis?

$\mathcal{P} = (p_1, p_2, \ldots, p_m)$ - public polynomials,
$\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m$ - matrix representations of the coordinates of $\mathcal{P}$.



$\mathcal{P} = (p_1, p_2, \ldots, p_m)$ - public polynomials,
$\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m$ - matrix representations of the coordinates of $\mathcal{P}$.

$$\mathrm{Rank}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right) \leqslant r$$



We know there exists
a secret matrix of rank r
Can be found using MinRank

Application of MinRank - Cryptanalysis of HFE

HFE central map [Patarin '96]:

$$\mathcal{F}(X) = \sum_{\substack{0 \leqslant i \leqslant j < n \\ q^i + q^j \leqslant D}} A_{i,j} X^{q^i + q^j} + \sum_{\substack{0 \leqslant i < n \\ q^i \leqslant D}} B_i X^{q^i} + C \in \mathbb{F}_{q^n}[X] \text{ where } D \in \mathbb{N}\,.$$

Application of MinRank - Cryptanalysis of HFE

HFE central map [Patarin '96]:

$$\mathcal{F}(X) = \sum_{\substack{0\leqslant i\leqslant j<n \\ q^i+q^j\leqslant D}} A_{i,j} X^{q^i+q^j} + \sum_{\substack{0\leqslant i<n \\ q^i\leqslant D}} B_i X^{q^i} + C \in \mathbb{F}_{q^n}[X] \text{ where } D \in \mathbb{N}\,.$$

Kipnis-Shamir '99:

$$\underline{X}\mathfrak{F}\underline{X}^\top, \text{ with } \underline{X} = (X, X^q, \ldots, X^{q^{n-1}})$$

[non-standard Matrix Representation]

$$\mathfrak{F} = \begin{pmatrix} A_{1,1} & \ldots & A_{1,\ell} & 0 & \ldots & 0 \\ \vdots & \ldots & \vdots & \vdots & \ldots & \vdots \\ A_{\ell,1} & \ldots & A_{\ell,\ell} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ldots & \vdots & \vdots & \ldots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & 0 \end{pmatrix}$$

Application of MinRank - Cryptanalysis of HFE

HFE central map [Patarin '96]:

$$\mathcal{F}(X) = \sum_{\substack{0 \leqslant i \leqslant j < n \\ q^i + q^j \leqslant D}} A_{i,j} X^{q^i + q^j} + \sum_{\substack{0 \leqslant i < n \\ q^i \leqslant D}} B_i X^{q^i} + C \in \mathbb{F}_{q^n}[X] \text{ where } D \in \mathbb{N}\,.$$

Kipnis-Shamir '99:

$$\underline{X} \mathfrak{F} \underline{X}^\top, \text{ with } \underline{X} = (X, X^q, \ldots, X^{q^{n-1}})$$

[non-standard Matrix Representation]

$$\mathfrak{F} = \begin{pmatrix} A_{1,1} & \ldots & A_{1,\ell} & 0 & \ldots & 0 \\ \vdots & \ldots & \vdots & \vdots & \ldots & \vdots \\ A_{\ell,1} & \ldots & A_{\ell,\ell} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ldots & \vdots & \vdots & \ldots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & 0 \end{pmatrix}$$

$q^i + q^j \leq D$

$\mathrm{rank}(\mathfrak{F}) = \log_q(\deg(\mathcal{F}(X))) = \log_q(D)$.

Application of MinRank - Cryptanalysis of HFE

HFE central map [Patarin '96]:

$$\mathcal{F}(X) = \sum_{\substack{0 \leqslant i \leqslant j < n \\ q^i + q^j \leqslant D}} A_{i,j} X^{q^i + q^j} + \sum_{\substack{0 \leqslant i < n \\ q^i \leqslant D}} B_i X^{q^i} + C \in \mathbb{F}_{q^n}[X] \text{ where } D \in \mathbb{N}\,.$$

Improved: Bettale, Faugère, Perret '11

Change basis between $\mathbb{F}_q^n$ and $\mathbb{F}_{q^n}$: $(x_1, x_2, \ldots, x_n)\mathbf{B} = (X, X^q, \ldots, X^{q^{n-1}})$

$$\sum_{i=1}^{n} \lambda_i \mathbf{P}_i = \mathbf{S} \cdot \mathbf{B} \cdot \mathfrak{F} \cdot \mathbf{B}^\top \cdot \mathbf{S}^\top \quad \Rightarrow \quad \textbf{Rank of } \mathfrak{F} \textbf{ preserved!}$$

Hence, we can use directly the public matrices over $\mathbb{F}_q$:

$$\text{Find } (\lambda_1, \ldots, \lambda_n) \in (\mathbb{F}_{q^n})^n \text{ s.t. } \operatorname{rank}\left(\sum_{i=1}^n \lambda_i \mathbf{P}_i\right) = \log_q(D).$$

Application of MinRank - Cryptanalysis of HFEv-

Recall HFEv- (and GeMSS) central map:

$$\widetilde{\mathcal{F}}(X,x) = \sum_{\substack{0 \le i,j \le D \\ q^i + q^j \le D}} a_{ij} X^{q^i+q^j} + \sum_{\substack{0 \le k \le D \\ q^k \le D}} b_k(x_{n+1}, x_{n+2}, \ldots, x_{n+v}) X^{q^k} + c(x_{n+1}, x_{n+2}, \ldots, x_{n+v})$$

Attack of: Tao, Petzoldt, Ding '21

Use Matrix representation: $\underline{X}\mathfrak{F}\underline{X}^\top$, with $\underline{X} = (X, X^q, \ldots, X^{q^{n-1}}, x_{n+1}, x_{n+2}, \ldots, x_{n+v})$

Change basis between $\mathbb{F}_q^{n+v}$ and $\mathbb{F}_q^n \times \mathbb{F}_{q^v}$: $(x_1, x_2, \ldots, x_{n+v})\tilde{\mathbf{B}} = (X, X^q, \ldots, X^{q^{n-1}}, x_{n+1}, x_{n+2}, \ldots, x_{n+v})$

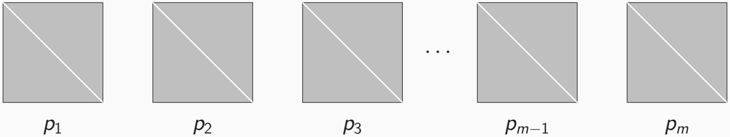
$$\sum_{i=1}^{n} \lambda_i \mathbf{P}_i = \mathbf{S} \cdot \tilde{\mathbf{B}} \cdot \mathfrak{F} \cdot \tilde{\mathbf{B}}^\top \cdot \mathbf{S}^\top \quad \Rightarrow \quad \textbf{Rank of } \mathfrak{F} \textbf{ preserved!}$$

Hence, we can use directly the public matrices over $\mathbb{F}_q$:

$$\text{Find } (\lambda_1, \ldots, \lambda_{n-a}) \in (\mathbb{F}_{q^n})^{n-a} \text{ s.t. } \operatorname{rank}\left(\sum_{i=1}^n \lambda_i \mathbf{P}_i\right) = \log_q(D).$$

How do we use MinRank in cryptanalysis?

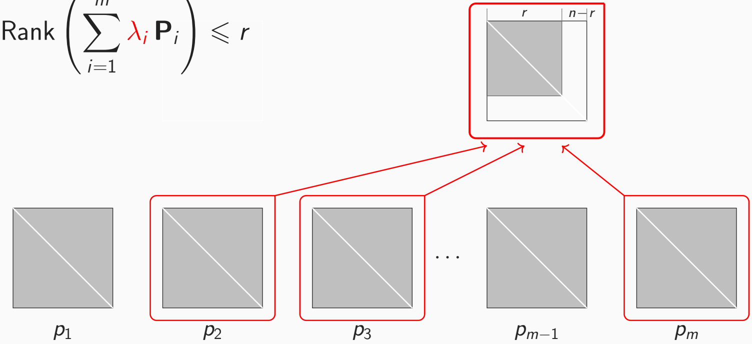
$\mathcal{P} = (p_1, p_2, \ldots, p_m)$ - public polynomials,
$\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m$ - matrix representations of the coordinates of $\mathcal{P}$.



How do we use MinRank in cryptanalysis?

$\mathcal{P} = (p_1, p_2, \ldots, p_m)$ - public polynomials,
$\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m$ - matrix representations of the coordinates of $\mathcal{P}$.

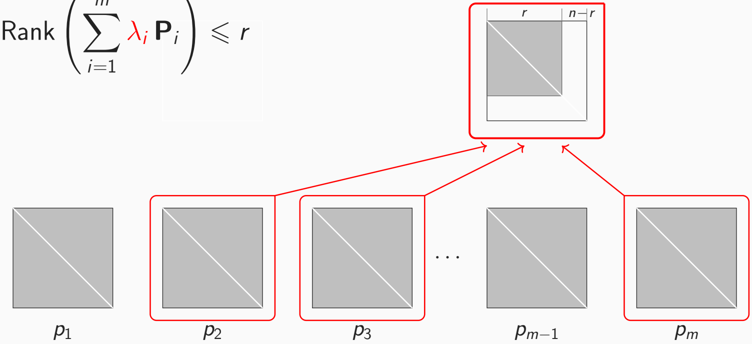
$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right) \leqslant r$$



How do we use MinRank in cryptanalysis?

$\mathcal{P} = (p_1, p_2, \ldots, p_m)$ - public polynomials,
$\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m$ - matrix representations of the coordinates of $\mathcal{P}$.

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right) \leqslant r$$



$\mathcal{S}$ is determined by

$$\text{Ker}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right)$$

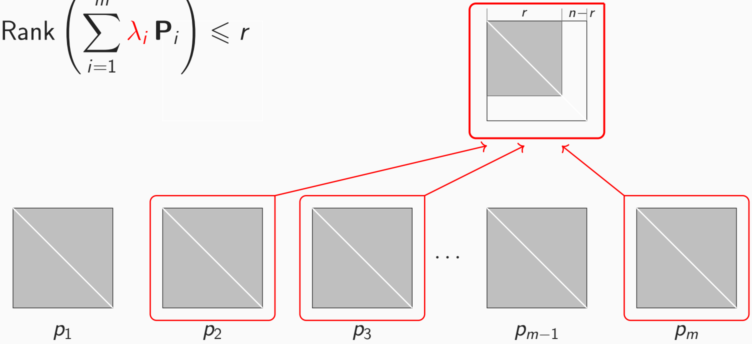
$\mathcal{T}$ is determined by

$$\langle(\lambda_1, \ldots, \lambda_m)\rangle$$

How do we use MinRank in cryptanalysis?

$\mathcal{P} = (p_1, p_2, \ldots, p_m)$ - public polynomials,
$\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m$ - matrix representations of the coordinates of $\mathcal{P}$.

$$\text{Rank}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right) \leqslant r$$



$\mathcal{S}$ is determined by

$$\text{Ker}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right)$$

$\mathcal{T}$ is determined by

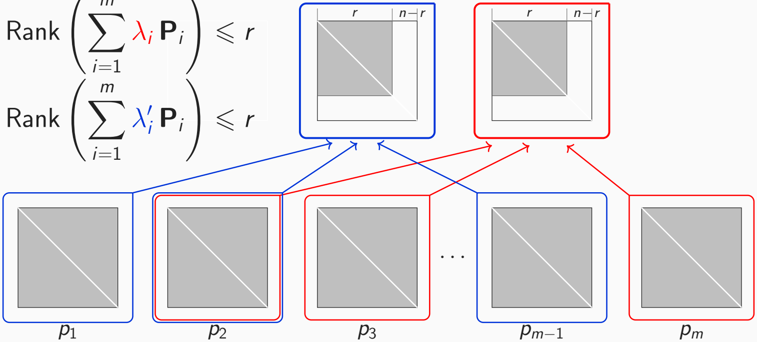
$$\langle(\lambda_1, \ldots, \lambda_m)\rangle$$

How do we use MinRank in cryptanalysis?

$\mathcal{P} = (p_1, p_2, \ldots, p_m)$ - public polynomials,
$\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m$ - matrix representations of the coordinates of $\mathcal{P}$.

$$\mathrm{Rank}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right) \leqslant r$$

$$\mathrm{Rank}\left(\sum_{i=1}^{m} \lambda'_i \mathbf{P}_i\right) \leqslant r$$



$\mathcal{S}$ is determined by

$$\mathrm{Ker}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right)$$

$\mathcal{T}$ is determined by

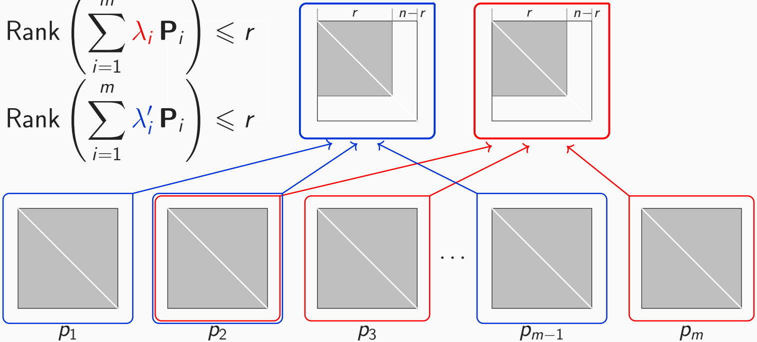
$$\langle(\lambda_1, \ldots, \lambda_m)\rangle$$

How do we use MinRank in cryptanalysis?

$\mathcal{P} = (p_1, p_2, \ldots, p_m)$ - public polynomials,
$\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m$ - matrix representations of the coordinates of $\mathcal{P}$.

$$\mathrm{Rank}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right) \leqslant r$$

$$\mathrm{Rank}\left(\sum_{i=1}^{m} \lambda'_i \mathbf{P}_i\right) \leqslant r$$



$\mathcal{S}$ is determined by

$$\mathrm{Ker}\left(\sum_{i=1}^{m} \lambda_i \mathbf{P}_i\right)$$

$$\cap\, \mathrm{Ker}\left(\sum_{i=1}^{m} \lambda'_i \mathbf{P}_i\right)$$

$\mathcal{T}$ is determined by

$$\langle(\lambda_1, \ldots, \lambda_m),$$

$$(\lambda_1, \ldots, \lambda_m)\rangle$$

Baby example

$$p_1(x_1,\dots,x_6) = x_1x_3 + x_3x_5 + x_4x_5 + x_5 + x_4x_6 + x_6$$

$$p_2(x_1,\dots,x_6) = x_1x_2 + x_1x_3 + x_1x_5 + x_1x_6 + x_2x_6 + x_3x_4 + x_3x_5 + x_3x_6 + x_4x_6 + x_6$$

$$p_3(x_1,\dots,x_6) = x_1x_2 + x_2x_3 + x_1x_4 + x_3x_5 + x_4x_6 + x_5x_6$$

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_3 = \begin{bmatrix} 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 \end{bmatrix}$$

Baby example

$$p_1(x_1,\ldots,x_6) = x_1x_3 + x_3x_5 + x_4x_5 + x_5 + x_4x_6 + x_6$$

$$p_2(x_1,\ldots,x_6) = x_1x_2 + x_1x_3 + x_1x_5 + x_1x_6 + x_2x_6 + x_3x_4 + x_3x_5 + x_3x_6 + x_4x_6 + x_6$$

$$p_3(x_1,\ldots,x_6) = x_1x_2 + x_2x_3 + x_1x_4 + x_3x_5 + x_4x_6 + x_5x_6$$

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_3 = \begin{bmatrix} 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 \end{bmatrix}$$

$$\mathrm{Rank}(\mathbf{P}_3) = 4$$

$$\mathrm{Rank}(\mathbf{P}_2 + \mathbf{P}_3) = 4$$

Baby example

$$p_1(x_1,\dots,x_6) = x_1x_3 + x_3x_5 + x_4x_5 + x_5 + x_4x_6 + x_6$$

$$p_2(x_1,\dots,x_6) = x_1x_2 + x_1x_3 + x_1x_5 + x_1x_6 + x_2x_6 + x_3x_4 + x_3x_5 + x_3x_6 + x_4x_6 + x_6$$

$$p_3(x_1,\dots,x_6) = x_1x_2 + x_2x_3 + x_1x_4 + x_3x_5 + x_4x_6 + x_5x_6$$

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_3 = \begin{bmatrix} 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 \end{bmatrix}$$

$$\mathrm{Rank}(\mathbf{P}_3) = 4$$

$$\mathrm{Rank}(\mathbf{P}_2 + \mathbf{P}_3) = 4$$

$$\mathrm{Ker}(\mathbf{P}_3) \cap \mathrm{Ker}(\mathbf{P}_2 + \mathbf{P}_3) = \langle (0,1,0,1,1,0), (1,1,1,1,1,1) \rangle$$

Baby example

$$p_1(x_1,\dots,x_6) = x_1x_3 + x_3x_5 + x_4x_5 + x_5 + x_4x_6 + x_6$$

$$p_2(x_1,\dots,x_6) = x_1x_2 + x_1x_3 + x_1x_5 + x_1x_6 + x_2x_6 + x_3x_4 + x_3x_5 + x_3x_6 + x_4x_6 + x_6$$

$$p_3(x_1,\dots,x_6) = x_1x_2 + x_2x_3 + x_1x_4 + x_3x_5 + x_4x_6 + x_5x_6$$

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_3 = \begin{bmatrix} 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 \end{bmatrix}$$

$$\mathrm{Rank}(\mathbf{P}_3) = 4$$

$$\mathrm{Rank}(\mathbf{P}_2 + \mathbf{P}_3) = 4$$

$$\mathrm{Ker}(\mathbf{P}_3) \cap \mathrm{Ker}(\mathbf{P}_2 + \mathbf{P}_3) = \boxed{\langle (0,1,0,1,1,0), (1,1,1,1,1,1) \rangle}$$

$$p_1(x_1,\dots,x_6) = x_1x_3 + x_3x_5 + x_4x_5 + x_5 + x_4x_6 + x_6$$

$$p_2(x_1,\dots,x_6) = x_1x_2 + x_1x_3 + x_1x_5 + x_1x_6 + x_2x_6 + x_3x_4 + x_3x_5 + x_3x_6 + x_4x_6 + x_6$$

$$p_3(x_1,\dots,x_6) = x_1x_2 + x_2x_3 + x_1x_4 + x_3x_5 + x_4x_6 + x_5x_6$$

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_3 = \begin{bmatrix} 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 \end{bmatrix}$$

$\mathrm{Rank}(\mathbf{P}_3) = 4$

$\mathrm{Rank}(\mathbf{P}_2 + \mathbf{P}_3) = 4$

$$\mathrm{Ker}\boxed{(\mathbf{P}_3)} \cap \mathrm{Ker}\boxed{(\mathbf{P}_2 + \mathbf{P}_3)} = \boxed{\langle (0,1,0,1,1,0), (1,1,1,1,1,1) \rangle}$$

Two MinRank problems with common kernel

Baby example

$$p_1(x_1,\ldots,x_6) = x_1x_3 + x_3x_5 + x_4x_5 + x_5 + x_4x_6 + x_6$$

$$p_2(x_1,\ldots,x_6) = x_1x_2 + x_1x_3 + x_1x_5 + x_1x_6 + x_2x_6 + x_3x_4 + x_3x_5 + x_3x_6 + x_4x_6 + x_6$$

$$p_3(x_1,\ldots,x_6) = x_1x_2 + x_2x_3 + x_1x_4 + x_3x_5 + x_4x_6 + x_5x_6$$

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_3 = \begin{bmatrix} 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 \end{bmatrix}$$

$\mathrm{Rank}(\mathbf{P}_3) = 4$

$\mathrm{Rank}(\mathbf{P}_2 + \mathbf{P}_3) = 4$

$$\mathrm{Ker}(\boxed{\mathbf{P}_3}) \cap \mathrm{Ker}\boxed{(\mathbf{P}_2 + \mathbf{P}_3)} = \boxed{\langle (0,1,0,1,1,0), (1,1,1,1,1,1) \rangle}$$

After change of variables:

$$p_1(x_1,\ldots,x_6) = x_1x_4 + x_1x_6 + x_2x_3 + x_3x_4 + x_3x_5 + x_4x_5$$

$$p_2(x_1,\ldots,x_6) = x_1x_2 + x_1x_4 + x_2x_3$$

$$p_3(x_1,\ldots,x_6) = x_1x_3 + x_1x_4 + x_2x_3 + x_3x_4$$

Example - Good keys for the MQQ cryptosystems

- **MQQ (Multivariate Quadratic Quasigroups)** [GMK08]
  - The private $\mathcal{F}$ - quasigroup string transformations of MQQs
  - direct algebraic attack
- **MQQ-SIG** [GOJPFKM11]
  - signature scheme
  - **fastest on (eBACS) SUPERCOP**
  - $n/2$ equations removed - measure against the attack
- **MQQ-ENC** [GS12]
  - Atempt on an encryption scheme
  - light use of minus modifier
- All broken [FGPST'15] due to rank defects and nice good keys

Example - Good keys for the MQQ cryptosystems

- **MQQ (Multivariate Quadratic Quasigroups)** [GMK08]
  - The private $\mathcal{F}$ - quasigroup string transformations of MQQs
  - direct algebraic attack
- **MQQ-SIG** [GOJPFKM11]
  - signature scheme
  - **fastest on (eBACS) SUPERCOP**
  - $n/2$ equations removed - measure against the attack
- **MQQ-ENC** [GS12]
  - Atempt on an encryption scheme
  - light use of minus modifier
- All broken [FGPST'15] due to rank defects and nice good keys

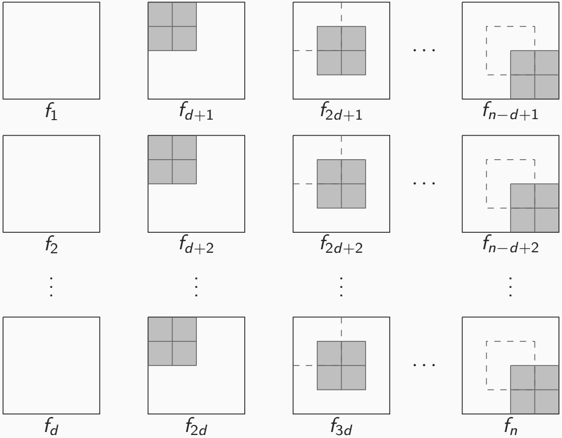
Example - Good keys for the MQQ cryptosystems

- **MQQ (Multivariate Quadratic Quasigroups)** [GMK08]
  - The private $\mathcal{F}$ - quasigroup string transformations of MQQs
  - direct algebraic attack
- **MQQ-SIG** [GOJPFKM11]
  - signature scheme
  - **fastest on (eBACS) SUPERCOP**
  - $n/2$ equations removed - measure against the attack
- **MQQ-ENC** [GS12]
  - Atempt on an encryption scheme
  - light use of minus modifier
- All broken [FGPST'15] due to rank defects and nice good keys

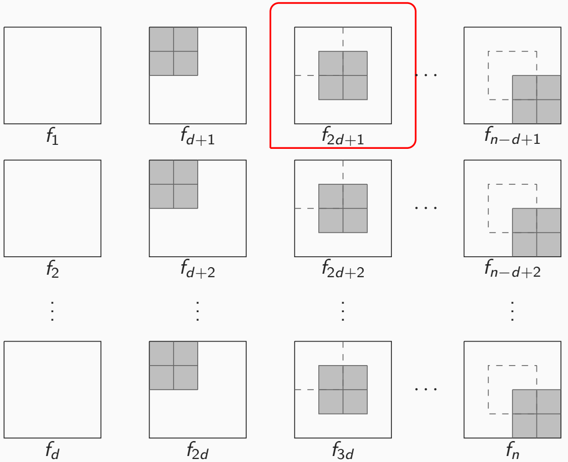
Example - Good keys for the MQQ cryptosystems

- **MQQ (Multivariate Quadratic Quasigroups)** [GMK08]
  - The private $\mathcal{F}$ - quasigroup string transformations of MQQs
  - direct algebraic attack
- **MQQ-SIG** [GOJPFKM11]
  - signature scheme
  - **fastest on (eBACS) SUPERCOP**
  - $n/2$ equations removed - measure against the attack
- **MQQ-ENC** [GS12]
  - Atempt on an encryption scheme
  - light use of minus modifier
- All broken [FGPST'15] due to rank defects and nice good keys

The MQQ central map

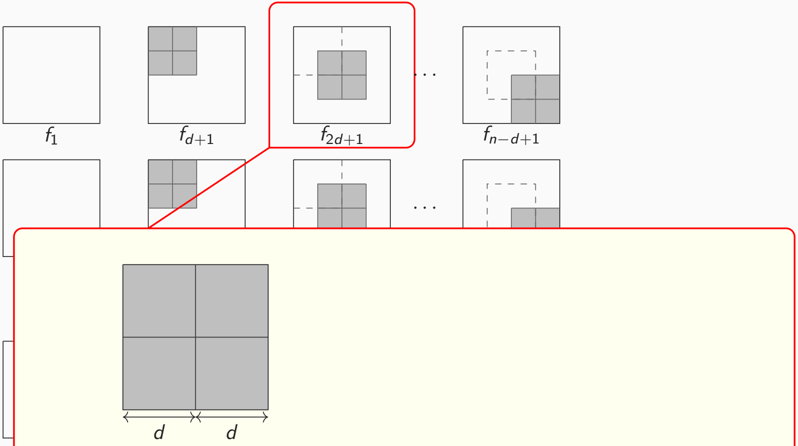


Matrix notation of $\mathcal{F}$



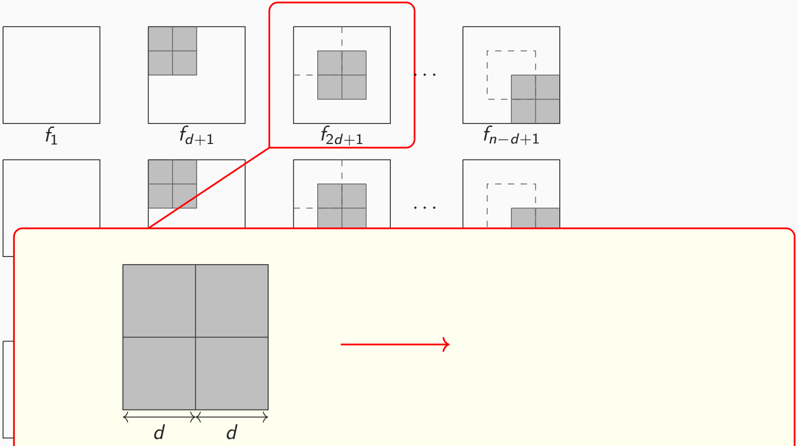
Matrix notation of $\mathcal{F}$

The MQQ central map



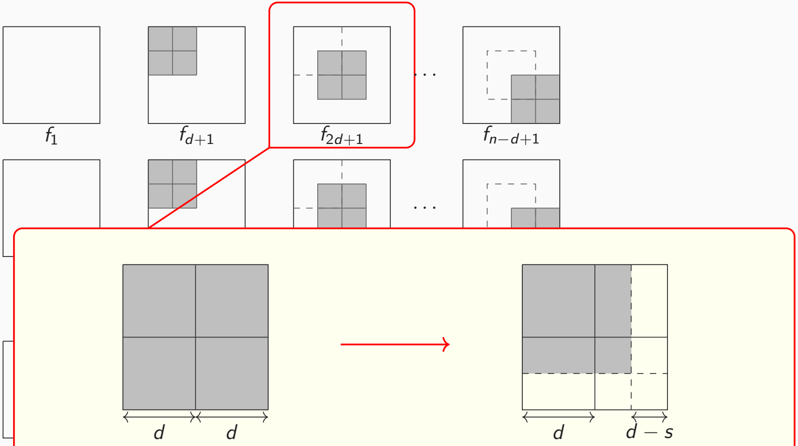
$$\mathcal{P} = \quad T \quad \circ \quad \mathcal{F} \quad \circ \quad S$$

The MQQ central map

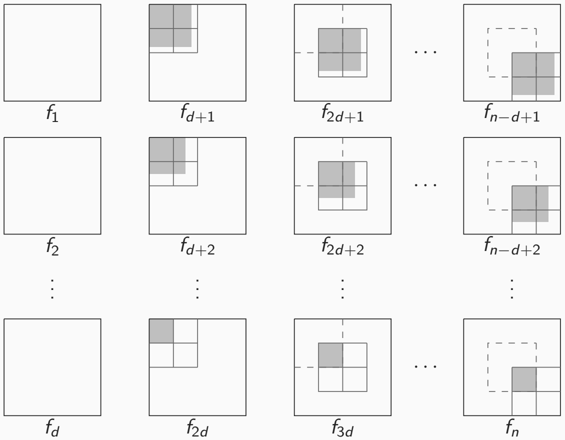


$\mathcal{P} = \quad T \quad \circ \quad \mathcal{F} \quad \circ \quad S$

The MQQ central map



$$\mathcal{P} = T \circ \mathcal{F} \circ S \qquad \mathcal{P} = (T \cdot B_1) \circ \mathcal{F}' \circ (B_2 \cdot S)$$



Matrix notation of $\mathcal{F}'$

Key Recovery Attack

Input: $n - r$ public polynomials $\mathcal{P}$ in n variables.

for number of variables $N := n$ down to $r + 2$ do

Step N**:**

Find a good key $(\overline{S}'_N, \overline{T}'_N)$

Transform the public key as $\mathcal{P} \leftarrow \overline{T}'_N \circ \mathcal{P} \circ \overline{S}'_N$,

end for;

Output: An equivalent key

$\overline{S}' = \overline{S}'_n \circ \overline{S}'_{n-1} \circ \cdots \circ \overline{S}'_{r+2}$ and $\overline{T}' = \overline{T}'_{r+2} \circ \cdots \circ \overline{T}'_{n-1} \circ \overline{T}'_n$.

Key Recovery Attack

Input: $n - r$ public polynomials $\mathcal{P}$ in n variables.

for number of variables $N := n$ down to $r + 2$ do

Step N**:**

Find a good key $(\overline{S}'_N, \overline{T}'_N)$

Transform the public key as $\mathcal{P} \leftarrow \overline{T}'_N \circ \mathcal{P} \circ \overline{S}'_N$,

end for;

Output: $\boxed{\text{An equivalent key } \overline{S}' = \overline{S}'_n \circ \overline{S}'_{n-1} \circ \cdots \circ \overline{S}'_{r+2} \text{ and } \overline{T}' = \overline{T}'_{r+2} \circ \cdots \circ \overline{T}'_{n-1} \circ \overline{T}'_n.}$

Essential structure preserved

Input: $n-r$ public polynomials $\mathcal{P}$ in n variables.

for number of variables $N := n$ down to $r+2$ do

The structure gradually revealed

Step N:

$\boxed{\textbf{Find a good key } (\overline{S}'_N, \overline{T}'_N)}$

Transform the public key as $\mathcal{P} \leftarrow \overline{T}'_N \circ \mathcal{P} \circ \overline{S}'_N$,

end for;

Output: **An equivalent key**
$\overline{S}' = \overline{S}'_n \circ \overline{S}'_{n-1} \circ \cdots \circ \overline{S}'_{r+2}$ and $\overline{T}' = \overline{T}'_{r+2} \circ \cdots \circ \overline{T}'_{n-1} \circ \overline{T}'_n$.

Essential structure preserved

Key Recovery Attack

Input: $n - r$ public polynomials $\mathcal{P}$ in n variables.

for number of variables $N := n$ down to $r + 2$ do

The structure gradually revealed

Step N:

one column at a time

$\boxed{\textbf{Find a good key } (\overline{S}'_N, \overline{T}'_N)}$

Transform the public key as $\mathcal{P} \leftarrow \overline{T}'_N \circ \mathcal{P} \circ \overline{S}'_N$,

end for;

Output: **An equivalent key**
$\overline{S}' = \overline{S}'_n \circ \overline{S}'_{n-1} \circ \cdots \circ \overline{S}'_{r+2}$ and $\overline{T}' = \overline{T}'_{r+2} \circ \cdots \circ \overline{T}'_{n-1} \circ \overline{T}'_n$.

Essential structure preserved

How to use good keys - a recipe

Consider the following "rewritting" of the public map

$$
\begin{aligned}
\mathcal{P} &= \mathcal{T} \circ \mathcal{F} \circ \mathcal{S} \Leftrightarrow \\
\mathcal{P} &= \underbrace{\mathcal{T} \circ \Sigma^{-1}} \circ \underbrace{\Sigma \circ \mathcal{F} \circ \Omega} \circ \underbrace{\Omega^{-1} \circ \mathcal{S}} \Leftrightarrow \\
\mathcal{P} &= \mathcal{T}' \circ \mathcal{F}' \circ \mathcal{S}'
\end{aligned}
$$

How to use good keys - a recipe

Consider the following "rewritting" of the public map

$$\begin{aligned}
\mathcal{P} &= \mathcal{T} \circ \mathcal{F} \circ \mathcal{S} \Leftrightarrow \\
\mathcal{P} &= \underbrace{\mathcal{T} \circ \Sigma^{-1}} \circ \underbrace{\Sigma \circ \mathcal{F} \circ \Omega} \circ \underbrace{\Omega^{-1} \circ \mathcal{S}} \Leftrightarrow \\
\mathcal{P} &= \mathcal{T}' \circ \mathcal{F}' \circ \mathcal{S}'
\end{aligned}$$

It does not matter whether we use $\mathcal{F}$ or $\mathcal{F}'$ as long as "essential" structure is preserved. We have then an "equivalent" key

How to use good keys - a recipe

Consider the following "rewritting" of the public map

$$
\begin{aligned}
\mathcal{P} &= \mathcal{T} \circ \mathcal{F} \circ \mathcal{S} \Leftrightarrow \\
\mathcal{P} &= \underbrace{\mathcal{T} \circ \Sigma^{-1}} \circ \underbrace{\Sigma \circ \mathcal{F} \circ \Omega} \circ \underbrace{\Omega^{-1} \circ \mathcal{S}} \Leftrightarrow \\
\mathcal{P} &= \mathcal{T}' \circ \mathcal{F}' \circ \mathcal{S}'
\end{aligned}
$$

It does not matter whether we use $\mathcal{F}$ or $\mathcal{F}'$ as long as "essential" structure is preserved. We have then an "equivalent" key

Use MinRank to recover an equivalent key in an array of steps

How to use good keys - a recipe

Consider the following "rewritting" of the public map

$$
\begin{array}{lcl}
\mathcal{P} & = & \mathcal{T} \quad \circ \quad \mathcal{F} \quad \circ \quad \mathcal{S} \Leftrightarrow \\
\mathcal{P} & = & \underbrace{\mathcal{T} \circ \Sigma^{-1}} \circ \underbrace{\Sigma \circ \mathcal{F} \circ \Omega} \circ \underbrace{\Omega^{-1} \circ \mathcal{S}} \Leftrightarrow \\
\mathcal{P} & = & \quad \mathcal{T}' \quad \circ \quad \mathcal{F}' \quad \circ \quad \mathcal{S}'
\end{array}
$$

It does not matter whether we use $\mathcal{F}$ or $\mathcal{F}'$ as long as "essential" structure is preserved. We have then an "equivalent" key

Use MinRank to recover an equivalent key in an array of steps

The actual recipe:

1. **Step** 1. Recover some structure (A "good" key)
2. **Step** 2. Some more structure (Another good key)
3. ...
4. **Step** *n*. All structure recovered (An equivalent key is found)

How to use it - a recipe

Recovering an equivalent key

Recovering an equivalent key

Rank defects
w.r.t. **unknown bases**

Public $\mathcal{P}$

Recovering an equivalent key

Rank defects
w.r.t. **unknown bases**

Goal: $\mathcal{P} \longrightarrow \mathcal{P}'$

Rank defects
w.r.t. **standard basis**

Recovering an equivalent key

Rank defects
w.r.t. **unknown bases**

Goal: $$\mathcal{P} = T' \circ \mathcal{P}' \circ S'$$

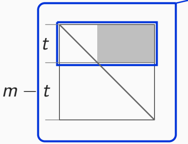
Rank defects
w.r.t. **standard basis**

Recovering an equivalent key

Rank defects
w.r.t. **unknown bases**

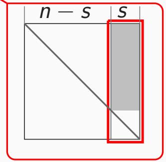
$$\mathcal{P} = T' \circ \mathcal{P}' \circ S'$$

Transform output space



Rank defects
w.r.t. **standard basis**

Transform input space



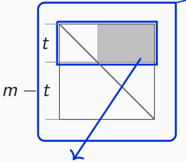
How to use it - a recipe

Recovering an equivalent key

Rank defects
w.r.t. **unknown bases**

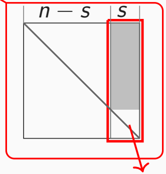
$$\mathcal{P} = T' \circ \mathcal{P}' \circ S'$$

Transform output space



Rank defects
w.r.t. **standard basis**

Transform input space



basis of polynomials

with common kernel

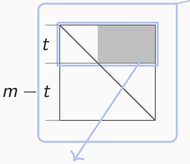
basis of kernel

How to use it - an improved recipe

Structure can be revealed gradually

$$\mathcal{P} \;=\; T' \circ \mathcal{P}' \circ S'$$

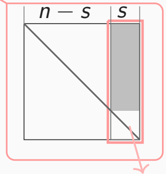
Transform output space



basis of polynomials

with common kernel B_P

Transform input space



basis of kernel B_K

How to use it - an improved recipe

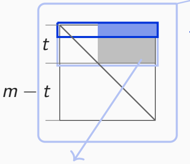
Structure can be revealed gradually

$$\mathcal{P} = T' \circ \mathcal{P}' \circ S'$$

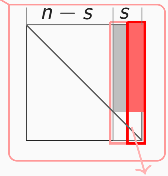
Transform output space

Transform input space

Find only c_1 vectors from B_P
Find only c_2 vectors from B_K



basis of polynomials
with common kernel B_P



basis of kernel B_K

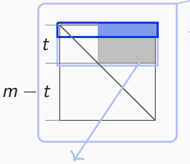
How to use it - an improved recipe

Structure can be revealed gradually

$$\mathcal{P} = T' \circ \mathcal{P}' \circ S'$$

Transform output space

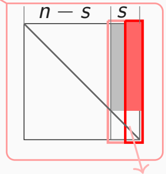
Transform input space



basis of polynomials

with common kernel B_P

Find only c_1 vectors from B_P
Find only c_2 vectors from B_K



basis of kernel B_K

Choose c_1 and c_2 s.t.
the system has unique solution

First (c_2, c_1) bilinear system
then (c_2, t) linear system
then (s, t) linear system

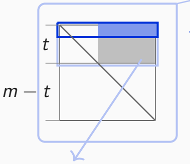
Structure can be revealed gradually

Type of **good key** [Thomae-Wolf '12]
with "good enough" structure

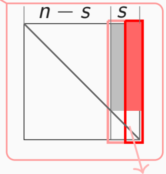
$$\mathcal{P} = T' \circ \mathcal{P}' \circ S'$$

Transform output space

Transform input space



Find only c_1 vectors from B_P
Find only c_2 vectors from B_K



Choose c_1 and c_2 s.t.
the system has unique solution

basis of polynomials
with common kernel B_P

First (c_2, c_1) bilinear system
then (c_2, t) linear system
then (s, t) linear system

basis of kernel B_K