Multivariate Fiat-Shamir signatures

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

Simona Samardjiska

July, 2024

Institute for Computing and Information Sciences
Radboud University

Recall the MQ problem from last time

Computational MQ problem

Given: m multivariate polynomials $p_1, p_2, \ldots, p_m \in \mathbb{F}_q[x_1, \ldots, x_n]$ of degree 2

Find: (if any) a vector $(u_1, \ldots, u_n) \in \mathbb{F}_q^n$ such that

$$\begin{cases} p_1(u_1, \ldots, u_n) = & 0 \\ p_2(u_1, \ldots, u_n) = & 0 \\ \quad \ldots & \\ p_m(u_1, \ldots, u_n) = & 0 \end{cases}$$

- Recall also that traditionally MQ schemes are ad-hoc
  - the hard problem is not the MQ problem, and not only the MQ problem
- What does it take to get a provably secure MQ scheme?
  - **MQDSS**: first signature with (lossy) ROM reduction to MQ
  - **SOFIA**: first signature with (lossy) QROM reduction to MQ

Some brainstorming in Sofia with Andy and Peter

- Lack of provable MQ signature
- Inefficient signatures from 3-pass IDS [Sakumoto et al. '11]
  - big soundness error (2/3)

Some brainstorming in Sofia with Andy and Peter

- Lack of provable MQ signature
- Inefficient signatures from 3-pass IDS [Sakumoto et al. '11]
  - big soundness error (2/3)
- Can we gain smth. if we consider signatures from 5-pass IDS?
  - smaller soundness error ($\frac{q+1}{2q}$ over $\mathbb{F}_q$) $\Rightarrow$ smaller signatures
  - FS transform for 5-pass already available [El Yousfi '12]
    - loose reduction in the ROM (as for 3-pass [Pointcheval & Stern '96])

Some brainstorming in Sofia with Andy and Peter

- Lack of provable MQ signature
- Inefficient signatures from 3-pass IDS [Sakumoto et al. '11]
  - big soundness error (2/3)
- Can we gain smth. if we consider signatures from 5-pass IDS?
  - smaller soundness error ($\frac{q+1}{2q}$ over $\mathbb{F}_q$) $\Rightarrow$ smaller signatures
  - FS transform for 5-pass already available [El Yousfi '12]
    - loose reduction in the ROM (as for 3-pass [Pointcheval & Stern '96])

Canonical Identification Schemes

$\mathcal{P}$, sk		$\mathcal{V}$, pk
$\mathsf{com} \leftarrow_R \mathcal{P}_0(\mathsf{sk})$	$\xrightarrow{\mathsf{com}}$	
		$\mathsf{ch} \leftarrow_R \mathsf{ChS}(1^k)$
	$\xleftarrow{\mathsf{ch}}$	
$\mathsf{resp} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch})$	$\xrightarrow{\mathsf{resp}}$	
		$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$

Informally:

(1) Prover commits to some (randomized) value derived from sk

(2) Verifier picks a challenge ‘ch’

(3) Prover computes response ‘resp’

(4) Verifier checks if response matches challenge

Properties of Canonical 3-pass IDS

$$
\begin{array}{lcl}
\mathcal{P}, \mathsf{sk} & & \mathcal{V}, \mathsf{pk} \\
\hline
\mathsf{com} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & \xrightarrow{\mathsf{com}} & \\
 & & \mathsf{ch} \leftarrow_R \mathsf{ChS}(1^k) \\
 & \xleftarrow{\mathsf{ch}} & \\
\mathsf{resp} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}) & & \\
 & \xrightarrow{\mathsf{resp}} & \\
 & & b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})
\end{array}
$$

- Special soundness
  There exists knowledge extractor $\mathcal{K}$ s.t. given two valid transcripts:

  $$\mathsf{trans} = (\mathsf{com}, \mathsf{ch}, \mathsf{resp}),\ \mathsf{trans}' = (\mathsf{com}, \mathsf{ch}', \mathsf{resp}'), \quad \mathsf{ch} \neq ch',$$

  extracts the secret sk with non-negligible probability
- (statistical) Honest-Verifier Zero-Knowledge
  There exists a PPT algorithm $\mathcal{S}$, called the simulator, such that the statistical distance between the real transcript and the simulated transcript is negligible in k.

The Fiat-Shamir transform

IDS

$\mathcal{P}$		$\mathcal{V}$
$\mathsf{com} \leftarrow \mathcal{P}_0\ (\mathsf{sk})$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow_R \mathsf{ChS}\ (1^k)$
$\mathsf{resp} \leftarrow \mathcal{P}_1\ (\mathsf{sk}, \mathsf{com}, \mathsf{ch})$	$\xrightarrow{\mathsf{resp}}$	
		$b \leftarrow \mathsf{Vf}\ (\mathsf{pk}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$

The Fiat-Shamir transform

IDS

$$
\begin{array}{lcr}
\mathcal{P}^{\color{red}r} & & \mathcal{V}^{\color{red}r} \\
\hline
\mathsf{com} \leftarrow \mathcal{P}_0{}^{\color{red}r}(\mathsf{sk}) & \xrightarrow{\quad \mathsf{com} \quad} & \\
 & & \mathsf{ch} \leftarrow_R \mathsf{ChS}^{\color{red}r}(1^k) \\
 & \xleftarrow{\quad \mathsf{ch} \quad} & \\
\mathsf{resp} \leftarrow \mathcal{P}_1{}^{\color{red}r}(\mathsf{sk}, \mathsf{com}, \mathsf{ch}) & \xrightarrow{\quad \mathsf{resp} \quad} & \\
 & & b \leftarrow \mathsf{Vf}^{\color{red}r}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})
\end{array}
$$

The Fiat-Shamir transform

IDS

$\mathcal{P}^r$		$\mathcal{V}^r$
$\mathsf{com} \leftarrow \mathcal{P}_0{}^r(\mathsf{sk})$	$\xrightarrow{\mathsf{com}}$	
$\mathsf{ch} \leftarrow_R \mathsf{ChS}^r(1^k)$	$\xleftarrow{\mathsf{ch}}$	
$\mathsf{resp} \leftarrow \mathcal{P}_1{}^r(\mathsf{sk}, \mathsf{com}, \mathsf{ch})$	$\xrightarrow{\mathsf{resp}}$	
		$b \leftarrow \mathsf{Vf}^r(\mathsf{pk}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$

$\downarrow \quad \downarrow \quad \downarrow$

FS signature

Signer
$\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk})$
$\mathsf{ch} \leftarrow H(m, \mathsf{com})$
$\mathsf{resp} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch})$
output : $\sigma = (\mathsf{com}, \mathsf{resp})$

Verifier
$\mathsf{ch} \leftarrow H(m, \mathsf{com})$
$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$
output : b

Security of FS signatures [Pointcheval & Stern '96]

IDS		**FS signature**
Special soundness	$\longrightarrow$	Secure under key-only attack

Security of FS signatures [Pointcheval & Stern '96]

IDS		**FS signature**
Special soundness	$\longrightarrow$	Secure under key-only attack

$$\mathcal{O} \leftrightarrow \mathcal{A}(\mathsf{pk}) \rightsquigarrow \sigma = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$$

Security of FS signatures [Pointcheval & Stern '96]

IDS		**FS signature**
Special soundness	$\longrightarrow$	Secure under key-only attack

$\mathcal{O}$

$\updownarrow$

$\mathcal{A}\,(\mathsf{pk}) \rightsquigarrow \sigma = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$

$\xrightarrow{\text{Fork.lemma}} \sigma' = (\mathsf{com}, \mathsf{ch}', \mathsf{resp}')$

Security of FS signatures [Pointcheval & Stern '96]

IDS		**FS signature**
Special soundness	$\longrightarrow$	Secure under key-only attack

$\mathcal{O}$

$\updownarrow$

$\mathcal{A}\,(\mathsf{pk}) \rightsquigarrow \sigma = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$

$\xrightarrow{\text{Fork.lemma}} \sigma' = (\mathsf{com}, \mathsf{ch}', \mathsf{resp}')$

$$\mathsf{trans} = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$$

$$\mathsf{trans}' = (\mathsf{com}, \mathsf{ch}', \mathsf{resp}')$$

IDS | **FS signature**

Special soundness $\longrightarrow$ Secure under key-only attack

$\mathcal{O}$

$\mathcal{A}\,(\mathsf{pk}) \rightsquigarrow \sigma = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$

$\sigma' = (\mathsf{com}, \mathsf{ch}', \mathsf{resp}')$

Fork.lemma

$\mathsf{trans} = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$

$\mathsf{trans}' = (\mathsf{com}, \mathsf{ch}', \mathsf{resp}')$

$\xrightarrow{\mathcal{K}}$ sk

Security of FS signatures [Pointcheval & Stern '96]

IDS		**FS signature**
Special soundness	$\longrightarrow$	Secure under key-only attack
$+$	$\searrow$	
HVZK	$\longrightarrow$	**eu-cma secure**

Security of FS signatures [Pointcheval & Stern ’96]

IDS		**FS signature**
Special soundness	$\longrightarrow$	Secure under key-only attack
$+$		
HVZK	$\longrightarrow$	**eu-cma secure**

$$\mathcal{O} \longleftrightarrow \mathcal{B}(\mathsf{pk}) \longleftrightarrow \mathcal{S}ign$$

$$\mathcal{B}(\mathsf{pk}) \rightsquigarrow \sigma = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$$

Security of FS signatures [Pointcheval & Stern '96]

IDS		**FS signature**
Special soundness	$\longrightarrow$	Secure under key-only attack
+		
HVZK	$\longrightarrow$	**eu-cma secure**

$\mathcal{O}$

$\updownarrow$

$\mathcal{B}(\mathsf{pk}) \rightsquigarrow \sigma = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$

$\updownarrow$

$\mathcal{S}$

Security of FS signatures [Pointcheval & Stern '96]

IDS		**FS signature**
Special soundness	$\longrightarrow$	Secure under key-only attack
$+$		
HVZK	$\longrightarrow$	**eu-cma secure**

$\mathcal{O} \leftrightarrow \mathcal{B}(\mathrm{pk}) \leftrightarrow \mathcal{S}$

$\mathcal{B}(\mathrm{pk}) \rightsquigarrow \sigma = (\mathrm{com}, \mathrm{ch}, \mathrm{resp})$

$\Rightarrow \quad \mathcal{A}(pk) \rightsquigarrow \mathrm{sk}$

Canonical 5-pass IDS

$$
\begin{array}{lcl}
\mathcal{P} & & \mathcal{V} \\
\hline
\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk}) & \xrightarrow{\quad \mathsf{com} \quad} & \\
 & \xleftarrow{\quad \mathsf{ch}_1 \quad} & \mathsf{ch}_1 \leftarrow_R \mathsf{ChS}_1(1^k) \\
\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1) & \xrightarrow{\quad \mathsf{resp}_1 \quad} & \\
 & \xleftarrow{\quad \mathsf{ch}_2 \quad} & \mathsf{ch}_2 \leftarrow_R \mathsf{ChS}_2(1^k) \\
\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2) & \xrightarrow{\quad \mathsf{resp}_2 \quad} & \\
 & & b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)
\end{array}
$$

The Fiat-Shamir transform on 5-pass IDS

IDS

$\mathcal{P}$		$\mathcal{V}$
$\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk})$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}_1}$	$\mathsf{ch}_1 \leftarrow_R \mathsf{ChS}_1(1^k)$
$\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1)$	$\xrightarrow{\mathsf{resp}_1}$	
	$\xleftarrow{\mathsf{ch}_2}$	$\mathsf{ch}_2 \leftarrow_R \mathsf{ChS}_2(1^k)$
$\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2)$	$\xrightarrow{\mathsf{resp}_2}$	
		$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$

The Fiat-Shamir transform on 5-pass IDS

IDS

$\mathcal{P}$		$\mathcal{V}$
$\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk})$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}_1}$	$\mathsf{ch}_1 \leftarrow_R \mathsf{ChS}_1(1^k)$
$\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1)$	$\xrightarrow{\mathsf{resp}_1}$	
	$\xleftarrow{\mathsf{ch}_2}$	$\mathsf{ch}_2 \leftarrow_R \mathsf{ChS}_2(1^k)$
$\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2)$	$\xrightarrow{\mathsf{resp}_2}$	
		$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$

$\downarrow \; \downarrow \; \downarrow$

FS signature

Signer

$\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk})$

$\mathsf{ch}_1 \leftarrow H_1(m, \mathsf{com})$

$\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1)$

$\mathsf{ch}_2 \leftarrow H_2(m, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1)$

$\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2)$

output : $\sigma = (\mathsf{com}, \mathsf{resp}_1, \mathsf{resp}_2)$

Verifier

$\mathsf{ch}_1 \leftarrow H_1(m, \mathsf{com})$

$\mathsf{ch}_2 \leftarrow H_2(m, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1)$

$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$

output : b

Sakumoto-Shirai-Hiwatari 5-pass IDS

$\mathcal{P}(\mathbf{F}, \mathbf{v}, \mathbf{s})$		$\mathcal{V}(\mathbf{F}, \mathbf{v})$
$\mathbf{r}_0, \mathbf{t}_0 \leftarrow_R \mathbb{F}_q^n, \mathbf{e}_0 \leftarrow_R \mathbb{F}_q^m$		
$\mathbf{r}_1 \leftarrow \mathbf{s} - \mathbf{r}_0$		
$c_0 \leftarrow Com(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$		
$c_1 \leftarrow Com(\mathbf{r}_1, \mathbf{G}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0)$	$\xrightarrow{(c_0, c_1)}$	
	$\xleftarrow{\alpha}$	$\alpha \leftarrow_R \mathbb{F}_q$
$\mathbf{t}_1 \leftarrow \alpha \mathbf{r}_0 - \mathbf{t}_0$		
$\mathbf{e}_1 \leftarrow \alpha \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_0$	$\xrightarrow{\text{resp}_1 = (\mathbf{t}_1, \mathbf{e}_1)}$	
	$\xleftarrow{\text{ch}_2}$	$\text{ch}_2 \leftarrow_R \{0, 1\}$
If $\text{ch}_2 = 0$, $\text{resp}_2 \leftarrow \mathbf{r}_0$		
Else $\text{resp}_2 \leftarrow \mathbf{r}_1$	$\xrightarrow{\text{resp}_2}$	
		If $\text{ch}_2 = 0$, Parse $\text{resp}_2 = \mathbf{r}_0$, check $c_0 \stackrel{?}{=} Com(\mathbf{r}_0, \alpha \mathbf{r}_0 - \mathbf{t}_1, \alpha \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_1)$
		Else Parse $\text{resp}_2 = \mathbf{r}_1$, check $c_1 \stackrel{?}{=} Com(\mathbf{r}_1, \alpha(\mathbf{v} - \mathbf{F}(\mathbf{r}_1)) - \mathbf{G}(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{e}_1)$

Sakumoto-Shirai-Hiwatari IDS

- Smaller soundness error ($\frac{q+1}{2q}$ over $\mathbb{F}_q$) $\Rightarrow$ **smaller signatures**
- Key technique: **cut-and-choose for** MQ

Sakumoto-Shirai-Hiwatari IDS

- Smaller soundness error ($\frac{q+1}{2q}$ over $\mathbb{F}_q$) $\Rightarrow$ **smaller signatures**
- Key technique: **cut-and-choose for** MQ

- Bilinear map $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$
  - Split $\mathbf{s}$ and $\mathbf{F}(\mathbf{s})$ into $\mathbf{r}_0, \mathbf{r}_1$ and $\mathbf{F}(\mathbf{r}_0), \mathbf{F}(\mathbf{r}_1)$
    - Since then $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1 \Rightarrow \mathbf{F}(\mathbf{s}) = \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1)$
  - Split $\mathbf{r}_0$ and $\mathbf{F}(\mathbf{r}_0)$ further into $\mathbf{t}_0, \mathbf{t}_1$ resp. $\mathbf{e}_0, \mathbf{e}_1$

Sakumoto-Shirai-Hiwatari IDS

- Smaller soundness error ($\frac{q+1}{2q}$ over $\mathbb{F}_q$) $\Rightarrow$ **smaller signatures**
- Key technique: **cut-and-choose for** MQ

- Bilinear map $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$
  - Split $\mathbf{s}$ and $\mathbf{F}(\mathbf{s})$ into $\mathbf{r}_0, \mathbf{r}_1$ and $\mathbf{F}(\mathbf{r}_0), \mathbf{F}(\mathbf{r}_1)$
    - Since then $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1 \Rightarrow \mathbf{F}(\mathbf{s}) = \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1)$
  - Split $\mathbf{r}_0$ and $\mathbf{F}(\mathbf{r}_0)$ further into $\mathbf{t}_0, \mathbf{t}_1$ resp. $\mathbf{e}_0, \mathbf{e}_1$

Sakumoto-Shirai-Hiwatari IDS

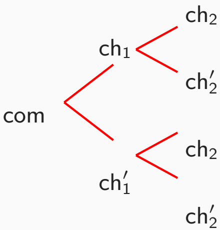
- Smaller soundness error ($\frac{q+1}{2q}$ over $\mathbb{F}_q$) $\Rightarrow$ **smaller signatures**
- Key technique: **cut-and-choose for** MQ

- Bilinear map $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$
  - Split $\mathbf{s}$ and $\mathbf{F}(\mathbf{s})$ into $\mathbf{r}_0, \mathbf{r}_1$ and $\mathbf{F}(\mathbf{r}_0), \mathbf{F}(\mathbf{r}_1)$
    - Since then $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1 \Rightarrow \mathbf{F}(\mathbf{s}) = \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1)$
  - Split $\mathbf{r}_0$ and $\mathbf{F}(\mathbf{r}_0)$ further into $\mathbf{t}_0, \mathbf{t}_1$ resp. $\mathbf{e}_0, \mathbf{e}_1$
  - $\mathbf{r_0} = \mathbf{t_0} + \mathbf{t_1}$
  - $\mathbf{F}(\mathbf{r_0}) = \mathbf{e_0} + \mathbf{e_1}$
- Using bilinearity, $\mathbf{v} = (\mathbf{G}(\mathbf{t_0}, \mathbf{r_1}) + \mathbf{e_0}) + (\mathbf{F}(\mathbf{r_1}) + \mathbf{G}(\mathbf{t_1}, \mathbf{r_1}) + \mathbf{e_1})$
- Result: reveal either $\mathbf{r}_0$ or $\mathbf{r}_1$, and $(\mathbf{t}_1, \mathbf{e}_1)$
- **Zero knowledge property satisfied**

Sakumoto et al. 5-pass IDS

The extractor $\mathcal{K}$ **needs 4 valid transcripts!**

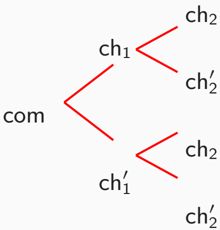
$$(\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$$
$$(\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2', \mathsf{resp}_2')$$
$$(\mathsf{com}, \mathsf{ch}_1', \mathsf{resp}_1', \mathsf{ch}_2, \mathsf{resp}_2'')$$
$$(\mathsf{com}, \mathsf{ch}_1', \mathsf{resp}_1', \mathsf{ch}_2', \mathsf{resp}_2''')$$



Sakumoto et al. 5-pass IDS

The extractor $\mathcal{K}$ **needs 4 valid transcripts!**

$$(\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$$
$$(\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2', \mathsf{resp}_2')$$
$$(\mathsf{com}, \mathsf{ch}_1', \mathsf{resp}_1', \mathsf{ch}_2, \mathsf{resp}_2'')$$
$$(\mathsf{com}, \mathsf{ch}_1', \mathsf{resp}_1', \mathsf{ch}_2', \mathsf{resp}_2''')$$



- Focus attention on 5-pass IDS with second challenge space $|\mathsf{ChS}_2| = 2$
  - Sakumoto et al. 5-pass IDS is such
  - Most in the literature are such

Next step ...

What is the problem with FS proof in the QROM?

- We need **to see** the signature σ before rewinding
- We need **to see** the oracle inputs
- Seeing (measuring) destroys the quantum state
- The proof fails terribly

Next step ...MQ signatures in the QROM

What is the problem with FS proof in the QROM?

- We need **to see** the signature σ before rewinding
- We need **to see** the oracle inputs
- Seeing (measuring) destroys the quantum state
- The proof fails terribly

A solution: Unruh transform [Unruh '14] adapted for $q2$ IDS

- Online extractability
- We can extract the witness without rewinding
- Enough transcripts directly available

Next step ...MQ signatures in the QROM

What is the problem with FS proof in the QROM?

- We need **to see** the signature σ before rewinding
- We need **to see** the oracle inputs
- Seeing (measuring) destroys the quantum state
- The proof fails terribly

A solution: Unruh transform [Unruh '14] adapted for $q2$ IDS

- Online extractability
- We can extract the witness without rewinding
- Enough transcripts directly available

We proposed SOFIA - MQ signature secure in the QROM

MQDSS vs SOFIA

	Sec.	q	n $(= m)$	r	pk (bytes)	sk (bytes)	Signature (bytes)
MQDSS-31-64 (AC '16)	128 (ROM)	31	48	269	72	64	40952
SOFIA-4-128 (PKC '18)	128 (QROM)	4	128	438	64	32	126176

- SOFIA still comparable to Picnic (with QROM proof),
- but much slower than SPHINCS + and lattice based schemes

NIST parameter sets MQDSS

	Sec. cat.	q	n $(= m)$	r	pk (bytes)	sk (bytes)	Signature (bytes)
MQDSS-31-48 (Round 2)	1-2	31	48	135	46	16	20854
MQDSS-31-64 (Round 2)	3-4	31	64	202	64	24	43728

An attack on MQDSS

- August 2019, Daniel Kales and Greg Zaverucha - forgery in approx. 2^{95} hash calls for MQDSS-31-48
- Can be mitigated by $\approx 1.4\times$(number of rounds)
- **Proof still valid!**
  - Attack is result of not taking into account non-tightness of proof for choosing parameters

An attack on MQDSS

- August 2019, Daniel Kales and Greg Zaverucha - forgery in approx. 2^{95} hash calls for MQDSS-31-48
- Can be mitigated by $\approx 1.4\times$(number of rounds)
- **Proof still valid!**
  - Attack is result of not taking into account non-tightness of proof for choosing parameters
- **New parameters after attack (estimate):**

	Sec. cat.	q	n	r	pk	sk	Signature
MQDSS-31-48 (new)	1-2	31	48	184	46B	16B	28400B
MQDSS-31-48 (Round 2)	1-2	31	48	135	46**B**	16**B**	20854**B**
MQDSS-31-64 (new)	3-4	31	64	277	64B	24B	59928B
MQDSS-31-64 (Round 2)	3-4	31	64	202	64**B**	24**B**	43728**B**

Developments during the NIST competition

- Fiat-Shamir shown secure in the QROM - SOFIA becomes superfluous
- MQDSS proven secure in the QROM - still, sizes are huge
- several approaches that drastically improve the signature size
- Mudfish [Beullens, Eurocrypt '20]
  - Idea to reduce soundness error by introducing a preprocessing phase with a trusted **Helper**
  - And then have a regular Σ-protocol (satisfies completeness, special soundness, HVZK)
  - Takes inspiration from SOFIA and MPC-in-the-head [Katz, Kolesnikov, Wang, '18]
- MEDS, ALTEQ - Fiat-Shamir Goldreich-Micali-Wigderson scheme based on variants of Isomorphism of Polynomials
- MQOM - Fiat-Shamir based on MPC-in-the-head paradigm

Zero-Knowledge Interactive Proof of knowledge from Equivalence Problems

[Goldreich–Micali–Wigderson '91]:

Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.

Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{array}{c} \mathcal{O}_0 \\ \Big\downarrow{\scriptstyle \phi} \\ \mathcal{O}_1 \end{array}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$	$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$

Zero-Knowledge Interactive Proof of knowledge from Equivalence Problems

[Goldreich–Micali–Wigderson '91]:

Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.

Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{tikzcd} \mathcal{O}_0 \arrow[r, "\phi_0"] \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\ \mathcal{O}_1 & \end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$	$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$

Zero-Knowledge Interactive Proof of knowledge from Equivalence Problems

[Goldreich–Micali–Wigderson '91]:

Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.

Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{tikzcd}
\mathcal{O}_0 \arrow[r, "\phi_0"] \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\
\mathcal{O}_1 \arrow[ur, "\phi_1"'] &
\end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$	$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$

Zero-Knowledge Interactive Proof of knowledge from Equivalence Problems

[Goldreich–Micali–Wigderson ’91]:
Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.
Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{tikzcd} \mathcal{O}_0 \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\ \mathcal{O}_1 & \end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$	$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
com $\leftarrow \mathcal{O}'$	

Zero-Knowledge Interactive Proof of knowledge from Equivalence Problems

[Goldreich–Micali–Wigderson '91]:

Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.

Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{tikzcd} \mathcal{O}_0 \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\ \mathcal{O}_1 & \end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
$\mathsf{com} \leftarrow \mathcal{O}'$	$\xrightarrow{\mathsf{com}}$	

Zero-Knowledge Interactive Proof of knowledge from Equivalence Problems

[Goldreich–Micali–Wigderson '91]:
Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.
Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{tikzcd}
\mathcal{O}_0 \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\
\mathcal{O}_1 &
\end{tikzcd}$$

$$\begin{array}{lcr}
\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi) & & \mathcal{V}(\mathcal{O}_0, \mathcal{O}_1) \\
\hline
\mathsf{com} \leftarrow \mathcal{O}' & & \\
 & \xrightarrow{\quad \mathsf{com} \quad} & \\
 & & \mathsf{ch} \leftarrow_R \{0, 1\}
\end{array}$$

[Goldreich–Micali–Wigderson '91]:
Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.
Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{tikzcd} \mathcal{O}_0 \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\ \mathcal{O}_1 & \end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
com $\leftarrow \mathcal{O}'$	$\xrightarrow{\text{com}}$	
	$\xleftarrow{\text{ch}}$	ch $\leftarrow_R \{0, 1\}$

[Goldreich–Micali–Wigderson '91]:
Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.
Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{tikzcd} \mathcal{O}_0 \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\ \mathcal{O}_1 & \end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
$\mathsf{com} \leftarrow \mathcal{O}'$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow_R \{0, 1\}$
$\mathsf{resp} \leftarrow \phi_{\mathsf{ch}}$		

Zero-Knowledge Interactive Proof of knowledge from Equivalence Problems

[Goldreich–Micali–Wigderson '91]:

Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.

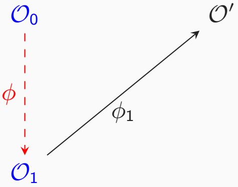
Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{tikzcd}
\mathcal{O}_0 \arrow[r, "\phi_0"] \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\
\mathcal{O}_1 &
\end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
$\mathsf{com} \leftarrow \mathcal{O}'$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow_R \{0, 1\}$
$\mathsf{resp} \leftarrow \phi_{\mathsf{ch}}$	$\xrightarrow{\mathsf{resp}}$	

Zero-Knowledge Interactive Proof of knowledge from Equivalence Problems

[Goldreich–Micali–Wigderson '91]:
Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.
Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it



$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
com $\leftarrow \mathcal{O}'$	$\xrightarrow{\text{com}}$	
	$\xleftarrow{\text{ch}}$	ch $\leftarrow_R \{0, 1\}$
resp $\leftarrow \phi_{\text{ch}}$	$\xrightarrow{\text{resp}}$	

Zero-Knowledge Interactive Proof of knowledge from Equivalence Problems

[Goldreich–Micali–Wigderson '91]:

Let ϕ be an isomorphism s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$.

Given $\mathcal{O}_0, \mathcal{O}_1$, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of ϕ without revealing any information about it

$$\begin{tikzcd}
\mathcal{O}_0 \arrow[r, "\phi_0"] \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\
\mathcal{O}_1 \arrow[ur, "\phi_1"'] &
\end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
$\mathsf{com} \leftarrow \mathcal{O}'$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow_R \{0, 1\}$
$\mathsf{resp} \leftarrow \phi_{\mathsf{ch}}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}' \overset{?}{=} \phi_{\mathsf{ch}}(\mathcal{O}_{\mathsf{ch}})$

Interesting concrete hard equivalence problems?

Generic hard equivalence problem $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
Given $\mathcal{O}_0$ and $\mathcal{O}_1$, find (if any) an isomorphism ϕ s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

Interesting concrete hard equivalence problems?

Generic hard equivalence problem $\mathsf{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
Given $\mathcal{O}_0$ and $\mathcal{O}_1$, find (if any) an isomorphism ϕ s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

Interesting concrete hard equivalence problems?

Generic hard equivalence problem $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
Given $\mathcal{O}_0$ and $\mathcal{O}_1$, find (if any) an isomorphism ϕ s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

- **Isomorphism of polynomials** - Patarin's signature, 1998

Interesting concrete hard equivalence problems?

Generic hard equivalence problem $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
Given $\mathcal{O}_0$ and $\mathcal{O}_1$, find (if any) an isomorphism ϕ s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

- **Isomorphism of polynomials** - Patarin's signature, 1998
- **Quasigroup isotopy** - Identification scheme, Denes, 2001

Interesting concrete hard equivalence problems?

Generic hard equivalence problem $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
Given $\mathcal{O}_0$ and $\mathcal{O}_1$, find (if any) an isomorphism ϕ s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

- **Isomorphism of polynomials** - Patarin's signature, 1998
- **Quasigroup isotopy** - Identification scheme, Denes, 2001
- **Isogeny on eliptic curves** - SeaSign 2018, SqiSign 2020, De Feo et al.

Interesting concrete hard equivalence problems?

Generic hard equivalence problem $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
Given $\mathcal{O}_0$ and $\mathcal{O}_1$, find (if any) an isomorphism ϕ s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

- **Isomorphism of polynomials** - Patarin's signature, 1998
- **Quasigroup isotopy** - Identification scheme, Denes, 2001
- **Isogeny on eliptic curves** - SeaSign 2018, SqiSign 2020, De Feo et al.
- **Code equivalence** - LESS - Biasse et al. 2020, LESS-FM - Barenghi et al. 2021

Interesting concrete hard equivalence problems?

Generic hard equivalence problem $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
Given $\mathcal{O}_0$ and $\mathcal{O}_1$, find (if any) an isomorphism ϕ s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

- **Isomorphism of polynomials** - Patarin's signature, 1998
- **Quasigroup isotopy** - Identification scheme, Denes, 2001
- **Isogeny on eliptic curves** - SeaSign 2018, SqiSign 2020, De Feo et al.
- **Code equivalence** - LESS - Biasse et al. 2020, LESS-FM - Barenghi et al. 2021
- **Alternate trilinear form equivalence** - Tang et al. 2022

Interesting concrete hard equivalence problems?

Generic hard equivalence problem $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
Given $\mathcal{O}_0$ and $\mathcal{O}_1$, find (if any) an isomorphism ϕ s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

- **Isomorphism of polynomials** - Patarin's signature, 1998
- **Quasigroup isotopy** - Identification scheme, Denes, 2001
- **Isogeny on eliptic curves** - SeaSign 2018, SqiSign 2020, De Feo et al.
- **Code equivalence** - LESS - Biasse et al. 2020, LESS-FM - Barenghi et al. 2021
- **Alternate trilinear form equivalence** - Tang et al. 2022
- **Lattice isomorphism** - Ducas and van Woerden 2022

Interesting concrete hard equivalence problems?

Generic hard equivalence problem $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
Given $\mathcal{O}_0$ and $\mathcal{O}_1$, find (if any) an isomorphism ϕ s.t. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

- **Isomorphism of polynomials** - Patarin's signature, 1998
- **Quasigroup isotopy** - Identification scheme, Denes, 2001
- **Isogeny on eliptic curves** - SeaSign 2018, SqiSign 2020, De Feo et al.
- **Code equivalence** - LESS - Biasse et al. 2020, LESS-FM - Barenghi et al. 2021
- **Alternate trilinear form equivalence** - Tang et al. 2022
- **Lattice isomorphism** - Ducas and van Woerden 2022
- **Matrix code equivalence** - with Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Tovohery Hajatiana Randrianarisoa, Lars Ran, Krijn Reijnders, Monika Trimoska , 2022
- ...

The Matrix Code Equivalence Problem

Matrix code - a subspace of $\mathcal{M}_{m\times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**

$$d(\mathbf{A}, \mathbf{B}) = \mathrm{Rank}(\mathbf{A} - \mathbf{B})$$

The Matrix Code Equivalence Problem

Matrix code - a subspace of $\mathcal{M}_{m\times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**

$$d(\mathbf{A},\mathbf{B}) = \mathrm{Rank}(\mathbf{A}-\mathbf{B})$$

Matrix Code Equivalence (MCE) problem [Berger,2003]

Input: Two k-dimensional matrix codes $\mathcal{C},\mathcal{D} \subset \mathcal{M}_{m,n}(q)$

Question: Find – if any – $\mathbf{A} \in \mathrm{GL}_m(q), \mathbf{B} \in \mathrm{GL}_n(q)$ (an isometry) s.t. for all $\mathbf{C} \in \mathcal{C}$, it holds that

$$\mathbf{ACB} \in \mathcal{D}$$

The Matrix Code Equivalence Problem

Matrix code - a subspace of $\mathcal{M}_{m \times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**

$$d(\mathbf{A}, \mathbf{B}) = \mathrm{Rank}(\mathbf{A} - \mathbf{B})$$

Matrix Code Equivalence (MCE) problem [Berger,2003]

Input: Two k-dimensional matrix codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m,n}(q)$

Question: Find – if any – $\mathbf{A} \in \mathrm{GL}_m(q), \mathbf{B} \in \mathrm{GL}_n(q)$ (an isometry) s.t. for all $\mathbf{C} \in \mathcal{C}$, it holds that

$$\mathbf{ACB} \in \mathcal{D}$$

Related problems

- *Matrix Codes Right (Left) Equivalence* problem (MCRE) – **A** (**B**) is trivial

Matrix code - a subspace of $\mathcal{M}_{m\times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**

$$d(\mathbf{A},\mathbf{B}) = \mathrm{Rank}(\mathbf{A}-\mathbf{B})$$

Matrix Code Equivalence (MCE) problem [Berger,2003]

Input: Two k-dimensional matrix codes $\mathcal{C},\mathcal{D} \subset \mathcal{M}_{m,n}(q)$

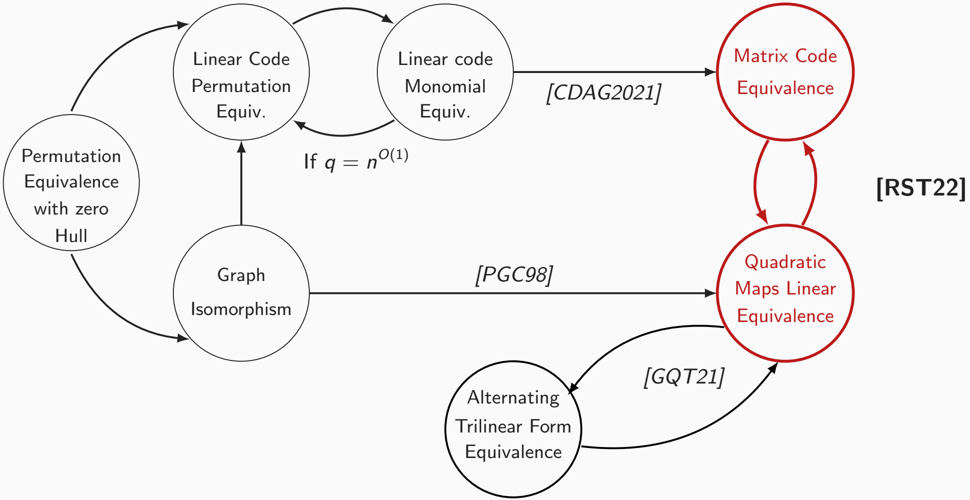
Question: Find – if any – $\mathbf{A} \in \mathrm{GL}_m(q), \mathbf{B} \in \mathrm{GL}_n(q)$ (an isometry) s.t. for all $\mathbf{C} \in \mathcal{C}$, it holds that

$$\mathbf{ACB} \in \mathcal{D}$$

Related problems

- *Matrix Codes Right (Left) Equivalence* problem (MCRE) – $\mathbf{A}$ ($\mathbf{B}$) is trivial
- $\mathbb{F}_{q^m}$*-linear codes* – MCE reduces to MCRE

Known results - relations to other problems



Quadratic Maps Linear Equivalence (QMLE) problem

Introduced by Patarin 1996 as Isomorphism of Polynomials (IP) problem for building an identification scheme and FS signature!

Quadratic Maps Linear Equivalence (QMLE) problem

Input: Two k-tuples of quadratic multivariate polynomials $\mathcal{F},\ \mathcal{P} \in \mathbb{F}_q[x_1, \ldots, x_n]^k$

Question: Find – if any – invertible matrices $\mathbf{S}, \mathbf{T}$ such that

$$\mathcal{P}(\mathbf{x}) = \mathbf{T}\mathcal{F}(\mathbf{S}\mathbf{x}).$$

Quadratic Maps Linear Equivalence (QMLE) problem

Introduced by Patarin 1996 as Isomorphism of Polynomials (IP) problem for building an identification scheme and FS signature!

Quadratic Maps Linear Equivalence (QMLE) problem

Input: Two k-tuples of quadratic multivariate polynomials $\mathcal{F}, \mathcal{P} \in \mathbb{F}_q[x_1, \ldots, x_n]^k$

Question: Find – if any – invertible matrices $\mathbf{S}, \mathbf{T}$ such that

$$\mathcal{P}(\mathbf{x}) = \mathbf{T}\mathcal{F}(\mathbf{S}\mathbf{x}).$$

Related problems

- *Isomorphism of Polynomials with one secret* (IP1$\mathcal{S}$), when $\mathbf{T}$ is trivial - easy

Quadratic Maps Linear Equivalence (QMLE) problem

Introduced by Patarin 1996 as Isomorphism of Polynomials (IP) problem for building an identification scheme and FS signature!

Quadratic Maps Linear Equivalence (QMLE) problem

Input: Two k-tuples of quadratic multivariate polynomials $\mathcal{F}, \mathcal{P} \in \mathbb{F}_q[x_1, \ldots, x_n]^k$

Question: Find – if any – invertible matrices $\mathbf{S}, \mathbf{T}$ such that

$$\mathcal{P}(\mathbf{x}) = \mathbf{T}\mathcal{F}(\mathbf{S}\mathbf{x}).$$

Related problems

- *Isomorphism of Polynomials with one secret* (IP1$\mathcal{S}$), when $\mathbf{T}$ is trivial - easy
- **homogenous version** hQMLE - **hard**
- inhomogenous version - easy (heuristic result [FP06])

Alternating trilinear form equivalence problem (ATFE)

Alternating trilinear form: $\phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{1 \leqslant i < j < s \leqslant n} c_{ijs} \begin{vmatrix} x_i & y_i & z_i \\ x_j & y_j & z_j \\ x_s & y_s & z_s \end{vmatrix}$ where $c_{ijs} \in \mathbb{F}_q$.

- Can be stored using $\binom{n}{3}$ entries: one for each c_{ijs} coefficient

Alternating trilinear form equivalence (ATFE) problem

Input: Two alternating trilinear forms $\phi, \psi \in \mathbb{F}_q[\mathbf{x}, \mathbf{y}, \mathbf{z}]$.

Question: Find – if any – invertible $\mathbf{A}$ such that $\psi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \phi(\mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{y}, \mathbf{A}\mathbf{z})$.

Alternating trilinear form equivalence problem (ATFE)

Alternating trilinear form: $\phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{1 \leqslant i < j < s \leqslant n} c_{ijs} \begin{vmatrix} x_i & y_i & z_i \\ x_j & y_j & z_j \\ x_s & y_s & z_s \end{vmatrix}$ where $c_{ijs} \in \mathbb{F}_q$.

- Can be stored using $\binom{n}{3}$ entries: one for each c_{ijs} coefficient

Alternating trilinear form equivalence (ATFE) problem

Input: Two alternating trilinear forms $\phi, \psi \in \mathbb{F}_q[\mathbf{x}, \mathbf{y}, \mathbf{z}]$.

Question: Find – if any – invertible $\mathbf{A}$ such that $\psi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \phi(\mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{y}, \mathbf{A}\mathbf{z})$.

- Used by Tang et al. '22 to build a signature scheme with competitive signature sizes
- Shown to be equivalent to hQMLE (Grochow et al. '21)

MCE, QMLE and ATFE look very similar!

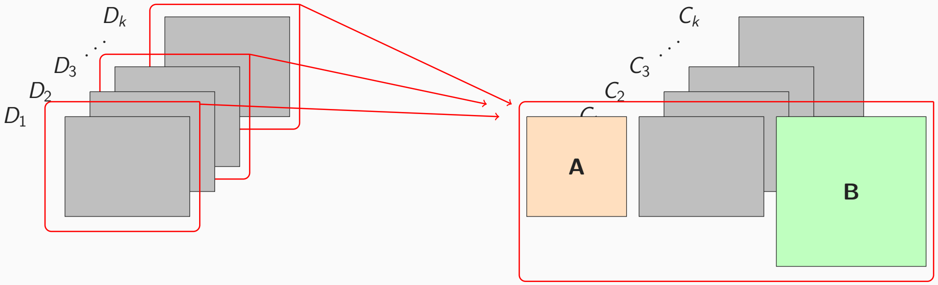
Matrix codes:

MCE:

- matrix codes of rectangular matrices

MCE, QMLE **and** ATFE **look very similar!**

Matrix codes:

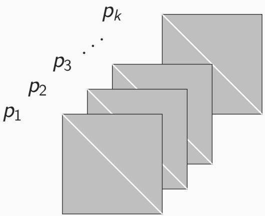


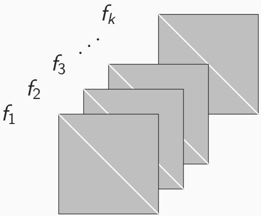
MCE:

- matrix codes of rectangular matrices
- isometry $(\mathbf{A}, \mathbf{B})$

MCE, QMLE **and** ATFE **look very similar!**

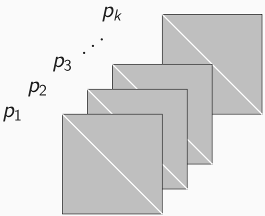
Systems of polynomials in matrix representation:

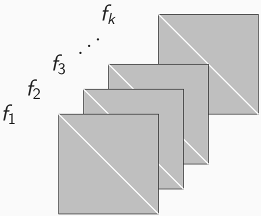




MCE, QMLE **and** ATFE **look very similar!**

Systems of polynomials in matrix representation:



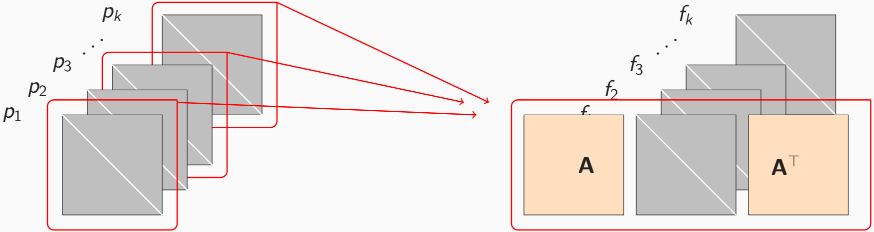


hQMLE:

- matrix codes of symmetric matrices

MCE, QMLE **and** ATFE **look very similar!**

Systems of polynomials in matrix representation:

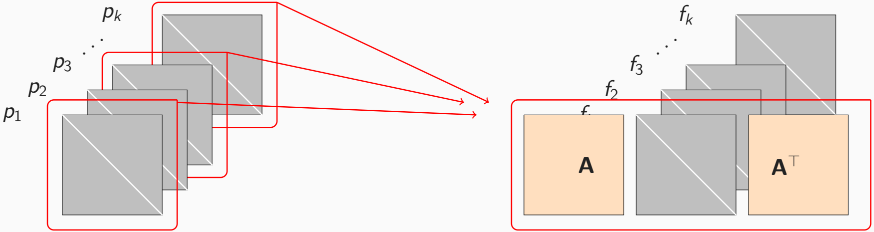


hQMLE:

- matrix codes of symmetric matrices
- isometry $(\mathbf{A}, \mathbf{A}^\top)$

MCE, QMLE and ATFE look very similar!

Systems of polynomials in matrix representation:

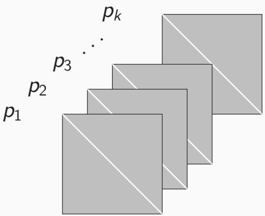


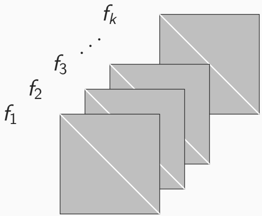
hQMLE:

- matrix codes of symmetric matrices
- isometry $(\mathbf{A}, \mathbf{A}^\top)$
- hQMLE can be seen as MCE, and vice versa

MCE, QMLE **and** ATFE **look very similar!**

Trilinear forms in matrix representation:





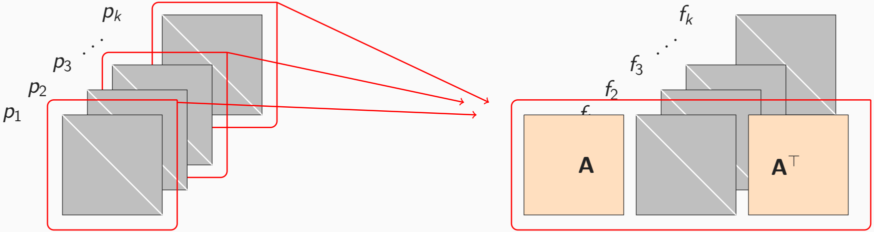
ATFE:

- matrix codes of skew symmetric matrices

We will come back to cryptanalysis of this problem!

MCE, QMLE and ATFE look very similar!

Trilinear forms in matrix representation:



ATFE:

- matrix codes of skew symmetric matrices
- isometry $(\mathbf{A}, \mathbf{A}^\top)$
- ATFE can be seen as MCE, and as hQMLE (and vice versa)

We will come back to cryptanalysis of this problem!

The basic protocol is not very eficient

$$
\begin{tikzcd}
\mathcal{O}_0 \arrow[rr, "\psi_0"] \arrow[d, dashed, "\phi"'] & & \mathcal{O}' \\
\mathcal{O}_1 \arrow[urr, "\psi_1"'] & &
\end{tikzcd}
$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
$\mathsf{com} \leftarrow \mathcal{O}'$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow_R \{0, 1\}$
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}' \stackrel{?}{=} \psi_{\mathsf{ch}}\ (\mathcal{O}_{\mathsf{ch}}\ )$

- **Challenge space is of size 2** $\Rightarrow$ Soundness error is $1/2$

The basic protocol is not very eficient

$$\begin{tikzcd}
\mathcal{O}_0 \arrow[r, "\psi_0"] \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\
\mathcal{O}_1 \arrow[ur, "\psi_1"'] &
\end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
$\mathsf{com} \leftarrow \mathcal{O}', \mathcal{O}'', \ldots, \mathcal{O}^{(r)}$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch} = (\mathsf{ch}_1, \ldots, \mathsf{ch}_r)}$	$\mathsf{ch} \leftarrow_R \{0,1\}^r$
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}_1}, \psi_{\mathsf{ch}_2}, \ldots, \psi_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}' \stackrel{?}{=} \psi_{\mathsf{ch}_1}(\mathcal{O}_{\mathsf{ch}_1})$
		$, \ldots, \mathcal{O}^{(r)} \stackrel{?}{=} \psi_{\mathsf{ch}_r}(\mathcal{O}_{\mathsf{ch}_r})$

- **Challenge space is of size 2** $\Rightarrow$ Soundness error is $1/2$
- For security of λ bits, **needs to be repeated $r = \lambda$ times!**

The basic protocol is not very eficient

$$
\begin{tikzcd}
\mathcal{O}_0 \arrow[r, "\psi_0"] \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\
\mathcal{O}_1 \arrow[ur, "\psi_1"'] &
\end{tikzcd}
$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
$\mathsf{com} \leftarrow \mathcal{O}', \mathcal{O}'', \ldots, \mathcal{O}^{(r)}$	$\xrightarrow{\mathsf{com}}$	
$\mathsf{ch} \leftarrow_R \{0,1\}^r$		
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}_1}, \psi_{\mathsf{ch}_2}, \ldots, \psi_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}' \stackrel{?}{=} \psi_{\mathsf{ch}_1}(\mathcal{O}_{\mathsf{ch}_1})$
		$, \ldots, \mathcal{O}^{(r)} \stackrel{?}{=} \psi_{\mathsf{ch}_r}(\mathcal{O}_{\mathsf{ch}_r})$

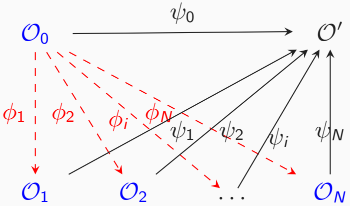
- **Challenge space is of size 2** $\Rightarrow$ Soundness error is $1/2$
- For security of λ bits, **needs to be repeated $r = \lambda$ times!**
- $\Rightarrow$ Signature contains λ isometries (from λ rounds)

The basic protocol is not very eficient

$$\begin{tikzcd} \mathcal{O}_0 \arrow[r, "\psi_0"] \arrow[d, dashed, "\phi"'] & \mathcal{O}' \\ \mathcal{O}_1 \arrow[ur, "\psi_1"'] & \end{tikzcd}$$

$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
com $\leftarrow \mathcal{O}', \mathcal{O}'', \ldots, \mathcal{O}^{(r)}$	$\xrightarrow{\text{com}}$	
ch $\leftarrow_R \{0,1\}^r$		
resp $\leftarrow \psi_{\mathsf{ch}_1}, \psi_{\mathsf{ch}_2}, \ldots, \psi_{\mathsf{ch}_r}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}' \stackrel{?}{=} \psi_{\mathsf{ch}_1}(\mathcal{O}_{\mathsf{ch}_1})$
		$, \ldots, \mathcal{O}^{(r)} \stackrel{?}{=} \psi_{\mathsf{ch}_r}(\mathcal{O}_{\mathsf{ch}_r})$

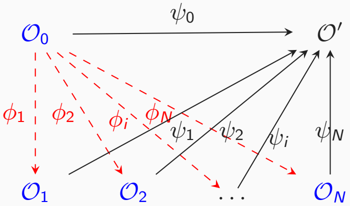
- **Challenge space is of size 2** $\Rightarrow$ Soundness error is $1/2$
- For security of λ bits, **needs to be repeated $r = \lambda$ times!**
- $\Rightarrow$ Signature contains λ isometries (from λ rounds)
- $\Rightarrow$ All operations in signing and verification need to be repeated λ times



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
com $\leftarrow \mathcal{O}'$	$\xrightarrow{\text{com}}$	
	$\xleftarrow{\text{ch}}$	ch $\leftarrow_R \{0, N-1\}$
resp $\leftarrow \psi_{\text{ch}}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\text{ch}_i}(\mathcal{O}_{\text{ch}_i})$

- **Challenge space is now of size** $N \Rightarrow$ Soundness error is $1/N$

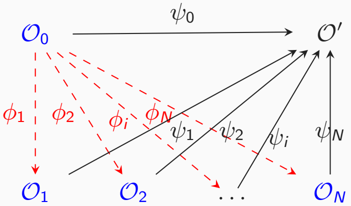
Optimization 1: Make the challenge space bigger (Multiple public keys)



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
$\mathsf{com} \leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch} = (\mathsf{ch}_1, \dots, \mathsf{ch}_r)}$	$\mathsf{ch} \leftarrow_R \{0, N-1\}^r$
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}_1}, \psi_{\mathsf{ch}_2}, \dots, \psi_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\mathsf{ch}_i}(\mathcal{O}_{\mathsf{ch}_i})$

- **Challenge space is now of size N** $\Rightarrow$ Soundness error is $1/N$
- For security of λ bits, **needs to be repeated $r = \frac{\lambda}{\log N}$ times!**

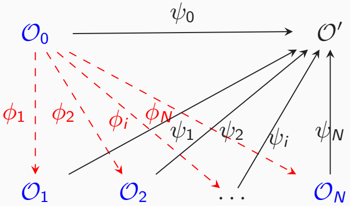
Optimization 1: Make the challenge space bigger (Multiple public keys)



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
$\mathsf{com} \leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\mathsf{com}}$	
$\mathsf{ch} \leftarrow_R \{0, N-1\}^r$		
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}_1}, \psi_{\mathsf{ch}_2}, \dots, \psi_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\mathsf{ch}_i}(\mathcal{O}_{\mathsf{ch}_i})$

- **Challenge space is now of size** N $\Rightarrow$ Soundness error is $1/N$
- For security of λ bits, **needs to be repeated** $r = \frac{\lambda}{\log N}$ **times!**
- $\Rightarrow$ Signature contains $\frac{\lambda}{\log N}$ isometries

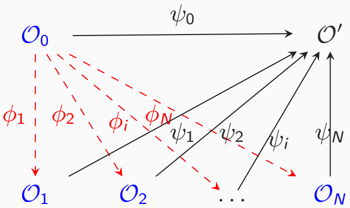
Optimization 1: Make the challenge space bigger (Multiple public keys)



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
$\mathsf{com} \leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\mathsf{com}}$	
$\mathsf{ch} \leftarrow_R \{0, N-1\}^r$		
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}_1}, \psi_{\mathsf{ch}_2}, \dots, \psi_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\mathsf{ch}_i}(\mathcal{O}_{\mathsf{ch}_i})$

- **Challenge space is now of size N** $\Rightarrow$ Soundness error is $1/N$
- For security of λ bits, **needs to be repeated $r = \frac{\lambda}{\log N}$ times!**
- $\Rightarrow$ Signature contains $\frac{\lambda}{\log N}$ isometries
- $\Rightarrow$ All operations in signing and verification need to be repeated $\frac{\lambda}{\log N}$ times

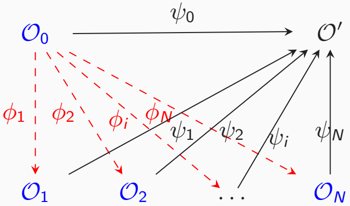
Optimization 1: Make the challenge space bigger (Multiple public keys)



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
com $\leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\text{com}}$	
ch $\leftarrow_R \{0, N-1\}^r$		
resp $\leftarrow \psi_{\text{ch}_1}, \psi_{\text{ch}_2}, \dots, \psi_{\text{ch}_r}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\text{ch}_i}(\mathcal{O}_{\text{ch}_i})$

- **Challenge space is now of size N** $\Rightarrow$ Soundness error is $1/N$
- For security of λ bits, **needs to be repeated $r = \frac{\lambda}{\log N}$ times!**
- $\Rightarrow$ Signature contains $\frac{\lambda}{\log N}$ isometries
- $\Rightarrow$ All operations in signing and verification need to be repeated $\frac{\lambda}{\log N}$ times
- **There is a cost - N-fold increase in public and private key**

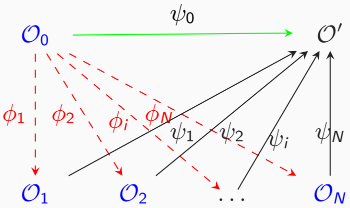
Optimization 1: Make the challenge space bigger (Multiple public keys)



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
com $\leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\text{com}}$	
ch $\leftarrow_R \{0, N-1\}^r$		
resp $\leftarrow \psi_{\text{ch}_1}, \psi_{\text{ch}_2}, \dots, \psi_{\text{ch}_r}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\text{ch}_i}(\mathcal{O}_{\text{ch}_i})$

- **Challenge space is now of size N** $\Rightarrow$ Soundness error is $1/N$
- For security of λ bits, **needs to be repeated $r = \frac{\lambda}{\log N}$ times!**
- $\Rightarrow$ Signature contains $\frac{\lambda}{\log N}$ isometries
- $\Rightarrow$ All operations in signing and verification need to be repeated $\frac{\lambda}{\log N}$ times
- **There is a cost - N-fold increase in public and private key**
- Always necessary to find the best trade-off

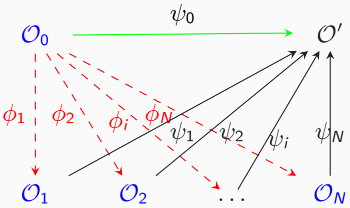
Optimization 2: Reduce signature size by using seeds



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
$\mathsf{com} \leftarrow \mathcal{O}'$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow_R \{0, N-1\}$
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\mathsf{ch}_i}(\mathcal{O}_{\mathsf{ch}_i})$

- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature

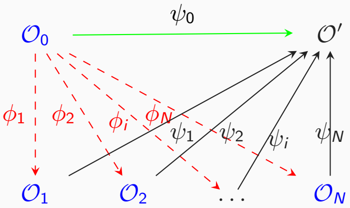
Optimization 2: Reduce signature size by using seeds



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
$\mathsf{com} \leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch} = (\mathsf{ch}_1, \dots, \mathsf{ch}_r)}$	$\mathsf{ch} \leftarrow_R \{0, N-1\}^r$
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}_1}, \psi_{\mathsf{ch}_2}, \dots, \psi_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\mathsf{ch}_i}(\mathcal{O}_{\mathsf{ch}_i})$

- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed

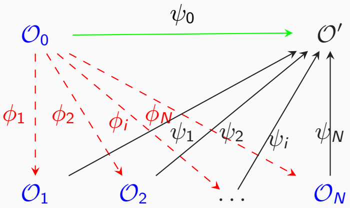
Optimization 2: Reduce signature size by using seeds



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
com $\leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\text{com}}$	
ch $\leftarrow_R \{0, N-1\}^r$		
resp $\leftarrow \psi_{\text{ch}_1}, \psi_{\text{ch}_2}, \dots, \psi_{\text{ch}_r}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\text{ch}_i}(\mathcal{O}_{\text{ch}_i})$

- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed
- **Problem:** This works only for ch $= 0$, and probability of choosing challenge 0 is $1/N$

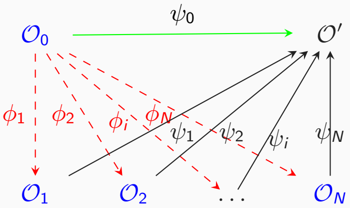
Optimization 2: Reduce signature size by using seeds



$\mathcal{P}(\mathcal{O}_0,\dots,\mathcal{O}_N,\phi_1,\dots,\phi_N)$		$\mathcal{V}(\mathcal{O}_0,\dots,\mathcal{O}_N)$
$\mathsf{com} \leftarrow \mathcal{O}',\mathcal{O}'',\dots,\mathcal{O}^{(r)}$	$\xrightarrow{\mathsf{com}}$	
$\mathsf{ch} \leftarrow_R \{0, N-1\}^r$		
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}_1},\psi_{\mathsf{ch}_2},\dots,\psi_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\mathsf{ch}_i}(\mathcal{O}_{\mathsf{ch}_i})$

- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed
- **Problem:** This works only for $\mathsf{ch} = 0$, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general

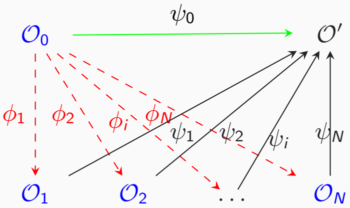
Optimization 2: Reduce signature size by using seeds



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
com $\leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\text{com}}$	
ch $\leftarrow_R \{0, N-1\}^r$		
resp $\leftarrow \psi_{\text{ch}_1}, \psi_{\text{ch}_2}, \dots, \psi_{\text{ch}_r}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\text{ch}_i}(\mathcal{O}_{\text{ch}_i})$

- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed
- **Problem:** This works only for ch $= 0$, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
  - $\Rightarrow$ **signature is not of constant size**

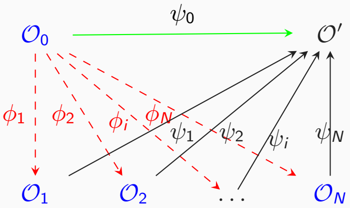
Optimization 2: Reduce signature size by using seeds



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
com $\leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\text{com}}$	
ch $\leftarrow_R \{0, N-1\}^r$		
resp $\leftarrow \psi_{\text{ch}_1}, \psi_{\text{ch}_2}, \dots, \psi_{\text{ch}_r}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\text{ch}_i}(\mathcal{O}_{\text{ch}_i})$

- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed
- **Problem:** This works only for ch $= 0$, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
  - $\Rightarrow$ **signature is not of constant size**
- **Idea:** Always have a fixed number M of 0 challenges

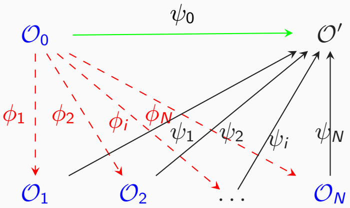
Optimization 2: Reduce signature size by using seeds



$\mathcal{P}(\mathcal{O}_0, \ldots, \mathcal{O}_N, \phi_1, \ldots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \ldots, \mathcal{O}_N)$
com $\leftarrow \mathcal{O}', \mathcal{O}'', \ldots, \mathcal{O}^{(r)}$	$\xrightarrow{\text{com}}$	
ch $\leftarrow_R \{0, N-1\}^r$		
resp $\leftarrow \psi_{\text{ch}_1}, \psi_{\text{ch}_2}, \ldots, \psi_{\text{ch}_r}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\text{ch}_i}(\mathcal{O}_{\text{ch}_i})$

- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed
- **Problem:** This works only for ch $= 0$, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
  - $\Rightarrow$ **signature is not of constant size**
- **Idea:** Always have a fixed number M of 0 challenges
  - **We need a special hash function** that always produces fixed weight outputs

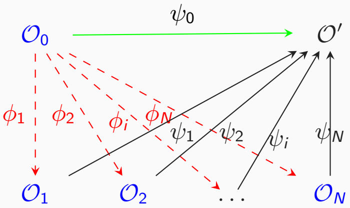
Optimization 2: Reduce signature size by using seeds



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
$\mathrm{com} \leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\mathrm{com}}$	
$\mathrm{ch} \leftarrow_R \{0, N-1\}^r$		
$\mathrm{resp} \leftarrow \psi_{\mathrm{ch}_1}, \psi_{\mathrm{ch}_2}, \dots, \psi_{\mathrm{ch}_r}$	$\xrightarrow{\mathrm{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\mathrm{ch}_i}(\mathcal{O}_{\mathrm{ch}_i})$

- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed
- **Problem:** This works only for ch $= 0$, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
  - $\Rightarrow$ **signature is not of constant size**
- **Idea:** Always have a fixed number M of 0 challenges
  - **We need a special hash function** that always produces fixed weight outputs
  - Always necessary to find the best trade-off

Optimization 2: Reduce signature size by using seeds



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
$\mathsf{com} \leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\mathsf{com}}$	
$\mathsf{ch} \leftarrow_R \{0, N-1\}^r$		
$\mathsf{resp} \leftarrow \psi_{\mathsf{ch}_1}, \psi_{\mathsf{ch}_2}, \dots, \psi_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\mathsf{ch}_i}(\mathcal{O}_{\mathsf{ch}_i})$

- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed
- **Problem:** This works only for $\mathsf{ch} = 0$, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
  - $\Rightarrow$ **signature is not of constant size**
- **Idea:** Always have a fixed number M of 0 challenges
  - **We need a special hash function** that always produces fixed weight outputs
  - Always necessary to find the best trade-off

So how better are these schemes in terms of performance?

For example, MEDS:

MEDS	q	n ($\approx m$)	r	pk (bytes)	Signature (bytes)
level 1	4093	25	144	21595	5456
level 2	4093	34	208	55520	10786
level 3	4093	44	272	122000	21052

Important to note:

- QMLE/MCE still not so well understood
- Not NP-hard but likely still hard
- Secure practical parameters significantly changed in the last few years, as our understanding improved

Solving matrix code equivalence problems

Solving matrix code equivalence problems

Several approaches:

- Graph-based techniques

Solving matrix code equivalence problems

Several approaches:

- Graph-based techniques
- Algebraic models

Solving matrix code equivalence problems

Several approaches:

- Graph-based techniques
- Algebraic models
- Leon-like approach (graph-based+algebraic)

Graph-based algorithm for QMLE

- [Faugére, Peret, 2006] – inhomogenous version is easy ($\mathcal{O}(n^9)$ heuristically)

Graph-based algorithm for QMLE

- [Faugére, Peret, 2006] – inhomogenous version is easy ($\mathcal{O}(n^9)$ heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph based approach** $\mathcal{O}(q^{\frac{2}{3}n}n^9)$

Graph-based algorithm for QMLE

- [Faugére, Peret, 2006] – inhomogenous version is easy ($\mathcal{O}(n^9)$ heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph based approach** $\mathcal{O}(q^{\frac{2}{3}n}n^9)$
  - Crucial observation: One collision point $\mathbf{b} = \mathbf{aS}$ turns a **hard homogenous instance** into easy inhomogenous one

Graph-based algorithm for QMLE

- [Faugére, Peret, 2006] – inhomogenous version is easy ($\mathcal{O}(n^9)$ heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph based approach** $\mathcal{O}(q^{\frac{2}{3}n}n^9)$
  - Crucial observation: One collision point $\mathbf{b} = \mathbf{aS}$ turns a **hard homogenous instance** into easy inhomogenous one

    if $\mathbf{b} = \mathbf{aS}$, then $\mathcal{P}(\mathbf{x} + \mathbf{a}) = \mathcal{F}(\mathbf{xS} + \mathbf{b})\mathbf{T}$.

Graph-based algorithm for QMLE

- [Faugére, Peret, 2006] – inhomogenous version is easy ($\mathcal{O}(n^9)$ heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph based approach** $\mathcal{O}(q^{\frac{2}{3}n}n^9)$
  - Crucial observation: One collision point $\mathbf{b} = \mathbf{aS}$ turns a **hard homogenous instance** into easy inhomogenous one

    if $\mathbf{b} = \mathbf{aS}$, then $\mathcal{P}(\mathbf{x} + \mathbf{a}) = \mathcal{F}(\mathbf{xS} + \mathbf{b})\mathbf{T}$.

    $\Rightarrow \mathcal{P}'(\mathbf{x}) = \mathcal{P}(\mathbf{x} + \mathbf{a}), \mathcal{F}'(\mathbf{x}) = \mathcal{F}(\mathbf{x} + \mathbf{b})$ **is easy instance!**

Graph-based algorithm for QMLE

- [Faugére, Peret, 2006] – inhomogenous version is easy ($\mathcal{O}(n^9)$ heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph based approach** $\mathcal{O}(q^{\frac{2}{3}n}n^9)$
  - Crucial observation: One collision point $\mathbf{b} = \mathbf{aS}$ turns a **hard homogenous instance** into easy inhomogenous one

    if $\mathbf{b} = \mathbf{aS}$, then $\mathcal{P}(\mathbf{x}+\mathbf{a}) = \mathcal{F}(\mathbf{xS}+\mathbf{b})\mathbf{T}$.

    $\Rightarrow \mathcal{P}'(\mathbf{x}) = \mathcal{P}(\mathbf{x}+\mathbf{a}), \mathcal{F}'(\mathbf{x}) = \mathcal{F}(\mathbf{x}+\mathbf{b})$ **is easy instance!**
  - **Problem is reduced to finding the collision**

Algebraic modelling of MCE - the straightforward way

For $(\mathbf{C}^{(1)}, \dots, \mathbf{C}^{(k)})$ and $(\mathbf{D}^{(1)}, \dots, \mathbf{D}^{(k)})$ bases of $\mathcal{C}$ and $\mathcal{D}$, find invertible $\mathbf{A}, \mathbf{B}$ and $\mathbf{T} = (t_{ij})$ s.t.:

$$\sum_{1 \leqslant s \leqslant k} t_{rs} \mathbf{D}^{(s)} = \mathbf{A}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, 1 \leqslant r \leqslant k$$

Algebraic modelling of MCE - the straightforward way

For $(\mathbf{C}^{(1)}, \dots, \mathbf{C}^{(k)})$ and $(\mathbf{D}^{(1)}, \dots, \mathbf{D}^{(k)})$ bases of $\mathcal{C}$ and $\mathcal{D}$, find invertible $\mathbf{A}, \mathbf{B}$ and $\mathbf{T} = (t_{ij})$ s.t.:

$$\sum_{1 \leqslant s \leqslant k} t_{rs} \mathbf{D}^{(s)} = \mathbf{A}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, 1 \leqslant r \leqslant k$$

- **Directly:** bilinear system of knm eqns in $m^2 + n^2 + k^2$ vars (bilinear $a_{ij} b_{i'j'}$, linear t_{rs})

Algebraic modelling of MCE - the straightforward way

For $(\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(k)})$ and $(\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(k)})$ bases of $\mathcal{C}$ and $\mathcal{D}$, find invertible $\mathbf{A}, \mathbf{B}$ and $\mathbf{T} = (t_{ij})$ s.t.:

$$\sum_{1 \leqslant s \leqslant k} t_{rs}\mathbf{D}^{(s)} = \mathbf{A}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, 1 \leqslant r \leqslant k$$

- **Directly:** bilinear system of knm eqns in $m^2 + n^2 + k^2$ vars (bilinear $a_{ij}b_{i'j'}$, linear t_{rs})
  - Security estimates too high

Algebraic modelling of MCE - the straightforward way

For $(\mathbf{C}^{(1)}, \dots, \mathbf{C}^{(k)})$ and $(\mathbf{D}^{(1)}, \dots, \mathbf{D}^{(k)})$ bases of $\mathcal{C}$ and $\mathcal{D}$, find invertible $\mathbf{A}, \mathbf{B}$ and $\mathbf{T} = (t_{ij})$ s.t.:

$$\sum_{1 \leqslant s \leqslant k} t_{rs} \mathbf{D}^{(s)} = \mathbf{A}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, 1 \leqslant r \leqslant k$$

- **Directly:** bilinear system of knm eqns in $m^2 + n^2 + k^2$ vars (bilinear $a_{ij} b_{i'j'}$, linear t_{rs})
  - Security estimates too high
- Coefs of $\mathbf{A}_{i\_}$ and $\mathbf{A}_{j\_}$ ($\mathbf{B}_{\_i}$ and $\mathbf{B}_{\_j}$) don't appear in same equation

Algebraic modelling of MCE - the straightforward way

For $(\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(k)})$ and $(\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(k)})$ bases of $\mathcal{C}$ and $\mathcal{D}$, find invertible $\mathbf{A}, \mathbf{B}$ and $\mathbf{T} = (t_{ij})$ s.t.:

$$\sum_{1 \leqslant s \leqslant k} t_{rs} \mathbf{D}^{(s)} = \mathbf{A}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, 1 \leqslant r \leqslant k$$

- **Directly:** bilinear system of knm eqns in $m^2 + n^2 + k^2$ vars (bilinear $a_{ij} b_{i'j'}$, linear t_{rs})
  - Security estimates too high
- Coefs of $\mathbf{A}_{i\_}$ and $\mathbf{A}_{j\_}$ ($\mathbf{B}_{\_i}$ and $\mathbf{B}_{\_j}$) don't appear in same equation
- **Consider only a small number α of rows of A**, i.e.,

$$\sum_{1 \leqslant s \leqslant k} t_{rs} \mathbf{D}^{(s)}_{i\_} = \mathbf{A}_{i\_}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, i, \ 1 \leqslant r \leqslant k, \ 1 \leqslant i \leqslant \alpha.$$

Algebraic modelling of MCE - the straightforward way

For $(\mathbf{C}^{(1)}, \dots, \mathbf{C}^{(k)})$ and $(\mathbf{D}^{(1)}, \dots, \mathbf{D}^{(k)})$ bases of $\mathcal{C}$ and $\mathcal{D}$, find invertible $\mathbf{A}, \mathbf{B}$ and $\mathbf{T} = (t_{ij})$ s.t.:

$$\sum_{1 \leqslant s \leqslant k} t_{rs} \mathbf{D}^{(s)} = \mathbf{A}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, 1 \leqslant r \leqslant k$$

- **Directly:** bilinear system of knm eqns in $m^2 + n^2 + k^2$ vars (bilinear $a_{ij}b_{i'j'}$, linear t_{rs})
  - Security estimates too high
- Coefs of $\mathbf{A}_{i\_}$ and $\mathbf{A}_{j\_}$ ($\mathbf{B}_{\_i}$ and $\mathbf{B}_{\_j}$) don't appear in same equation
- **Consider only a small number α of rows of $\mathbf{A}$**, i.e.,

$$\sum_{1 \leqslant s \leqslant k} t_{rs} \mathbf{D}_{i\_}^{(s)} = \mathbf{A}_{i\_}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, i, \ 1 \leqslant r \leqslant k, \ 1 \leqslant i \leqslant \alpha.$$

- Guess αm coefs of $\mathbf{A}_{i\_}$, and solve linear system of αkn equations in $n^2 + k^2$ variables

Algebraic modelling of MCE - the straightforward way

For $(\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(k)})$ and $(\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(k)})$ bases of $\mathcal{C}$ and $\mathcal{D}$, find invertible $\mathbf{A}, \mathbf{B}$ and $\mathbf{T} = (t_{ij})$ s.t.:

$$\sum_{1 \leqslant s \leqslant k} t_{rs} \mathbf{D}^{(s)} = \mathbf{A}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, 1 \leqslant r \leqslant k$$

- **Directly:** bilinear system of knm eqns in $m^2 + n^2 + k^2$ vars (bilinear $a_{ij} b_{i'j'}$, linear t_{rs})
  - Security estimates too high
- Coefs of $\mathbf{A}_{i\_}$ and $\mathbf{A}_{j\_}$ ($\mathbf{B}_{\_i}$ and $\mathbf{B}_{\_j}$) don't appear in same equation
- **Consider only a small number α of rows of A**, i.e.,

$$\sum_{1 \leqslant s \leqslant k} t_{rs} \mathbf{D}^{(s)}_{i\_} = \mathbf{A}_{i\_}\mathbf{C}^{(r)}\mathbf{B}, \quad \forall r, i, \; 1 \leqslant r \leqslant k, \; 1 \leqslant i \leqslant \alpha.$$

- Guess αm coefs of $\mathbf{A}_{i\_}$, and solve linear system of αkn equations in $n^2 + k^2$ variables
- For $m = n = k$, $\alpha = 2$ is enough $\rightarrow$ complexity is $\mathcal{O}(q^{2n} n^6)$

Improved Algebraic modelling of MCE

Make use of coding theory!

Improved Algebraic modelling of MCE

Make use of coding theory!

- $\mathbf{G}$ and $\mathbf{G}'$ – the $k \times mn$ generator matrices of $\mathcal{C}$ and $\mathcal{D}$

Improved Algebraic modelling of MCE

Make use of coding theory!

- $\mathbf{G}$ and $\mathbf{G}'$ – the $k \times mn$ generator matrices of $\mathcal{C}$ and $\mathcal{D}$
  - "Flatten" the matrices into vectors

Improved Algebraic modelling of MCE

Make use of coding theory!

- $\mathbf{G}$ and $\mathbf{G}'$ – the $k \times mn$ generator matrices of $\mathcal{C}$ and $\mathcal{D}$
  - “Flatten” the matrices into vectors
- $\mathbf{G}'^{\perp}$ - the generator matrix of the dual code of $\mathcal{D}$

Improved Algebraic modelling of MCE

Make use of coding theory!

- $\mathbf{G}$ and $\mathbf{G}'$ – the $k \times mn$ generator matrices of $\mathcal{C}$ and $\mathcal{D}$
  - "Flatten" the matrices into vectors
- $\mathbf{G}'^{\perp}$ - the generator matrix of the dual code of $\mathcal{D}$
- Take $\tilde{\mathbf{G}}(\mathbf{A}, \mathbf{B})$ - generator matrix of $\mathbf{A}\mathcal{C}\mathbf{B}$ (i.e. $\mathcal{D}$) for $\mathbf{A}$ and $\mathbf{B}$ with unknown coefficients

Improved Algebraic modelling of MCE

Make use of coding theory!

- $\mathbf{G}$ and $\mathbf{G}'$ – the $k \times mn$ generator matrices of $\mathcal{C}$ and $\mathcal{D}$
  - "Flatten" the matrices into vectors
- $\mathbf{G}'^{\perp}$ - the generator matrix of the dual code of $\mathcal{D}$
- Take $\tilde{\mathbf{G}}(\mathbf{A}, \mathbf{B})$ - generator matrix of $\mathbf{A}\mathcal{C}\mathbf{B}$ (i.e. $\mathcal{D}$) for $\mathbf{A}$ and $\mathbf{B}$ with unknown coefficients
- Construct the system:

$$\mathbf{G}'^{\perp} \cdot \tilde{\mathbf{G}}(\mathbf{A}, \mathbf{B})^{\top} = \mathbf{0},$$

of $(mn - k)k$ bilinear equations **in only $m^2 + n^2$ variables**

Improved Algebraic modelling of MCE

Make use of coding theory!

- $\mathbf{G}$ and $\mathbf{G}'$ – the $k \times mn$ generator matrices of $\mathcal{C}$ and $\mathcal{D}$
  - "Flatten" the matrices into vectors
- $\mathbf{G}'^{\perp}$ - the generator matrix of the dual code of $\mathcal{D}$
- Take $\tilde{\mathbf{G}}(\mathbf{A}, \mathbf{B})$ - generator matrix of $\mathbf{A}\mathcal{C}\mathbf{B}$ (i.e. $\mathcal{D}$) for $\mathbf{A}$ and $\mathbf{B}$ with unknown coefficients
- Construct the system:

$$\mathbf{G}'^{\perp} \cdot \tilde{\mathbf{G}}(\mathbf{A}, \mathbf{B})^{\top} = \mathbf{0},$$

  of $(mn - k)k$ bilinear equations **in only $m^2 + n^2$ variables**
  - **we got rid of t^2 variables!**
- Solve system with Bilinear XL for example

Improved Algebraic modelling of MCE

Make use of coding theory!

- $\mathbf{G}$ and $\mathbf{G}'$ – the $k \times mn$ generator matrices of $\mathcal{C}$ and $\mathcal{D}$
  - "Flatten" the matrices into vectors
- $\mathbf{G}'^{\perp}$ - the generator matrix of the dual code of $\mathcal{D}$
- Take $\tilde{\mathbf{G}}(\mathbf{A}, \mathbf{B})$ - generator matrix of $\mathbf{A}\mathcal{C}\mathbf{B}$ (i.e. $\mathcal{D}$) for $\mathbf{A}$ and $\mathbf{B}$ with unknown coefficients
- Construct the system:

$$\mathbf{G}'^{\perp} \cdot \tilde{\mathbf{G}}(\mathbf{A}, \mathbf{B})^{\top} = \mathbf{0},$$

  of $(mn - k)k$ bilinear equations **in only $m^2 + n^2$ variables**
  - **we got rid of t^2 variables!**
- Solve system with Bilinear XL for example
- Dimension of code – crucial for complexity
  - smallest for $k = mn/2$, and grows as k reduces or grows

Further improvments of the algebraic modelling of MCE

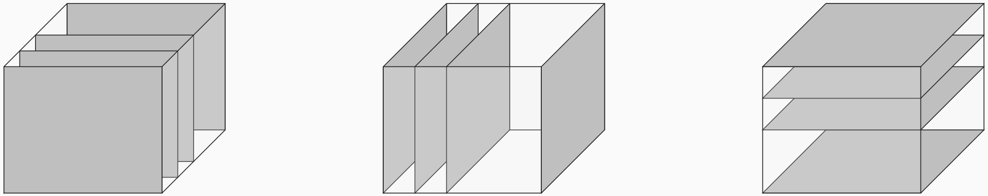
Trilinear form view:

$$\mathcal{C}(\mathbf{Ax}, \mathbf{By}, \mathbf{Tz}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

Further improvments of the algebraic modelling of MCE

Trilinear form view:

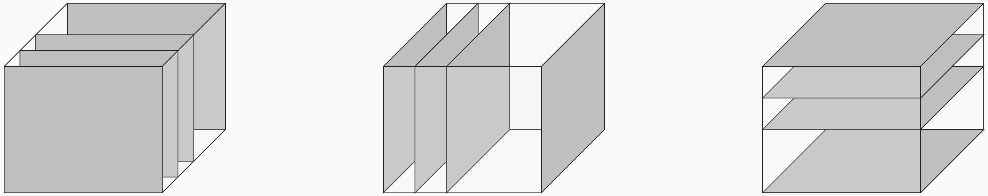
$$\mathcal{C}(\mathbf{Ax}, \mathbf{By}, \mathbf{Tz}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{z})$$



Further improvments of the algebraic modelling of MCE

Trilinear form view:

$$\mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{T}\mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

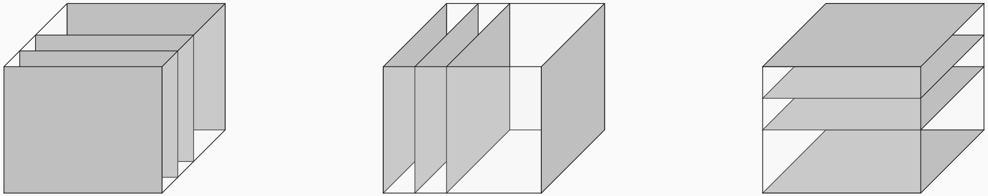


$\mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{T}^{-1}\mathbf{z})$ $\quad$ $\mathcal{C}(\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{T}\mathbf{z}) = \mathcal{D}(\mathbf{A}^{-1}\mathbf{x}, \mathbf{y}, \mathbf{z})$ $\quad$ $\mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{y}, \mathbf{T}\mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{B}^{-1}\mathbf{y}, \mathbf{z})$

Further improvments of the algebraic modelling of MCE

Trilinear form view:

$$\mathcal{C}(\mathbf{Ax}, \mathbf{By}, \mathbf{Tz}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{z})$$



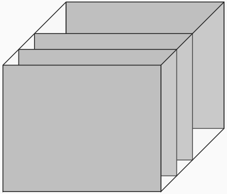
$\mathcal{C}(\mathbf{Ax}, \mathbf{By}, \mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{T}^{-1}\mathbf{z})$ $\quad$ $\mathcal{C}(\mathbf{x}, \mathbf{By}, \mathbf{Tz}) = \mathcal{D}(\mathbf{A}^{-1}\mathbf{x}, \mathbf{y}, \mathbf{z})$ $\quad$ $\mathcal{C}(\mathbf{Ax}, \mathbf{y}, \mathbf{Tz}) = \mathcal{D}(\mathbf{x}, \mathbf{B}^{-1}\mathbf{y}, \mathbf{z})$

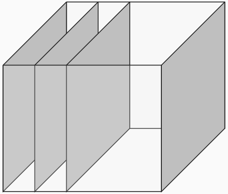
$\mathbf{G}_{\mathbf{z}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{z}}^{\top}(\mathbf{A}, \mathbf{B}) = \mathbf{0},$

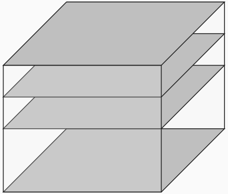
Further improvments of the algebraic modelling of MCE

Trilinear form view:

$$\mathcal{C}(\mathbf{Ax}, \mathbf{By}, \mathbf{Tz}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{z})$$







$$\mathcal{C}(\mathbf{Ax}, \mathbf{By}, \mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{T}^{-1}\mathbf{z})$$

$$\mathbf{G}_{\mathbf{z}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{z}}^{\top}(\mathbf{A}, \mathbf{B}) = \mathbf{0},$$

$$\mathcal{C}(\mathbf{x}, \mathbf{By}, \mathbf{Tz}) = \mathcal{D}(\mathbf{A}^{-1}\mathbf{x}, \mathbf{y}, \mathbf{z})$$

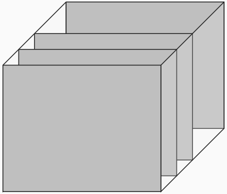
$$\mathbf{G}_{\mathbf{x}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{x}}^{\top}(\mathbf{B}, \mathbf{T}) = \mathbf{0},$$

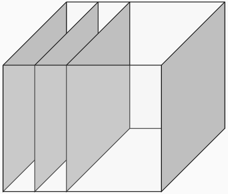
$$\mathcal{C}(\mathbf{Ax}, \mathbf{y}, \mathbf{Tz}) = \mathcal{D}(\mathbf{x}, \mathbf{B}^{-1}\mathbf{y}, \mathbf{z})$$

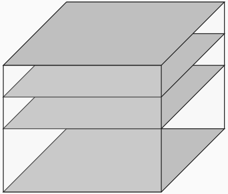
Further improvments of the algebraic modelling of MCE

Trilinear form view:

$$\mathcal{C}(\mathbf{Ax}, \mathbf{By}, \mathbf{Tz}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{z})$$







$$\mathcal{C}(\mathbf{Ax}, \mathbf{By}, \mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{T}^{-1}\mathbf{z})$$

$$\mathbf{G}_{\mathbf{z}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{z}}^{\top}(\mathbf{A}, \mathbf{B}) = \mathbf{0},$$

$$\mathcal{C}(\mathbf{x}, \mathbf{By}, \mathbf{Tz}) = \mathcal{D}(\mathbf{A}^{-1}\mathbf{x}, \mathbf{y}, \mathbf{z})$$

$$\mathbf{G}_{\mathbf{x}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{x}}^{\top}(\mathbf{B}, \mathbf{T}) = \mathbf{0},$$

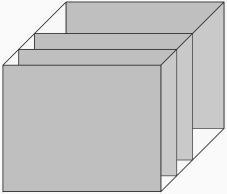
$$\mathcal{C}(\mathbf{Ax}, \mathbf{y}, \mathbf{Tz}) = \mathcal{D}(\mathbf{x}, \mathbf{B}^{-1}\mathbf{y}, \mathbf{z})$$

$$\mathbf{G}_{\mathbf{y}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{y}}^{\top}(\mathbf{A}, \mathbf{T}) = \mathbf{0},$$

Further improvments of the algebraic modelling of MCE

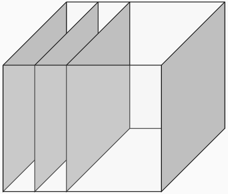
Trilinear form view:

$$\mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{T}\mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{z})$$



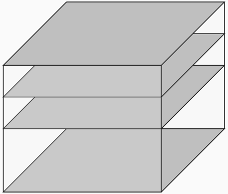
$$\mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{T}^{-1}\mathbf{z})$$

$$\mathbf{G}'^{\perp}_{\mathbf{z}} \cdot \tilde{\mathbf{G}}_{\mathbf{z}}^{\top}(\mathbf{A}, \mathbf{B}) = \mathbf{0},$$



$$\mathcal{C}(\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{T}\mathbf{z}) = \mathcal{D}(\mathbf{A}^{-1}\mathbf{x}, \mathbf{y}, \mathbf{z})$$

$$\mathbf{G}'^{\perp}_{\mathbf{x}} \cdot \tilde{\mathbf{G}}_{\mathbf{x}}^{\top}(\mathbf{B}, \mathbf{T}) = \mathbf{0},$$



$$\mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{y}, \mathbf{T}\mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{B}^{-1}\mathbf{y}, \mathbf{z})$$

$$\mathbf{G}'^{\perp}_{\mathbf{y}} \cdot \tilde{\mathbf{G}}_{\mathbf{y}}^{\top}(\mathbf{A}, \mathbf{T}) = \mathbf{0},$$

Further improvments of the algebraic modelling of MCE

$$\mathbf{G}_{\mathbf{z}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{z}}^{\top}(\mathbf{A}, \mathbf{B}) = \mathbf{0}, \tag{1}$$

$$\mathbf{G}_{\mathbf{x}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{x}}^{\top}(\mathbf{B}, \mathbf{T}) = \mathbf{0}, \tag{2}$$

$$\mathbf{G}_{\mathbf{y}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{y}}^{\top}(\mathbf{A}, \mathbf{T}) = \mathbf{0}. \tag{3}$$

- A total of $(mn-k)k + (nk-m)m + (mk-n)n$ equations in $n^2 + m^2 + k^2$ variables

Further improvments of the algebraic modelling of MCE

$$\mathbf{G}_{\mathbf{z}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{z}}^{\top}(\mathbf{A}, \mathbf{B}) = \mathbf{0}, \tag{1}$$
$$\mathbf{G}_{\mathbf{x}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{x}}^{\top}(\mathbf{B}, \mathbf{T}) = \mathbf{0}, \tag{2}$$
$$\mathbf{G}_{\mathbf{y}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{y}}^{\top}(\mathbf{A}, \mathbf{T}) = \mathbf{0}. \tag{3}$$

- A total of $(mn-k)k + (nk-m)m + (mk-n)n$ equations in $n^2 + m^2 + k^2$ variables
- **The system is not semi-regular though!**
  - Syzygies in degree 3 from linear combination of types $t_{ij} \cdot (1)$, $a_{ij} \cdot (2)$, $b_{ij} \cdot (3)$

Further improvments of the algebraic modelling of MCE

$$\mathbf{G}_{\mathbf{z}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{z}}^{\top}(\mathbf{A}, \mathbf{B}) = \mathbf{0}, \tag{1}$$

$$\mathbf{G}_{\mathbf{x}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{x}}^{\top}(\mathbf{B}, \mathbf{T}) = \mathbf{0}, \tag{2}$$

$$\mathbf{G}_{\mathbf{y}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{y}}^{\top}(\mathbf{A}, \mathbf{T}) = \mathbf{0}. \tag{3}$$

- A total of $(mn-k)k + (nk-m)m + (mk-n)n$ equations in $n^2 + m^2 + k^2$ variables
- **The system is not semi-regular though!**
  - Syzygies in degree 3 from linear combination of types $t_{ij} \cdot (1)$, $a_{ij} \cdot (2)$, $b_{ij} \cdot (3)$
- We can estimate the Hilbert series and complexity for solving using Bilinear XL

Further improvments of the algebraic modelling of MCE

$$\mathbf{G}_{\mathbf{z}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{z}}^{\top}(\mathbf{A}, \mathbf{B}) = \mathbf{0}, \qquad (1)$$

$$\mathbf{G}_{\mathbf{x}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{x}}^{\top}(\mathbf{B}, \mathbf{T}) = \mathbf{0}, \qquad (2)$$

$$\mathbf{G}_{\mathbf{y}}'^{\perp} \cdot \tilde{\mathbf{G}}_{\mathbf{y}}^{\top}(\mathbf{A}, \mathbf{T}) = \mathbf{0}. \qquad (3)$$

- A total of $(mn-k)k + (nk-m)m + (mk-n)n$ equations in $n^2 + m^2 + k^2$ variables
- **The system is not semi-regular though!**
  - Syzygies in degree 3 from linear combination of types $t_{ij} \cdot (1)$, $a_{ij} \cdot (2)$, $b_{ij} \cdot (3)$
- We can estimate the Hilbert series and complexity for solving using Bilinear XL

$n = m = k$	plain	improved
14	169	148
22	255	218
30	349	299

Summary of the course

- **Monday - Designs**
  - General
  - Classic designs
- **Tuesday - Design and general MQ solving techniques**
  - Key size optimization techniques
  - Algorithms for solving the MQ problem
- **Wednesday - Cryptanalysis**
  - MinRank
  - Equivalent keys attacks
- **Thursday - Cryptanalysis**
  - Attacks on UOV
- **Friday - Provably secure designs**
  - Fiat-Shamir signatures MQDSS, SOFIA, MEDS

Thank you for listening!

And attending this course!