Digital signatures from equivalence problems - A closer look at MEDS and ALTEQ

Simona Samardjiska Radboud University, Netherlands
Youming Qiao University of Technology Sydney, Australia

NIST PQC Seminar

Acknowledgement

- The rest of the MEDS team: Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Tovohery Hajatiana Randrianarisoa, Lars Ran, Krijn Reijnders, Monika Trimoska
- The rest of the ALTEQ team: Markus Bläser, Dung Hoang Duong, Anand Kumar Narayanan, Thomas Plantard, Arnaud Sipasseuth, Gang Tang.

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. $\mathcal{O}_1 = \phi(\mathcal{O}_0)$

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 case - when problem is **hard**! What can we do with it?

Motivation

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 case - when problem is **hard**! What can we do with it? Turns out - a lot!

Motivation

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 case - when problem is **hard**! What can we do with it? Turns out - a lot!

- **Zero-Knowledge protocols**

Motivation

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 case - when problem is **hard**! What can we do with it? Turns out - a lot!

- **Zero-Knowledge protocols**
- **Identification schemes (IDS)**

Motivation

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 case - when problem is **hard**! What can we do with it? Turns out - a lot!

- **Zero-Knowledge protocols**
- **Identification schemes (IDS)**
- **Digital Signatures via Fiat-Shamir transform**
  - F-S is a common strategy for PQ signatures
    - Dilithium, MQDSS, Picnic in first 3 rounds of NIST competition

Motivation

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 case - when problem is **hard**! What can we do with it? Turns out - a lot!

- **Zero-Knowledge protocols**
- **Identification schemes (IDS)**
- **Digital Signatures via Fiat-Shamir transform**
    - F-S is a common strategy for PQ signatures
        - Dilithium, MQDSS, Picnic in first 3 rounds of NIST competition
        - More than 15 in the additional round!

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

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

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

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

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

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

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

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

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

[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 $\leftarrow \mathcal{O}'$	$\xrightarrow{\text{com}}$	

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

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

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

[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

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

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

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

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

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

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

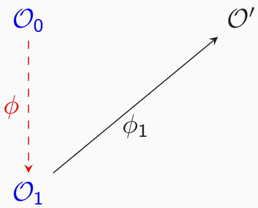
$\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi)$		$\mathcal{V}(\mathcal{O}_0, \mathcal{O}_1)$
com $\leftarrow \mathcal{O}'$	$\xrightarrow{\text{com}}$	
	$\xleftarrow{\text{ch}}$	ch $\leftarrow_R \{0, 1\}$
resp $\leftarrow \phi_{\text{ch}}$	$\xrightarrow{\text{resp}}$	

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



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

[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

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

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

Digital Signatures via the Fiat-Shamir transform

IDS

$$
\begin{array}{lcl}
\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi) & & \mathcal{V}(\mathcal{O}_0, \mathcal{O}_1) \\
\hline
\mathsf{com} \leftarrow \mathcal{O}' & & \\
 & \xrightarrow{\quad \mathsf{com} \quad} & \\
 & & \mathsf{ch} \leftarrow_R \{0,1\} \\
 & \xleftarrow{\quad \mathsf{ch} \quad} & \\
\mathsf{resp} \leftarrow \phi_{\mathsf{ch}} & & \\
 & \xrightarrow{\quad \mathsf{resp} \quad} & \\
 & & \mathcal{O}' \stackrel{?}{=} \phi_{\mathsf{ch}}\ (\mathcal{O}_{\mathsf{ch}}\ )
\end{array}
$$

Digital Signatures via the Fiat-Shamir transform

IDS

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

Digital Signatures via the Fiat-Shamir transform

IDS

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

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

FS signature

Signer(pk, sk)

$\mathsf{com} \leftarrow (\mathcal{O}', \mathcal{O}'', \ldots, \mathcal{O}^{(r)})$

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

$\mathsf{resp} \leftarrow (\phi_{\mathsf{ch}_1}, \phi_{\mathsf{ch}_2}, \ldots, \phi_{\mathsf{ch}_r})$

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

Verifier(pk)

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

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

output : b

The basic protocol is not very eficient

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

$$
\begin{array}{lcl}
\mathcal{P}(\mathcal{O}_0, \mathcal{O}_1, \phi) & & \mathcal{V}(\mathcal{O}_0, \mathcal{O}_1) \\
\hline
\mathsf{com} \leftarrow \mathcal{O}' & & \\
 & \xrightarrow{\quad \mathsf{com} \quad} & \\
 & & \mathsf{ch} \leftarrow_R \{0, 1\} \\
 & \xleftarrow{\quad \mathsf{ch} \quad} & \\
\mathsf{resp} \leftarrow \psi_{\mathsf{ch}} & & \\
 & \xrightarrow{\quad \mathsf{resp} \quad} & \\
 & & \mathcal{O}' \stackrel{?}{=} \psi_{\mathsf{ch}} \left(\mathcal{O}_{\mathsf{ch}} \right)
\end{array}
$$

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

The basic protocol is not very eficient

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

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

- **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)

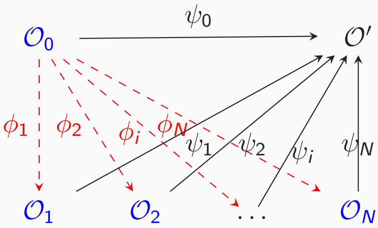
The basic protocol is not very eficient

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

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

- **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

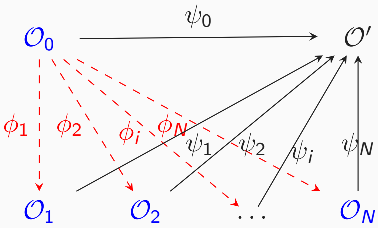
Optimization 1: Make the challenge space bigger (Multiple public keys)



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
com $\leftarrow \mathcal{O}'$	$\xrightarrow{\text{com}}$	
	$\xleftarrow{\text{ch}}$	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 $1/N$

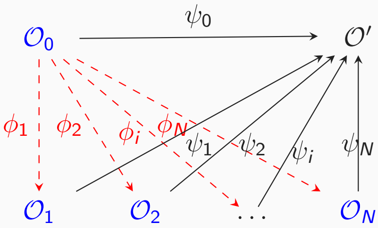
Optimization 1: Make the challenge space bigger (Multiple public keys)



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

- **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!**

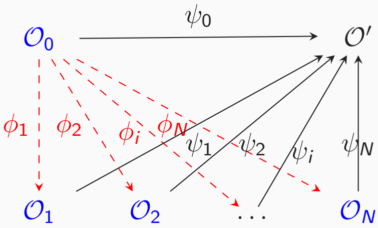
Optimization 1: Make the challenge space bigger (Multiple public keys)



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

- **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

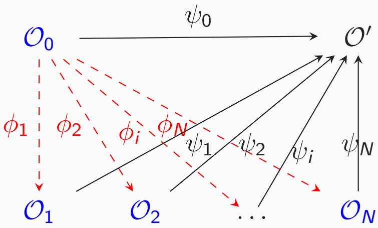
Optimization 1: Make the challenge space bigger (Multiple public keys)



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

- **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

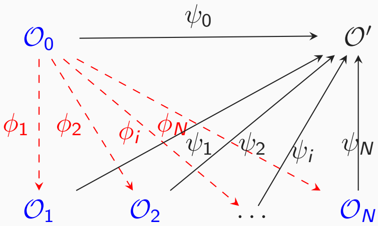
Optimization 1: Make the challenge space bigger (Multiple public keys)



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

- **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
- **There is a cost - N-fold increase in public and private key**

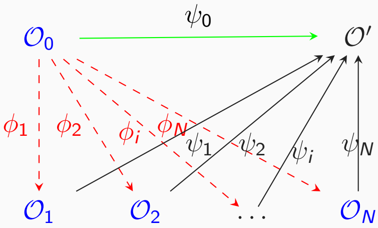
Optimization 1: Make the challenge space bigger (Multiple public keys)



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
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

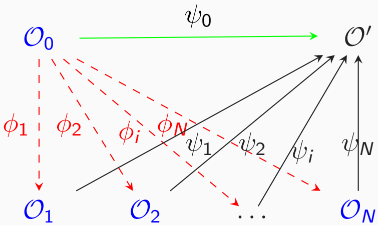
Optimization 2: Reduce signature size by using seeds



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

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

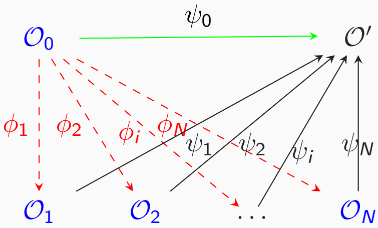
Optimization 2: Reduce signature size by using seeds



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

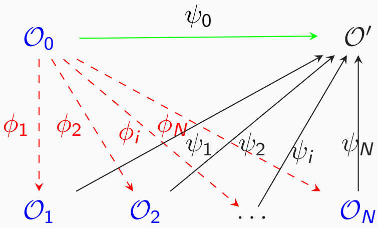
- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed

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 $\leftarrow \psi_{\text{ch}_1}, \psi_{\text{ch}_2}, \dots, \psi_{\text{ch}_r}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\text{ch}_i}(\mathcal{O}_{\text{ch}_i})$

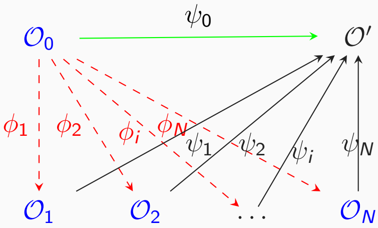
- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature
  - ψ_0 can be reconstructed from the seed
- **Problem:** This works only for ch $= 0$, and probability of choosing challenge 0 is $1/N$



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

- **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

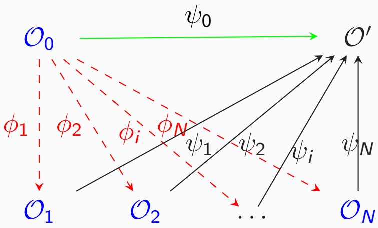
Optimization 2: Reduce signature size by using seeds



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

- **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 $\mathrm{ch} = 0$, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
  - $\Rightarrow$ **signature is not of constant size**

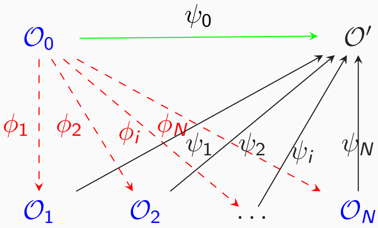
Optimization 2: Reduce signature size by using seeds



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

- **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 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

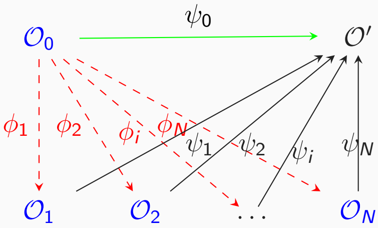
Optimization 2: Reduce signature size by using seeds



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

- **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 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

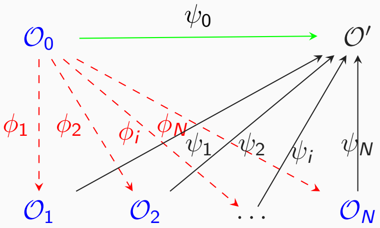
Optimization 2: Reduce signature size by using seeds



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

- **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 $\text{ch} = 0$, 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 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

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 $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:

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 $\mathrm{EQ}(\mathcal{O}_0, \mathcal{O}_1)$:
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–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–van Woerden 2022]
- **Matrix code equivalence** - [Reijnders–Samardjiska–Trimoska 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–van Woerden 2022]
- **Matrix code equivalence** - [Reijnders–Samardjiska–Trimoska 2022]
- ...

Equivalence problems for MEDS and ALTEQ

MEDS: Matrix Code Equivalence

- MEDS is based on the following equivalence problem.
- **Matrix code** - a subspace of $\mathcal{M}_{m \times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**.

- MEDS is based on the following equivalence problem.
- **Matrix code** - a subspace of $\mathcal{M}_{m\times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**.

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

MCE$(k, n, m, q, \mathcal{C}, \mathcal{D})$:

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

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

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

- ALTEQ is based on the following equivalence problem.
- **Alternating trilinear form** - a map $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ that
  (1) is linear in each argument, and
  (2) evaluates to 0 whenever two arguments are the same.

- ALTEQ is based on the following equivalence problem.
- **Alternating trilinear form** - a map $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ that
  (1) is linear in each argument, and
  (2) evaluates to 0 whenever two arguments are the same.

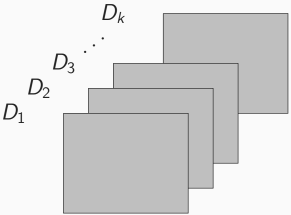
Alternating Trilinear Form Equivalence (ATFE) [Grochow-Qiao-Tang, 2021]

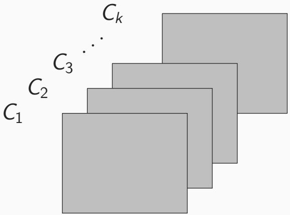
$\mathrm{ALTEQ}(n, q, \phi, \psi)$:

Input: Two alternating trilinear forms $\phi, \psi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$.

Question: Find – if any – $\mathbf{A} \in \mathrm{GL}_n(q)$ s.t. for any $u, v, w \in \mathbb{F}_q^n$, $\phi(u, v, w) = \psi(\mathbf{A}^t(u), \mathbf{A}^t(v), \mathbf{A}^t(w))$.

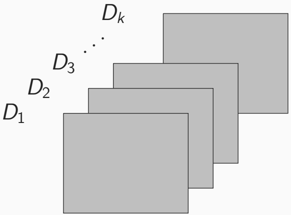
Matrix codes:

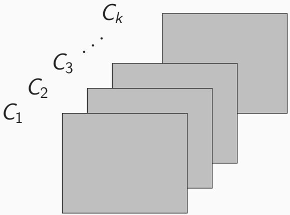




MCE and ATFE look very similar!

Matrix codes:



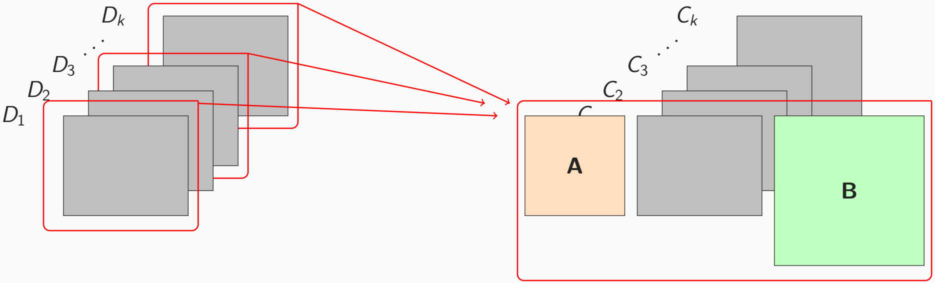


MCE:

- matrix codes of rectangular matrices

MCE and ATFE look very similar!

Matrix codes:



MCE:

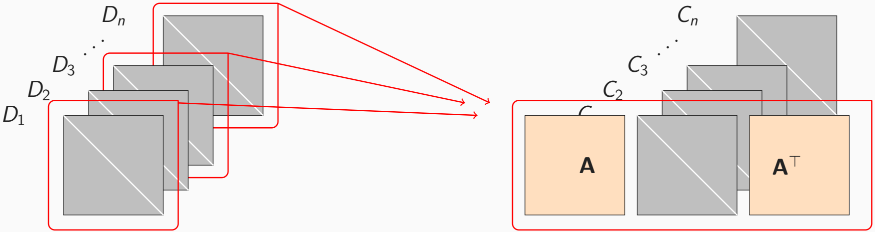
- matrix codes of rectangular matrices
- isometry $(\mathbf{A}, \mathbf{B})$

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

- An alternating trilinear form is $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$.
- We can record ϕ as an $n \times n \times n$ 3-way array $\mathtt{C} = [c_{i,j,k}]$, where $c_{i,j,k} = \phi(e_i, e_j, e_k)$.
  - Note that $c_{i,j,k} = -c_{j,i,k} = -c_{k,j,i} = -c_{i,k,j} = c_{j,k,i} = c_{k,i,j}$.
- A 3-way array $\mathtt{C}$ can also be represented as a matrix tuple $(C_1, \ldots, C_n)$, $C_i \in \mathcal{M}_n(q)$.

MCE and ATFE look very similar!

- An alternating trilinear form is $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$.
- We can record ϕ as an $n \times n \times n$ 3-way array $\mathtt{C} = [c_{i,j,k}]$, where $c_{i,j,k} = \phi(e_i, e_j, e_k)$.
  - Note that $c_{i,j,k} = -c_{j,i,k} = -c_{k,j,i} = -c_{i,k,j} = c_{j,k,i} = c_{k,i,j}$.
- A 3-way array $\mathtt{C}$ can also be represented as a matrix tuple $(C_1, \ldots, C_n)$, $C_i \in \mathcal{M}_n(q)$.



ATFE:

- matrix codes with "symmetries in the three directions".
- isometry $(\mathbf{A}, \mathbf{A}^\top)$ and $\mathbf{A}$ on the third direction too

MCE and ATFE are polynomial-time equivalent

- The objects in MCE and ATFE are both 3-way arrays.
  - A 2-way array, $[c_{i,j}]$, is a matrix.
  - A 3-way array, $[c_{i,j,k}]$, is sometimes called a 3-tensor.
  - The 3-way arrays from ATFE are subject to certain structural constraints.

MCE and ATFE are polynomial-time equivalent

- The objects in MCE and ATFE are both 3-way arrays.
  - A 2-way array, $[c_{i,j}]$, is a matrix.
  - A 3-way array, $[c_{i,j,k}]$, is sometimes called a 3-tensor.
  - The 3-way arrays from ATFE are subject to certain structural constraints.
- The isomorphisms in MCE and ATFE are both invertible matrices.
  - $L, R \in \mathrm{GL}_n(q)$ sends $C \in \mathcal{M}_n(q)$ to $L^t C R$.
  - $L, R, T = (t_{i,j}) \in \mathrm{GL}_n(q)$ sends $(C_1, \ldots, C_n) \in \mathcal{M}_n(q)^n$ to $(L^t C_1' R, \ldots, L^t C_n' R)$, where $C_i' = \sum_j t_{i,j} C_j$.
  - The isomorphism in ATFE imposes that $L = R = T$.

MCE **and** ATFE **are polynomial-time equivalent**

- The objects in MCE and ATFE are both 3-way arrays.
  - A 2-way array, $[c_{i,j}]$, is a matrix.
  - A 3-way array, $[c_{i,j,k}]$, is sometimes called a 3-tensor.
  - The 3-way arrays from ATFE are subject to certain structural constraints.
- The isomorphisms in MCE and ATFE are both invertible matrices.
  - $L, R \in \mathrm{GL}_n(q)$ sends $C \in \mathcal{M}_n(q)$ to $L^t C R$.
  - $L, R, T = (t_{i,j}) \in \mathrm{GL}_n(q)$ sends $(C_1, \dots, C_n) \in \mathcal{M}_n(q)^n$ to $(L^t C_1' R, \dots, L^t C_n' R)$, where $C_i' = \sum_j t_{i,j} C_j$.
  - The isomorphism in ATFE imposes that $L = R = T$.

Theorem ([Grochow-Qiao-Tang, 2023])

MCE *and* ATFE *are polynomial-time equivalent.*

A complexity class for isomorphism problems of algebraic structures

- Relations between isomorphism problems for some algebraic structures are studied in [Reijnders–Samardjiska–Trimoska, Grochow–Qiao–Tang, D'Alconzo, Couvreur–Debris-Alazard–Gaborit. . .]

A complexity class for isomorphism problems of algebraic structures

- Relations between isomorphism problems for some algebraic structures are studied in [Reijnders–Samardjiska–Trimoska, Grochow–Qiao–Tang, D'Alconzo, Couvreur–Debris-Alazard–Gaborit. . .]
- The complexity class TI was defined in [Grochow-Qiao], consisting of problems polynomial-time reducible to MCE.
  - MCE was called 3-Tensor Isomorphism in [Grochow-Qiao].
  - In analogy with the complexity class GI for Graph Isomorphism.
- MCE and ATFE are TI-complete.

A complexity class for isomorphism problems of algebraic structures

- Relations between isomorphism problems for some algebraic structures are studied in [Reijnders–Samardjiska–Trimoska, Grochow–Qiao–Tang, D'Alconzo, Couvreur–Debris-Alazard–Gaborit. . .]
- The complexity class TI was defined in [Grochow-Qiao], consisting of problems polynomial-time reducible to MCE.
  - MCE was called 3-Tensor Isomorphism in [Grochow-Qiao].
  - In analogy with the complexity class GI for Graph Isomorphism.
- MCE and ATFE are TI-complete.
- TI-complete problems include isomorphism problems for tensors, finite groups, (associative and Lie) algebras, (systems of) polynomials. . .

Relations with other isomorphism problems

- TI-complete problems appear in computational group theory, multivariate cryptography, and quantum information.
  - Experiences from these areas suggest that TI-complete problems are difficult to solve in practice.

Relations with other isomorphism problems

- TI-complete problems appear in computational group theory, multivariate cryptography, and quantum information.
  - Experiences from these areas suggest that TI-complete problems are difficult to solve in practice.
- Isomorphism problems for cubic forms and quadratic polynomial systems, as studied since 1996 [Patarin], are TI-complete.
  - Results from the study of polynomial isomorphism are valuable for MCE and ATFE.

Relations with other isomorphism problems

- TI-complete problems appear in computational group theory, multivariate cryptography, and quantum information.
  - Experiences from these areas suggest that TI-complete problems are difficult to solve in practice.
- Isomorphism problems for cubic forms and quadratic polynomial systems, as studied since 1996 [Patarin], are TI-complete.
  - Results from the study of polynomial isomorphism are valuable for MCE and ATFE.
- Linear code monomial equivalence and graph isomorphism are in TI [Couvreur–Debris-Alazard–Gaborit, Grochow–Qiao].
  - Linear code monomial equivalence supports LESS.

Why use MCE and ATFE in post-quantum cryptography?

- A natural development of Shor's quantum algorithms for integer factorisation and discrete logarithm is the hidden subgroup problem framework.
- MCE and ATFE can be cast in this framework for general linear groups.

Why use MCE and ATFE in post-quantum cryptography?

- A natural development of Shor's quantum algorithms for integer factorisation and discrete logarithm is the hidden subgroup problem framework.
- MCE and ATFE can be cast in this framework for general linear groups.
- A strong negative evidence for the "standard technique" to work in this setting [Hallgren-Moore-Rötteler-Russell-Sen, 2010].

  [Moore-Russell-Vazirani] . . . the strongest such insights we have about the limits of quantum algorithms.

Cryptanalysis for MCE and ATFE

Algorithms for MCE and ATFE

- Consider 3-way arrays of size $n \times n \times n$ over $\mathbb{F}_q$ under the action of (L, R, T) or $(T, T, T) \in \mathrm{GL}_n(q) \times \mathrm{GL}_n(q) \times \mathrm{GL}_n(q)$.
- Brute-force algorithm: $q^{n^2} \cdot \mathrm{poly}(n, \log q)$.
  - After fixing T, to recover L and R can be done in time $\mathrm{poly}(n, \log q)$.

Algorithms for MCE and ATFE

- Consider 3-way arrays of size $n \times n \times n$ over $\mathbb{F}_q$ under the action of (L, R, T) or $(T, T, T) \in \mathrm{GL}_n(q) \times \mathrm{GL}_n(q) \times \mathrm{GL}_n(q)$.
- Brute-force algorithm: $q^{n^2} \cdot \mathrm{poly}(n, \log q)$.
  - After fixing T, to recover L and R can be done in time $\mathrm{poly}(n, \log q)$.
- We will introduce three approaches.
  - Direct Gröbner basis attack.
  - Hybrid Gröbner basis: $q^n \cdot \mathrm{poly}(n, \log q)$.
  - Utilising low-rank points (via birthday paradox and invariants).

Direct Gröbner basis attack: the basic idea

- Let $\mathtt{C} = [c_{i,j,k}]$ and $\mathtt{D} = [d_{i,j,k}]$ be two $n \times n \times n$ 3-way arrays over $\mathbb{F}_q$.
- We view $\mathtt{C}$ as a matrix tuple $(C_1, \ldots, C_n)$, $C_i \in \mathcal{M}_n(q)$.

Direct Gröbner basis attack: the basic idea

- Let $\mathtt{C} = [c_{i,j,k}]$ and $\mathtt{D} = [d_{i,j,k}]$ be two $n \times n \times n$ 3-way arrays over $\mathbb{F}_q$.
- We view $\mathtt{C}$ as a matrix tuple $(C_1, \ldots, C_n)$, $C_i \in \mathcal{M}_n(q)$.
- Recall that $L, R, T = (t_{i,j}) \in \mathrm{GL}_n(q)$ sends $(C_1, \ldots, C_n) \in \mathcal{M}_n(q)^n$ to $(L^t C_1' R, \ldots, L^t C_n' R)$, where $C_i' = \sum_j t_{i,j} C_j$.

- Let $\mathtt{C} = [c_{i,j,k}]$ and $\mathtt{D} = [d_{i,j,k}]$ be two $n \times n \times n$ 3-way arrays over $\mathbb{F}_q$.
- We view $\mathtt{C}$ as a matrix tuple $(C_1, \dots, C_n)$, $C_i \in \mathcal{M}_n(q)$.
- Recall that $L, R, T = (t_{i,j}) \in \mathrm{GL}_n(q)$ sends $(C_1, \dots, C_n) \in \mathcal{M}_n(q)^n$ to $(L^t C_1' R, \dots, L^t C_n' R)$, where $C_i' = \sum_j t_{i,j} C_j$.
- Viewing the entries of L, R and T as variables, the question is whether $(L^t C_1' R, \dots, L^t C_n' R) = (D_1, \dots, D_n)$.
  - This amounts to n^3 cubic polynomials in $3n^2$ variables.

Direct Gröbner basis attack: more efficient modellings

Cubic modelling $(L^t C_1' R, \ldots, L^t C_n' R) = (D_1, \ldots, D_n)$ where $C_i' = \sum_j t_{i,j} C_j$.

- This gives rise to n^3 cubic polynomials in $3n^2$ variables for MCE.
- And $\binom{n}{3}$ cubic polynomials in n^2 variables for ATFE.

Direct Gröbner basis attack: more efficient modellings

Cubic modelling $(L^t C_1' R, \ldots, L^t C_n' R) = (D_1, \ldots, D_n)$ where $C_i' = \sum_j t_{i,j} C_j$.

- This gives rise to n^3 cubic polynomials in $3n^2$ variables for MCE.
- And $\binom{n}{3}$ cubic polynomials in n^2 variables for ATFE.

Quadratic inverse modelling For ATFE, let $T' = [t'_{i,j}]$. Then set

$$(T^t C_1 T, \ldots, T^t C_n T) = (D_1', \ldots, D_n') \text{ where } D_i' = \textstyle\sum_j t'_{i,j} D_j, \text{ and } TT' = I_n.$$

- This is by [Bouillaguet-Faugère-Fouque-Perret, 2010].
- $n \cdot \binom{n}{2} + n^2$ quadratic polynomials in $2n^2$ variables.

Direct Gröbner basis attack: more efficient modellings

Cubic modelling $(L^t C'_1 R, \ldots, L^t C'_n R) = (D_1, \ldots, D_n)$ where $C'_i = \sum_j t_{i,j} C_j$.

- This gives rise to n^3 cubic polynomials in $3n^2$ variables for MCE.
- And $\binom{n}{3}$ cubic polynomials in n^2 variables for ATFE.

Quadratic inverse modelling For ATFE, let $T' = [t'_{i,j}]$. Then set

$$(T^t C_1 T, \ldots, T^t C_n T) = (D'_1, \ldots, D'_n) \text{ where } D'_i = \sum_j t'_{i,j} D_j, \text{ and } TT' = I_n.$$

- This is by [Bouillaguet-Faugère-Fouque-Perret, 2010].
- $n \cdot \binom{n}{2} + n^2$ quadratic polynomials in $2n^2$ variables.

Quadratic dual modelling Use the dual space of $\mathcal{D}$ to express that $L^t C_i R \in \mathcal{D}$.

- This is by [Chou-Niederhagen-Persichetti-Randrianarisoa-Reijnders-Samardjiska-Trimoska].
- This gives rise to $n \cdot (n^2 - n)$ homogeneous quadratic polynomials in $2n^2$ variables for MCE.
- And $n \cdot (\binom{n}{2} - n)$ quadratic polynomials in n^2 variables for ATFE.
- Note that some syzygies arise, complicating the analysis [MEDS spec].

Hybrid Gröbner basis attacks

- We set up $n \times n$ variable matrices L and R for MCE (or T and T' for ATFE).
- In [Faugère-Perret, 2006], it was discovered that Gröbner basis runs in polynomial time, <u>provided</u> that one (or two) rows of L are known.

Hybrid Gröbner basis attacks

- We set up $n \times n$ variable matrices L and R for MCE (or T and T' for ATFE).
- In [Faugère-Perret, 2006], it was discovered that Gröbner basis runs in polynomial time, provided that one (or two) rows of L are known.
- For ATFE, knowing one row of T is enough, leading to a $q^n \cdot \mathrm{poly}(n, \log q)$-time algorithm.
- For MCE, knowing two rows of L is enough, leading to an $q^{2n} \cdot \mathrm{poly}(n, \log q)$-time algorithm.

Hybrid Gröbner basis attacks

- We set up $n \times n$ variable matrices L and R for MCE (or T and T' for ATFE).
- In [Faugère-Perret, 2006], it was discovered that Gröbner basis runs in polynomial time, provided that one (or two) rows of L are known.
- For ATFE, knowing one row of T is enough, leading to a $q^n \cdot \mathrm{poly}(n, \log q)$-time algorithm.
- For MCE, knowing two rows of L is enough, leading to an $q^{2n} \cdot \mathrm{poly}(n, \log q)$-time algorithm.
- Further observations from [Beullens, 2023]:
  - Knowing one row of T up to scalar is enough.
  - For low-rank points, the kernel information can be incorporated.

Utilising low-rank points

- Let $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be an alternating trilinear form.
- For $u \in \mathbb{F}_q^n$, let $\phi_u : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ by $\phi_u(v, w) = \phi(u, v, w)$.
- An isomorphism invariant for u: $r = \mathrm{Rank}(\phi_u)$.

Utilising low-rank points

- Let $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be an alternating trilinear form.
- For $u \in \mathbb{F}_q^n$, let $\phi_u : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ by $\phi_u(v, w) = \phi(u, v, w)$.
- An isomorphism invariant for u: $r = \text{Rank}(\phi_u)$.
- Algorithms based on birthday paradox and hybrid Gröbner basis [Bouillaguet-Fouque-Véber, 2013; Beullens, 2023].
  - Suppose there exist $\approx q^k$-many rank-r points for a random ϕ.

  (1) Sample $q^{k/2}$-many rank-r points for ϕ and ψ, respectively.
  (2) For every pair, use hybrid Gröbner basis to find a "matched" pair.

  - Algorithm cost: $O(q^{k/2} \cdot \text{samp-cost} + q^k \cdot \text{gb-cost})$.

Utilising low-rank points

- Let $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be an alternating trilinear form.
- For $u \in \mathbb{F}_q^n$, let $\phi_u : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ by $\phi_u(v, w) = \phi(u, v, w)$.
- An isomorphism invariant for u: $r = \mathrm{Rank}(\phi_u)$.
- Algorithms based on birthday paradox and hybrid Gröbner basis [Bouillaguet-Fouque-Véber, 2013; Beullens, 2023].
  - Suppose there exist $\approx q^k$-many rank-r points for a random ϕ.

  (1) Sample $q^{k/2}$-many rank-r points for ϕ and ψ, respectively.
  (2) For every pair, use hybrid Gröbner basis to find a "matched" pair.

  - Algorithm cost: $O(q^{k/2} \cdot \text{samp-cost} + q^k \cdot \text{gb-cost})$.
- Sampling step: min-rank or graph-walking [Beullens, 2023]

Utilising low-rank points, cont'd

- Algorithms based on distinguishing isomorphism invariants with low-rank points [Bouillaguet-Fouque-Véber, 2013; Beullens, 2023].
  - Suppose there exist $\approx q^k$-many rank-r points for a random ϕ.
  - Suppose there exist distinguishing isomorphism invariants associated with such points.

Utilising low-rank points, cont'd

- Algorithms based on distinguishing isomorphism invariants with low-rank points [Bouillaguet-Fouque-Véber, 2013; Beullens, 2023].
  - Suppose there exist $\approx q^k$-many rank-r points for a random ϕ.
  - Suppose there exist distinguishing isomorphism invariants associated with such points.

(1) Sample $q^{k/2}$-many rank-r points for ϕ and ψ, respectively.
(2) For every point, compute the isomorphism invariant.
(3) By birthday paradox, there exists a pair of points of the same invariant. Use hybrid Gröbner basis to complete.

Utilising low-rank points, cont'd

- Algorithms based on distinguishing isomorphism invariants with low-rank points [Bouillaguet-Fouque-Véber, 2013; Beullens, 2023].
  - Suppose there exist $\approx q^k$-many rank-r points for a random ϕ.
  - Suppose there exist distinguishing isomorphism invariants associated with such points.

(1) Sample $q^{k/2}$-many rank-r points for ϕ and ψ, respectively.
(2) For every point, compute the isomorphism invariant.
(3) By birthday paradox, there exists a pair of points of the same invariant. Use hybrid Gröbner basis to complete.

  - Algorithm cost: $O(q^{k/2} \cdot (\text{samp-cost} + \text{inv-cost}) + \text{gb-cost})$.

Utilising low-rank points, cont'd

- Algorithms based on distinguishing isomorphism invariants with low-rank points [Bouillaguet-Fouque-Véber, 2013; Beullens, 2023].
  - Suppose there exist $\approx q^k$-many rank-r points for a random ϕ.
  - Suppose there exist distinguishing isomorphism invariants associated with such points.

(1) Sample $q^{k/2}$-many rank-r points for ϕ and ψ, respectively.
(2) For every point, compute the isomorphism invariant.
(3) By birthday paradox, there exists a pair of points of the same invariant. Use hybrid Gröbner basis to complete.

  - Algorithm cost: $O(q^{k/2} \cdot (\text{samp-cost} + \text{inv-cost}) + \text{gb-cost})$.

- Distinguishing isomorphism invariant candidates: ranks of the neighbours of low-rank points, and more [Narayanan-Qiao-Tang].

Parameters and performances of MEDS and ALTEQ

Parameters and performance of MEDS

Level	param. set	public key size (KB)	signature size (KB)	key gen (ms)	sign (ms)	verify (ms)
I	MEDS-9923	9.9	9.9	1	272	271
	MEDS-13220	13.2	13	1.3	46.7	46
III	MEDS-41711	41.7	41	5.1	779	762
	MEDS-69497	55.6	54.7	6.7	203.8	200.4

Table: An overview of the parameters and performance of MEDS.

Optimizations:

- **Standard:** Multiple Public Keys + Fixed-Weight Challenge Strings + Seed tree
- **New:** Public Key Compression
  - generate public key partially from seed $\Rightarrow$ signature size reduction
  - **Work in progress:** use similar idea during signing

Parameters and performance of ALTEQ

Level	mode	public key size (KB)	signature size (KB)	key gen (ms)	sign (ms)	verify (ms)
I	Balanced	8	16	0.093	0.629	0.496
	ShortSig	512	10	1.902	0.194	0.092
III	Balanced	32	48	0.582	6.986	6.483
	ShortSig	1024	24	5.152	1.705	1.304

Table: An overview of the parameters and performance of ALTEQ.

Optimizations:

- **Standard:** Multiple Public Keys + Fixed-Weight Challenge Strings (+ Seed tree)
- **New:** Invertible matrix decomposition
  - Represent an invertible matrix as a product of column matrices for faster signing and verification

Summary

- Digital signature based on equivalence problems: design and optimisations
- Matrix code equivalence (MCE) and alternating trilinear form equivalence (ATFE)
- Algorithms for MCE and ATFE
- MEDS and ALTEQ: parameters and performances

Thank you for listening!