Radboud University

Matrix Code Equivalence and Applications

Monika Trimoska (based on joint work with **Tung Chou**, **Ruben Niederhagen**, **Edoardo Persichetti**, **Tovohery Hajatiana Randrianarisoa**, **Krijn Reijnders** and **Simona Samardjiska**)

Séminaire Laboratoire MIS
November 24th, 2022

Matrix Code Equivalence (MCE)

The Matrix Code Equivalence Problem

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

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

Isometry μ: a homomorphism of matrix codes $\mathcal{C}\to\mathcal{D}$ such that for all $\mathbf{C}\in\mathcal{C}$,

$$\mathrm{Rank}\,\mathbf{C} = \mathrm{Rank}\,\mu(\mathbf{C})$$

The Matrix Code Equivalence Problem

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

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

Isometry μ: a homomorphism of matrix codes $\mathcal{C} \to \mathcal{D}$ such that for all $\mathbf{C} \in \mathcal{C}$,

$$\mathrm{Rank}\,\mathbf{C} = \mathrm{Rank}\,\mu(\mathbf{C})$$

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

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

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

Question: Find – if any – an isometry $\mu : \mathcal{C} \to \mathcal{D}$.

The Matrix Code Equivalence Problem

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

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

Isometry μ: a homomorphism of matrix codes $\mathcal{C} \to \mathcal{D}$ such that for all $\mathbf{C} \in \mathcal{C}$,

$$\mathrm{Rank}\,\mathbf{C} = \mathrm{Rank}\,\mu(\mathbf{C})$$

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

MCE$(k, n, m, \mathcal{C}, \mathcal{D})$:
Input: Two k-dimensional matrix codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m\times n}(\mathbb{F}_q)$
Question: Find – if any – an isometry $\mu : \mathcal{C} \to \mathcal{D}$.

Known: Any isometry $\mu : \mathcal{C} \to \mathcal{D}$ can be written, for some $\mathbf{A} \in \mathrm{GL}_m(q), \mathbf{B} \in \mathrm{GL}_n(q)$, as

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

Known results [Couvreur, Debris–Alazard & Gaborit, 2021]

$$\mu : \mathbf{C} \mapsto \mathbf{ACB} \in \mathcal{D}, \quad \text{with } \mathbf{A} \in \mathrm{GL}_m(q) \text{ and } \mathbf{B} \in \mathrm{GL}_n(q)$$

- when $\mathbf{A} = \mathrm{Id}_m$, or $\mathbf{B} = \mathrm{Id}_n$, finding μ is easy (MCRE)

Known results [Couvreur, Debris–Alazard & Gaborit, 2021]

$$\mu : \mathbf{C} \mapsto \mathbf{ACB} \in \mathcal{D}, \quad \text{with } \mathbf{A} \in \mathrm{GL}_m(q) \text{ and } \mathbf{B} \in \mathrm{GL}_n(q)$$

- when $\mathbf{A} = \mathrm{Id}_m$, or $\mathbf{B} = \mathrm{Id}_n$, finding μ is easy (MCRE)
- implicit upper bound $\mathcal{O}^*(q^{m^2})$ time: brute force smallest side, then solve MCRE

Known results [Couvreur, Debris–Alazard & Gaborit, 2021]

$$\mu : \mathbf{C} \mapsto \mathbf{ACB} \in \mathcal{D}, \quad \text{with } \mathbf{A} \in \mathrm{GL}_m(q) \text{ and } \mathbf{B} \in \mathrm{GL}_n(q)$$

- when $\mathbf{A} = \mathrm{Id}_m$, or $\mathbf{B} = \mathrm{Id}_n$, finding μ is easy (MCRE)
- implicit upper bound $\mathcal{O}^*(q^{m^2})$ time: brute force smallest side, then solve MCRE
- code equivalence for $\mathbb{F}_{q^m}$-linear codes with rank metric reduces to MCRE

Known results [Couvreur, Debris–Alazard & Gaborit, 2021]

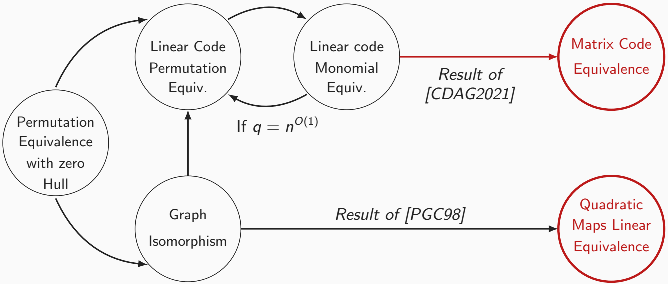
$$\mu : \mathbf{C} \mapsto \mathbf{ACB} \in \mathcal{D}, \quad \text{with } \mathbf{A} \in \mathrm{GL}_m(q) \text{ and } \mathbf{B} \in \mathrm{GL}_n(q)$$

- when $\mathbf{A} = \mathrm{Id}_m$, or $\mathbf{B} = \mathrm{Id}_n$, finding μ is easy (MCRE)
- implicit upper bound $\mathcal{O}^*(q^{m^2})$ time: brute force smallest side, then solve MCRE
- code equivalence for $\mathbb{F}_{q^m}$-linear codes with rank metric reduces to MCRE
- MCE is at least as hard as Monomial Equivalence Problem in the Hamming metric

Known results [Couvreur, Debris–Alazard & Gaborit, 2021]

$$\mu : \mathbf{C} \mapsto \mathbf{ACB} \in \mathcal{D}, \quad \text{with } \mathbf{A} \in \mathrm{GL}_m(q) \text{ and } \mathbf{B} \in \mathrm{GL}_n(q)$$

- when $\mathbf{A} = \mathrm{Id}_m$, or $\mathbf{B} = \mathrm{Id}_n$, finding μ is easy (MCRE)
- implicit upper bound $\mathcal{O}^*(q^{m^2})$ time: brute force smallest side, then solve MCRE
- code equivalence for $\mathbb{F}_{q^m}$-linear codes with rank metric reduces to MCRE
- MCE is at least as hard as Monomial Equivalence Problem in the Hamming metric



What is QMLE?

Multivariate crypto basics

- systems of multivariate polynomials $\mathcal{P} = (p_1, p_2, \ldots, p_k)$, every p_s polynomial in N variables $x_1, \ldots, x_N$

Multivariate crypto basics

- systems of multivariate polynomials $\mathcal{P} = (p_1, p_2, \ldots, p_k)$, every p_s polynomial in N variables $x_1, \ldots, x_N$
- most interesting when each p_s is at most degree 2

$$p_s(x_1, \ldots, x_N) = \sum \gamma_{ij}^{(s)} x_i x_j + \sum \beta_i^{(s)} x_i + \alpha^{(s)}, \qquad \alpha^{(s)}, \beta_i^{(s)}, \gamma_{ij}^{(s)} \in \mathbb{F}_q$$

Multivariate crypto basics

- systems of multivariate polynomials $\mathcal{P} = (p_1, p_2, \ldots, p_k)$, every p_s polynomial in N variables $x_1, \ldots, x_N$
- most interesting when each p_s is at most degree 2 and homogeneous

$$p_s(x_1, \ldots, x_N) = \sum \gamma_{ij}^{(s)} x_i x_j \qquad \gamma_{ij}^{(s)} \in \mathbb{F}_q$$

Multivariate crypto basics

- systems of multivariate polynomials $\mathcal{P} = (p_1, p_2, \ldots, p_k)$, every p_s polynomial in N variables $x_1, \ldots, x_N$
- most interesting when each p_s is at most degree 2 and homogeneous

$$p_s(x_1, \ldots, x_N) = \sum \gamma_{ij}^{(s)} x_i x_j \qquad\qquad \gamma_{ij}^{(s)} \in \mathbb{F}_q$$

Quadratic Maps Linear Equivalence (QMLE) problem

QMLE$(N, k, \mathcal{F}, \mathcal{P})$:

Input: Two k-tuples of quadratic maps
$\mathcal{F} = (f_1, f_2, \ldots, f_k),\ \mathcal{P} = (p_1, p_2, \ldots, p_k) \in \mathbb{F}_q[x_1, \ldots, x_N]^k$

Question: Find – if any – $\mathbf{S} \in \mathrm{GL}_N(q), \mathbf{T} \in \mathrm{GL}_k(q)$ such that

$$\mathcal{P}(\mathbf{x}) = \mathcal{F}(\mathbf{xS}) \cdot \mathbf{T}$$

Quadratic maps

$$p_s = \sum \gamma_{ij}^{(s)} x_i x_j \quad = \quad (x_1, \ldots, x_N) \underbrace{\begin{pmatrix} \gamma_{11} & \cdots & \frac{\gamma_{1N}}{2} \\ & & \\ \frac{\gamma_{N1}}{2} & \cdots & \gamma_{NN} \end{pmatrix}}_{\mathbf{P}^{(s)} \ \in \ \mathcal{M}_{N\times N}(\mathbb{F}_q)} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$$

Quadratic maps

$$p_s = \sum \gamma_{ij}^{(s)} x_i x_j \quad = \quad (x_1, \ldots, x_N) \underbrace{\begin{pmatrix} \gamma_{11} & \cdots & \frac{\gamma_{1N}}{2} \\ & & \\ \frac{\gamma_{N1}}{2} & \cdots & \gamma_{NN} \end{pmatrix}}_{\mathbf{P}^{(s)} \in \mathcal{M}_{N \times N}(\mathbb{F}_q)} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$$

so with $\mathbf{x} = (x_1, \ldots, x_N)$, we get $p_s(\mathbf{x}) = \mathbf{x}\mathbf{P}^{(s)}\mathbf{x}^T$

Quadratic maps

$$p_s = \sum \gamma_{ij}^{(s)} x_i x_j \quad = \quad (x_1, \ldots, x_N) \underbrace{\begin{pmatrix} \gamma_{11} & \cdots & \frac{\gamma_{1N}}{2} \\ & & \\ \frac{\gamma_{N1}}{2} & \cdots & \gamma_{NN} \end{pmatrix}}_{\mathbf{P}^{(s)} \in \mathcal{M}_{N\times N}(\mathbb{F}_q)} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$$

so with $\mathbf{x} = (x_1, \ldots, x_N)$, we get $p_s(\mathbf{x}) = \mathbf{x}\mathbf{P}^{(s)}\mathbf{x}^T$

so QMLE can be rewritten in matrix form

$$\sum_{1 \leqslant r \leqslant k} \widetilde{t}_{rs} \mathbf{P}^{(r)} = \mathbf{S}\mathbf{F}^{(s)}\mathbf{S}^{\top}, \quad \forall s, 1 \leqslant s \leqslant k,$$

where $\widetilde{t}_{ij}$ are entries of $\mathbf{T}^{-1}$

Solving MCE [Reijnders, Samardjiska & T., 2022]

- reduction: an MCE instance $(k, n, m, \mathcal{C}, \mathcal{D})$ results in a QMLE instance $(m+n, k, \mathcal{F}, \mathcal{P})$ with

$$\mathbf{S} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^\top \end{bmatrix}$$

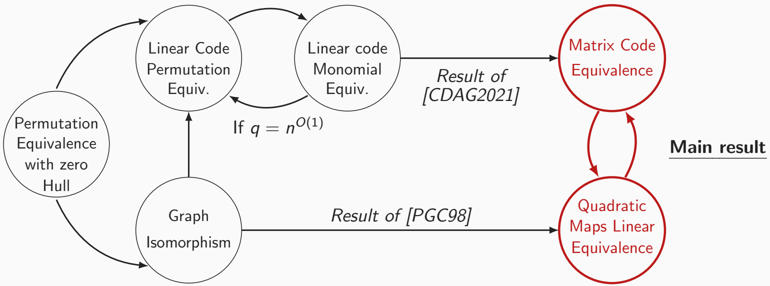
Solving MCE [Reijnders, Samardjiska & T., 2022]

- reduction: an MCE instance $(k, n, m, \mathcal{C}, \mathcal{D})$ results in a QMLE instance $(m+n, k, \mathcal{F}, \mathcal{P})$ with

$$\mathbf{S} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^\top \end{bmatrix}$$

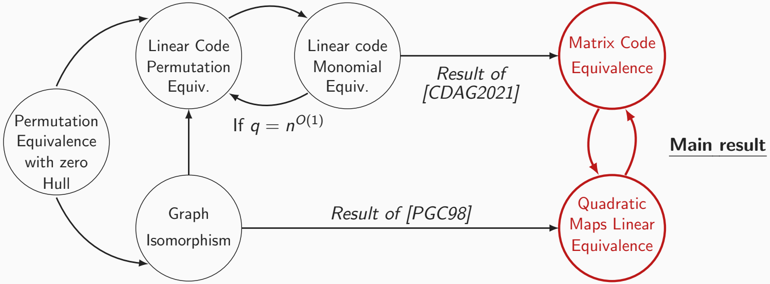
- solving the instance using a birthday-based algorithm $\mathcal{O}^*(q^{2/3(m+n)})$ [Bouillaguet, Fouque & Véber, 2013]

Solving MCE [Reijnders, Samardjiska & T., 2022]



- Main result: MCE **is equivalent to** QMLE

Solving MCE [Reijnders, Samardjiska & T., 2022]



- Main result: MCE **is equivalent to** QMLE
- Gives **improved upper bound** to complexity of solving MCE (w.l.o.g. assume $m \leqslant n$)
  - solvable in $\mathcal{O}^*(q^{2/3(m+n)})$ time, when $k \leqslant n + m$ can be improved to $\mathcal{O}^*(q^m)$

General Birthday-based Equivalence Finder

$S_1, S_2 \in U$, $|S_1| = |S_2| = N$,

Problem: Find an equivalence function $\phi : S_1 \to S_2$

General Birthday-based Equivalence Finder

$S_1, S_2 \in U$, $|S_1| = |S_2| = N$,
Problem: Find an equivalence function $\phi : S_1 \to S_2$

Algorithm 1: General Birthday-based Equivalence Finder

Assumptions:

- Efficient predicate $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ,
- Efficient FindFunction: if collision $(b = \phi(a))$ return ϕ and $\bot$ otherwise.

General Birthday-based Equivalence Finder

$S_1, S_2 \in U$, $|S_1| = |S_2| = N$,
Problem: Find an equivalence function $\phi : S_1 \to S_2$

Algorithm 1: General Birthday-based Equivalence Finder

1: **function** $\textsc{SampleSet}(S, \mathbb{P})$
2: $\quad L \leftarrow \emptyset$
3: $\quad$ **repeat**
4: $\quad\quad a \xleftarrow{\$} S$
5: $\quad\quad$ **if** $\mathbb{P}(a)$ **then** $L \leftarrow L \cup \{a\}$
6: $\quad\quad$ **until** $|L| = \ell$
7: $\quad\quad$ **return** L

Assumptions:

- Efficient predicate $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ,
- Efficient $\textsc{FindFunction}$: if collision $(b = \phi(a))$ return ϕ and $\bot$ otherwise.

General Birthday-based Equivalence Finder

$S_1, S_2 \in U$, $|S_1| = |S_2| = N$,
Problem: Find an equivalence function $\phi : S_1 \to S_2$

Algorithm 1: General Birthday-based Equivalence Finder

function $\textsc{SampleSet}(S, \mathbb{P})$
  $L \leftarrow \emptyset$
  repeat
    $a \xleftarrow{\$} S$
    if $\mathbb{P}(a)$ **then** $L \leftarrow L \cup \{a\}$
    until $|L| = \ell$
    return L

function $\textsc{CollisionFind}(S_1, S_2)$
  $L_i \leftarrow \textsc{SampleSet}(S_i, \mathbb{P})$, $i \in \{1, 2\}$
  for all $(a, b) \in L_1 \times L_2$ **do**
    $\phi \leftarrow \textsc{FindFunction}(a, b)$
  if $\phi \neq \perp$ **then**
    return solution ϕ
return $\perp$

Assumptions:

- Efficient predicate $\mathbb{P} : U \to \{\top, \perp\}$ invariant under the equivalence ϕ,
- Efficient $\textsc{FindFunction}$: if collision $(b = \phi(a))$ return ϕ and $\perp$ otherwise.

Solving hQMLE using the Birthday-based Equivalence Finder

- Works **for any** $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ

Solving hQMLE using the Birthday-based Equivalence Finder

- Works **for any** $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ
  - Crucial for performance: **density** of the predicate $\mathbb{P}$

$$d = |U_\top| / |U|, \quad U_\top = \{x \in U \mid \mathbb{P}(x) = \top\}$$

Solving hQMLE using the Birthday-based Equivalence Finder

- Works **for any** $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ
  - Crucial for performance: **density** of the predicate $\mathbb{P}$

$$d = |U_\top|/|U|, \quad U_\top = \{x \in U \mid \mathbb{P}(x) = \top\}$$

  - dN elements in $S_1(S_2)$ satisfying $\mathbb{P}$

Solving hQMLE using the Birthday-based Equivalence Finder

- Works **for any** $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ
  - Crucial for performance: **density** of the predicate $\mathbb{P}$

$$d = |U_\top|/|U|, \quad U_\top = \{x \in U \mid \mathbb{P}(x) = \top\}$$

  - dN elements in $S_1(S_2)$ satisfying $\mathbb{P}$
  - we need lists of size $\ell = \sqrt{d \cdot N}$ for collision with success probability $1 - 1/e$

Solving hQMLE using the Birthday-based Equivalence Finder

- Works **for any** $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ
  - Crucial for performance: **density** of the predicate $\mathbb{P}$

$$d = |U_\top|/|U|, \quad U_\top = \{x \in U \mid \mathbb{P}(x) = \top\}$$

  - dN elements in $S_1(S_2)$ satisfying $\mathbb{P}$
  - we need lists of size $\ell = \sqrt{d \cdot N}$ for collision with success probability $1 - 1/e$
- Queries $\textsc{FindFunction}$ $d \cdot N$ times

Solving hQMLE using the Birthday-based Equivalence Finder

- Works **for any** $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ
  - Crucial for performance: **density** of the predicate $\mathbb{P}$

$$d = |U_\top|/|U|, \quad U_\top = \{x \in U \mid \mathbb{P}(x) = \top\}$$

  - dN elements in $S_1(S_2)$ satisfying $\mathbb{P}$
  - we need lists of size $\ell = \sqrt{d \cdot N}$ for collision with success probability $1 - 1/e$
- Queries $\textsc{FindFunction}$ $d \cdot N$ times
- Performs $\ell/d = \sqrt{N/d}$ checks in $\textsc{SampleSet}$

Solving hQMLE using the Birthday-based Equivalence Finder

- Works **for any** $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ
  - Crucial for performance: **density** of the predicate $\mathbb{P}$

$$d = |U_\top|/|U|, \quad U_\top = \{x \in U \mid \mathbb{P}(x) = \top\}$$

  - dN elements in $S_1(S_2)$ satisfying $\mathbb{P}$
  - we need lists of size $\ell = \sqrt{d \cdot N}$ for collision with success probability $1 - 1/e$
- Queries $\textsc{FindFunction}$ $d \cdot N$ times
- Performs $\ell/d = \sqrt{N/d}$ checks in $\textsc{SampleSet}$
- Best performance if blue and green balanced

Solving hQMLE using the Birthday-based Equivalence Finder

- Works **for any** $\mathbb{P} : U \to \{\top, \bot\}$ invariant under the equivalence ϕ
  - Crucial for performance: **density** of the predicate $\mathbb{P}$

$$d = |U_\top|/|U|, \quad U_\top = \{x \in U \mid \mathbb{P}(x) = \top\}$$

  - dN elements in $S_1(S_2)$ satisfying $\mathbb{P}$
  - we need lists of size $\ell = \sqrt{d \cdot N}$ for collision with success probability $1 - 1/e$
- Queries $\textsc{FindFunction}$ $d \cdot N$ times
- Performs $\ell/d = \sqrt{N/d}$ checks in $\textsc{SampleSet}$
- Best performance if blue and green balanced
  - when $d = N^{-1/3}$: $N^{\frac{2}{3}}$ checks in $\textsc{SampleSet}$ and $N^{\frac{2}{3}}$ queries to $\textsc{FindFunction}$

Solving MCE using the Birthday-based Equivalence Finder

Concrete complexity of solving MCE:

$$\max\left(\sqrt{q^{m+n}/d}\cdot C_{\mathbb{P}},\ dq^{m+n}\cdot C_{\mathsf{iQ}}\right)$$

Solving MCE using the Birthday-based Equivalence Finder

Concrete complexity of solving MCE:

$$\max(\sqrt{q^{m+n}/d} \cdot C_{\mathbb{P}}, dq^{m+n} \cdot C_{\mathrm{iQ}})$$

Asymptotic complexity of solving MCE:

$$\mathcal{O}(q^{\frac{2}{3}(n+m)} \cdot C_{\mathrm{iQ}}^{\frac{1}{3}})$$

(a perfect balance between the two steps of the algorithm)

*(success prob. $1 - 1/e$)

Matrix code equivalence: a cryptographic group action?

MCE as cryptographic group action.

$$\begin{aligned} \mu : \mathcal{C} &\to \mathcal{D} \\ \mathbf{C} &\mapsto \mathbf{ACB} \end{aligned}$$

- μ can be seen as element $(\mathbf{A}, \mathbf{B}) \in \mathrm{GL}_m(q) \times \mathrm{GL}_n(q)$

MCE as cryptographic group action.

$$\begin{aligned} \mu : \mathcal{C} &\to \mathcal{D} \\ \mathbf{C} &\mapsto \mathbf{ACB} \end{aligned}$$

- μ can be seen as element $(\mathbf{A}, \mathbf{B}) \in \mathrm{GL}_m(q) \times \mathrm{GL}_n(q)$
- μ acts on k-dimensional codes: $\mathcal{D} = \mu \cdot \mathcal{C}$

MCE as cryptographic group action.

$$\begin{aligned}\mu : \mathcal{C} &\to \mathcal{D} \\ \mathbf{C} &\mapsto \mathbf{ACB}\end{aligned}$$

- μ can be seen as element $(\mathbf{A}, \mathbf{B}) \in \mathrm{GL}_m(q) \times \mathrm{GL}_n(q)$
- μ acts on k-dimensional codes: $\mathcal{D} = \mu \cdot \mathcal{C}$
- hence, $\mathrm{GL}_m(q) \times \mathrm{GL}_n(q)$ acts on k-dimensional matrix codes $\mathcal{C} \subset \mathcal{M}_{m \times n}(\mathbb{F}_q)$.

MCE as cryptographic group action.

$$\begin{aligned}\mu : \mathcal{C} &\to \mathcal{D}\\ \mathbf{C} &\mapsto \mathbf{ACB}\end{aligned}$$

- μ can be seen as element $(\mathbf{A}, \mathbf{B}) \in \mathrm{GL}_m(q) \times \mathrm{GL}_n(q)$
- μ acts on k-dimensional codes: $\mathcal{D} = \mu \cdot \mathcal{C}$
- hence, $\mathrm{GL}_m(q) \times \mathrm{GL}_n(q)$ acts on k-dimensional matrix codes $\mathcal{C} \subset \mathcal{M}_{m\times n}(\mathbb{F}_q)$.
- one-way: our analysis show that MCE is hard.

Building crypto from group actions

Cryptographic Group Action: $G \times X \to X$

Given x_1 and x_2, it is hard to find an element g s.t. $x_2 = g \cdot x_1$

Building crypto from group actions

Cryptographic Group Action: $G \times X \to X$
Given x_1 and x_2, it is hard to find an element g s.t. $x_2 = g \cdot x_1$

What can we do with it?

Building crypto from group actions

Cryptographic Group Action: $G \times X \to X$
Given x_1 and x_2, it is hard to find an element g s.t. $x_2 = g \cdot x_1$

What can we do with it?

- **Zero-Knowledge Interactive Proof of knowledge**
  - Zero-Knowledgness
  - soundness
  - can be used as identification scheme (IDS)

Building crypto from group actions

Cryptographic Group Action: $G \times X \to X$
Given x_1 and x_2, it is hard to find an element g s.t. $x_2 = g \cdot x_1$

What can we do with it?

- **Zero-Knowledge Interactive Proof of knowledge**
  - Zero-Knowledgness
  - soundness
  - can be used as identification scheme (IDS)
- **Digital Signature via Fiat-Shamir transform**
  - F-S is a common strategy for PQ signatures
    - Dilithium, MQDSS, Picnic in NIST competition
  - From cryptographic group actions
    - Patarin's signature, LESS-FM, CSIDH, SeaSign ...

Zero-Knowledge Interactive Proof of knowledge from group actions

Let g be an element s.t. $x_1 = g \cdot x_0$.
Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it

$$\begin{tikzcd} x_0 \arrow[d, dashed, "g"'] \\ x_1 \end{tikzcd}$$

$\mathcal{P}(x_0, x_1, g)$	$\mathcal{V}(x_0, x_1)$

Zero-Knowledge Interactive Proof of knowledge from group actions

Let g be an element s.t. $x_1 = g \cdot x_0$.

Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it

$$\begin{tikzcd}
x_0 \arrow[r, "g_0"] \arrow[d, dashed, "g"'] & x' \\
x_1 &
\end{tikzcd}$$

$\mathcal{P}(x_0, x_1, g)$	$\mathcal{V}(x_0, x_1)$

Let g be an element s.t. $x_1 = g \cdot x_0$.
Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it

$$\begin{tikzcd} x_0 \arrow[r, "g_0"] \arrow[d, dashed, "g"'] & x' \\ x_1 \arrow[ur, "g_1"'] & \end{tikzcd}$$

$\mathcal{P}(x_0, x_1, g)$	$\mathcal{V}(x_0, x_1)$

Zero-Knowledge Interactive Proof of knowledge from group actions

Let g be an element s.t. $x_1 = g \cdot x_0$.
Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it

$$\begin{tikzcd} x_0 \arrow[d, dashed, "g"'] & x' \\ x_1 & \end{tikzcd}$$

$\mathcal{P}(x_0, x_1, g)$	$\mathcal{V}(x_0, x_1)$
$\mathrm{com} \leftarrow x'$	

Zero-Knowledge Interactive Proof of knowledge from group actions

Let g be an element s.t. $x_1 = g \cdot x_0$.

Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it

$$\begin{tikzcd} x_0 \arrow[d, dashed, "g"'] & & x' \\ x_1 & & \end{tikzcd}$$

$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
$\text{com} \leftarrow x'$	$\xrightarrow{\text{com}}$	

Let g be an element s.t. $x_1 = g \cdot x_0$.

Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it

$$\begin{tikzcd} x_0 \arrow[d, dashed, "g"'] & x' \\ x_1 & \end{tikzcd}$$

$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
$\mathsf{com} \leftarrow x'$	$\xrightarrow{\mathsf{com}}$	
		$\mathsf{ch} \leftarrow_R \{0, 1\}$

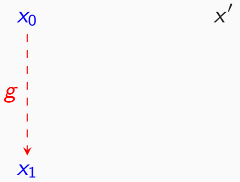
Let g be an element s.t. $x_1 = g \cdot x_0$.
Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it

$$\begin{tikzcd} x_0 \arrow[d, dashed, "g"'] & x' \\ x_1 & \end{tikzcd}$$

$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
com $\leftarrow x'$	$\xrightarrow{\text{com}}$	
	$\xleftarrow{\text{ch}}$	ch $\leftarrow_R \{0, 1\}$

Let g be an element s.t. $x_1 = g \cdot x_0$.

Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it



$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
$\mathrm{com} \leftarrow x'$	$\xrightarrow{\mathrm{com}}$	
	$\xleftarrow{\mathrm{ch}}$	$\mathrm{ch} \leftarrow_R \{0,1\}$
$\mathrm{resp} \leftarrow g_{\mathrm{ch}}$		

Zero-Knowledge Interactive Proof of knowledge from group actions

Let g be an element s.t. $x_1 = g \cdot x_0$.

Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it

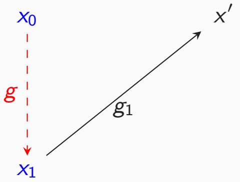
$$\begin{tikzcd} x_0 \arrow[r, "g_0"] \arrow[d, dashed, "g"'] & x' \\ x_1 & \end{tikzcd}$$

$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
$\mathsf{com} \leftarrow x'$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow_R \{0, 1\}$
$\mathsf{resp} \leftarrow g_{\mathsf{ch}}$	$\xrightarrow{\mathsf{resp}}$	

Zero-Knowledge Interactive Proof of knowledge from group actions

Let g be an element s.t. $x_1 = g \cdot x_0$.

Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it



$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
$\mathsf{com} \leftarrow x'$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow_R \{0,1\}$
$\mathsf{resp} \leftarrow g_{\mathsf{ch}}$	$\xrightarrow{\mathsf{resp}}$	

Zero-Knowledge Interactive Proof of knowledge from group actions

Let g be an element s.t. $x_1 = g \cdot x_0$.

Given x_0, x_1, the prover $\mathcal{P}$ wants to prove to the verifier $\mathcal{V}$ knowledge of g without revealing any information about it

$$\begin{tikzcd}
x_0 \arrow[r, "g_0"] \arrow[d, dashed, "g"'] & x' \\
x_1 \arrow[ur, "g_1"'] &
\end{tikzcd}$$

$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
$\mathsf{com} \leftarrow x'$	$\xrightarrow{\quad \mathsf{com} \quad}$	
	$\xleftarrow{\quad \mathsf{ch} \quad}$	$\mathsf{ch} \leftarrow_R \{0, 1\}$
$\mathsf{resp} \leftarrow g_{\mathsf{ch}}$	$\xrightarrow{\quad \mathsf{resp} \quad}$	
		$x' \stackrel{?}{=} g_{\mathsf{ch}} \cdot x_{\mathsf{ch}}$

Digital Signatures via the Fiat-Shamir transform

IDS

$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
$\mathsf{com} \leftarrow x'$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}}$	$\mathsf{ch} \leftarrow_R \{0,1\}$
$\mathsf{resp} \leftarrow g_{\mathsf{ch}}$	$\xrightarrow{\mathsf{resp}}$	
		$x' \stackrel{?}{=} g_{\mathsf{ch}} \cdot x_{\mathsf{ch}}$

Digital Signatures via the Fiat-Shamir transform

IDS

$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
$\mathsf{com} \leftarrow x', x'', \ldots, x^{(r)}$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch} = (\mathsf{ch}_1, \ldots, \mathsf{ch}_r)}$	$\mathsf{ch} \leftarrow_R \{0,1\}^r$
$\mathsf{resp} \leftarrow g_{\mathsf{ch}_1}, g_{\mathsf{ch}_2}, \ldots, g_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$x' \stackrel{?}{=} g_{\mathsf{ch}_1} \cdot x_{\mathsf{ch}_1}, \ldots, x^{(r)} \stackrel{?}{=} g_{\mathsf{ch}_r} \cdot x_{\mathsf{ch}_r}$

Digital Signatures via the Fiat-Shamir transform

IDS

$\mathcal{P}(x_0, x_1, g)$		$\mathcal{V}(x_0, x_1)$
$\mathsf{com} \leftarrow x', x'', \ldots, x^{(r)}$	$\xrightarrow{\mathsf{com}}$	
$\mathsf{ch} \leftarrow_R \{0,1\}^r$		
$\mathsf{resp} \leftarrow g_{\mathsf{ch}_1}, g_{\mathsf{ch}_2}, \ldots, g_{\mathsf{ch}_r}$	$\xrightarrow{\mathsf{resp}}$	
		$x' \stackrel{?}{=} g_{\mathsf{ch}_1} \cdot x_{\mathsf{ch}_1}, \ldots, x^{(r)} \stackrel{?}{=} g_{\mathsf{ch}_r} \cdot x_{\mathsf{ch}_r}$

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

FS signature

Signer(pk, sk)

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

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

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

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

Optimization techniques

The basic protocol is not very efficient

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

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

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

The basic protocol is not very efficient

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

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

- **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 efficient

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

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

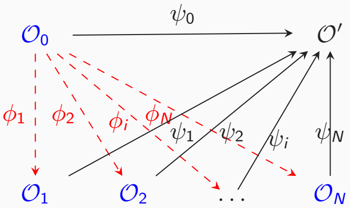
The basic protocol is not very efficient

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

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

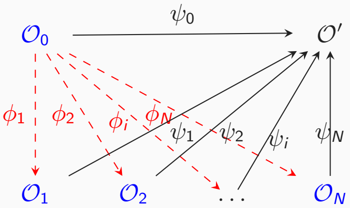
- **Challenge space is of size 2** $\Rightarrow$ Soundness error is $1/2$
- For security of λ bits, **needs to be repeated $r = \lambda$ times!**
- $\Rightarrow$ Signature contains λ isometries (from λ rounds)
- $\Rightarrow$ All operations in signing and verification need to be repeated λ times

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

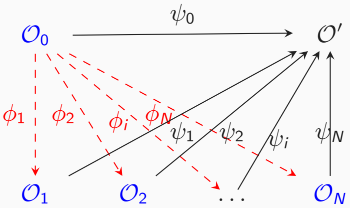
- **Challenge space is now of size** $N \Rightarrow$ Soundness error is $1/N$



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

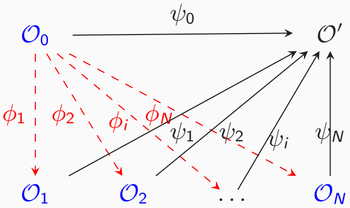
- **Challenge space is now of size** $N \Rightarrow$ Soundness error is $1/N$
- For security of λ bits, **needs to be repeated** $r = \frac{\lambda}{\log N}$ **times!**

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



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

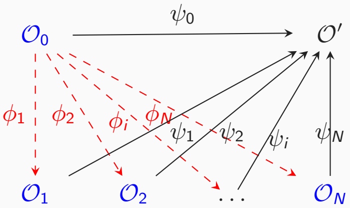
- **Challenge space is now of size N** $\Rightarrow$ Soundness error is $1/N$
- For security of λ bits, **needs to be repeated $r = \frac{\lambda}{\log N}$ times!**
- $\Rightarrow$ Signature contains $\frac{\lambda}{\log N}$ isometries



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

- **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

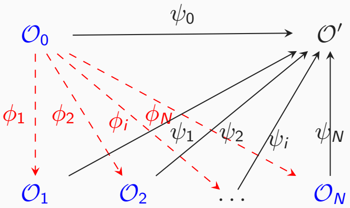
Optimization 1: Make the challenge space bigger (Multiple public keys)



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

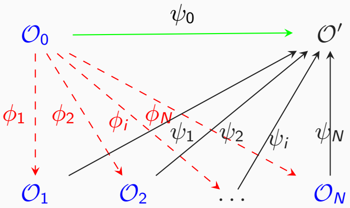
Optimization 1: Make the challenge space bigger (Multiple public keys)



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

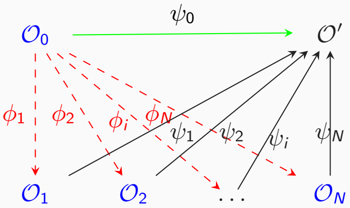
- **Challenge space is now of size** $N \Rightarrow$ Soundness error is $1/N$
- For security of λ bits, **needs to be repeated** $r = \frac{\lambda}{\log N}$ **times!**
- $\Rightarrow$ Signature contains $\frac{\lambda}{\log N}$ isometries
- $\Rightarrow$ All operations in signing and verification need to be repeated $\frac{\lambda}{\log N}$ times
- **There is a cost - N-fold increase in public and private key**
- 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 $\leftarrow \psi_{\text{ch}}$	$\xrightarrow{\text{resp}}$	
		$\mathcal{O}^{(i)} \stackrel{?}{=} \psi_{\text{ch}_i}(\mathcal{O}_{\text{ch}_i})$

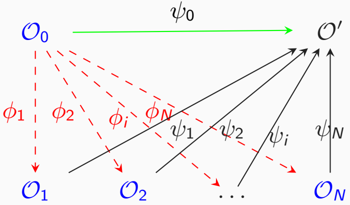
- **The map ψ_0 is chosen at random** $\Rightarrow$ one can include **only seed** in signature



$\mathcal{P}(\mathcal{O}_0, \dots, \mathcal{O}_N, \phi_1, \dots, \phi_N)$		$\mathcal{V}(\mathcal{O}_0, \dots, \mathcal{O}_N)$
com $\leftarrow \mathcal{O}', \mathcal{O}'', \dots, \mathcal{O}^{(r)}$	$\xrightarrow{\text{com}}$	
	$\xleftarrow{\text{ch} = (\text{ch}_1, \dots, \text{ch}_r)}$	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

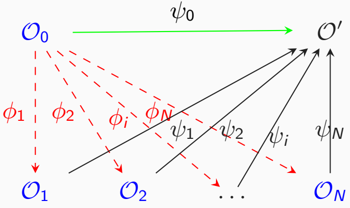
Optimization 2: Reduce signature size by using seeds



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

- **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 1:** This works only for challenge 0, and probability of choosing challenge 0 is $1/N$

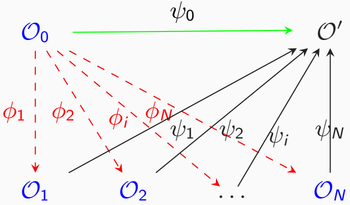
Optimization 2: Reduce signature size by using seeds



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

- **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 1:** This works only for challenge 0, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general

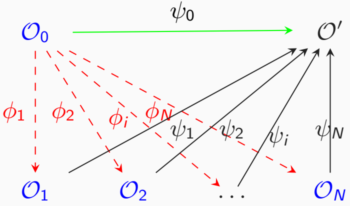
Optimization 2: Reduce signature size by using seeds



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

- **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 1:** This works only for challenge 0, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
- **Problem 2:** We don't even know that this is going to happen **exactly** $1/N$ of times

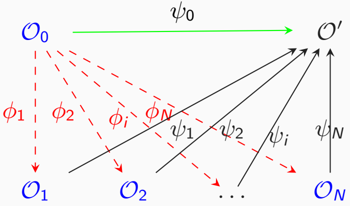
Optimization 2: Reduce signature size by using seeds



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

- **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 1:** This works only for challenge 0, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
- **Problem 2:** We don't even know that this is going to happen **exactly** $1/N$ of times
  - $\Rightarrow$ **signature is not of constant size**

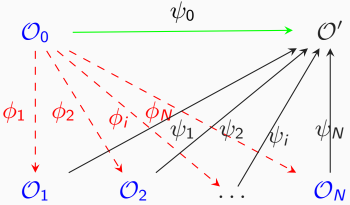
Optimization 2: Reduce signature size by using seeds



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

- **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 1:** This works only for challenge 0, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
- **Problem 2:** We don't even know that this is going to happen **exactly** $1/N$ of times
  - $\Rightarrow$ **signature is not of constant size**
- **Idea:** Always have a fixed number M of 0 challenges

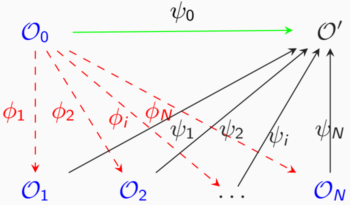
Optimization 2: Reduce signature size by using seeds



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

- **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 1:** This works only for challenge 0, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
- **Problem 2:** We don't even know that this is going to happen **exactly** $1/N$ of times
  - $\Rightarrow$ **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 outputs with fixed number of 0s

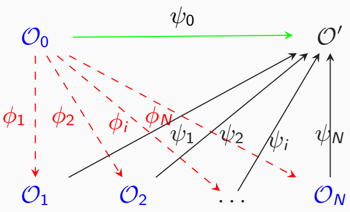
Optimization 2: Reduce signature size by using seeds



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

- **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 1:** This works only for challenge 0, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
- **Problem 2:** We don't even know that this is going to happen **exactly** $1/N$ of times
  - $\Rightarrow$ **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 outputs with fixed number of 0s

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

- **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 1:** This works only for challenge 0, and probability of choosing challenge 0 is $1/N$
  - $\Rightarrow$ not a big benefit in general
- **Problem 2:** We don't even know that this is going to happen **exactly** $1/N$ of times
  - $\Rightarrow$ **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 outputs with fixed number of 0s
- Always necessary to find the best trade-off

Matrix Code Equivalence as a cryptographic primitive!

(1) MCE is “easy to understand”

Matrix Code Equivalence as a cryptographic primitive!

(1) MCE is "easy to understand"

(2) Complexity linked to well-studied problem in multivariate crypto (IP)

Matrix Code Equivalence as a cryptographic primitive!

(1) MCE is “easy to understand”

(2) Complexity linked to well-studied problem in multivariate crypto (IP)

(3) Cryptographic group action: great building block!

Matrix Code Equivalence as a cryptographic primitive!

(1) MCE is "easy to understand"

(2) Complexity linked to well-studied problem in multivariate crypto (IP)

(3) Cryptographic group action: great building block!

(4) We construct a digital signature scheme

Matrix Code Equivalence as a cryptographic primitive!

(1) MCE is "easy to understand"

(2) Complexity linked to well-studied problem in multivariate crypto (IP)

(3) Cryptographic group action: great building block!

(4) We construct a digital signature scheme

(5) We construct (linkable) ring signatures

preprints:

- MCE hardness: https://eprint.iacr.org/2022/276.pdf
- MEDS: https://eprint.iacr.org/2022/1559.pdf