

# Improved algorithms of post-quantum cryptographic group actions

**Valerie Gilchrist**  
**Nov 25, 2025**



# Story time



# Story time



My social insurance  
number is...

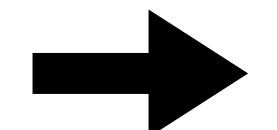


# Story time



# Story time

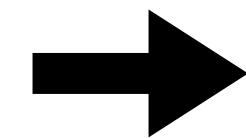
My social insurance  
number is...



0101001011101010  
1111110111010111  
1010100011100110  
0101001001111010  
1010010101101001

# Story time

My social insurance  
number is...

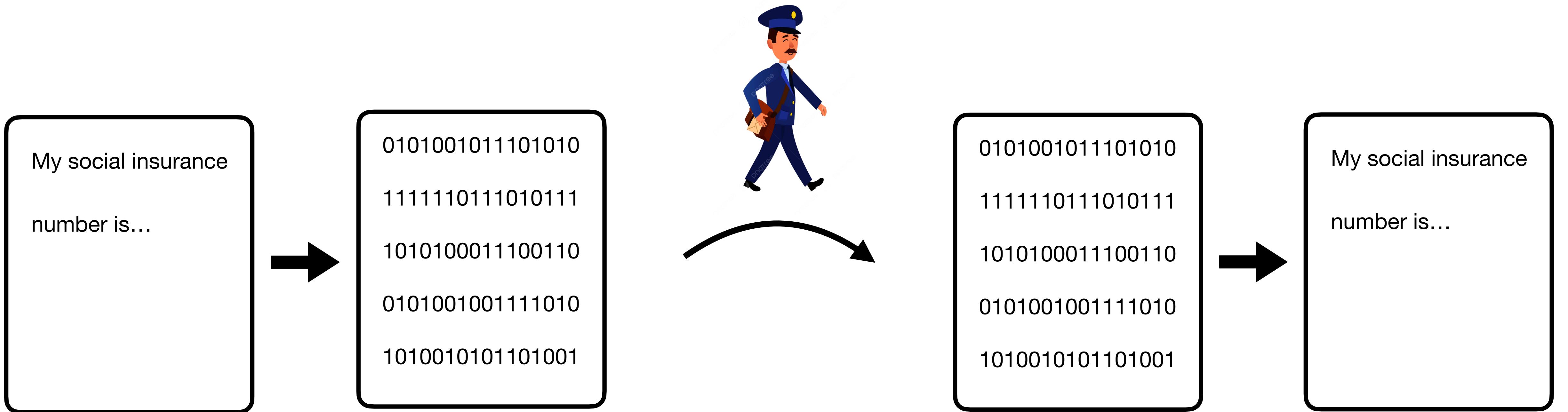


0101001011101010  
1111110111010111  
1010100011100110  
0101001001111010  
1010010101101001

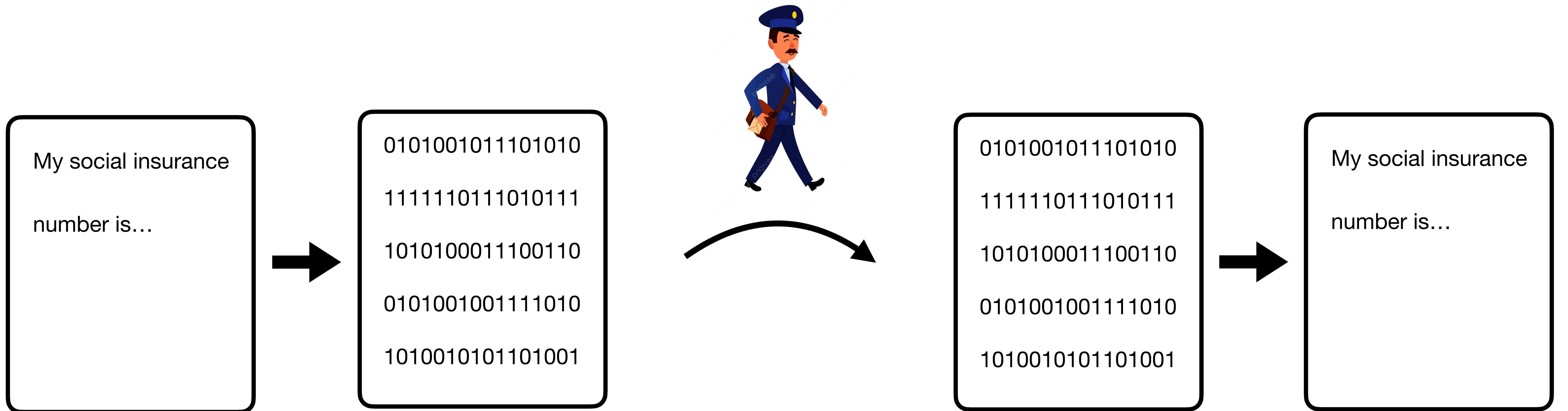


0101001011101010  
1111110111010111  
1010100011100110  
0101001001111010  
1010010101101001

# Story time

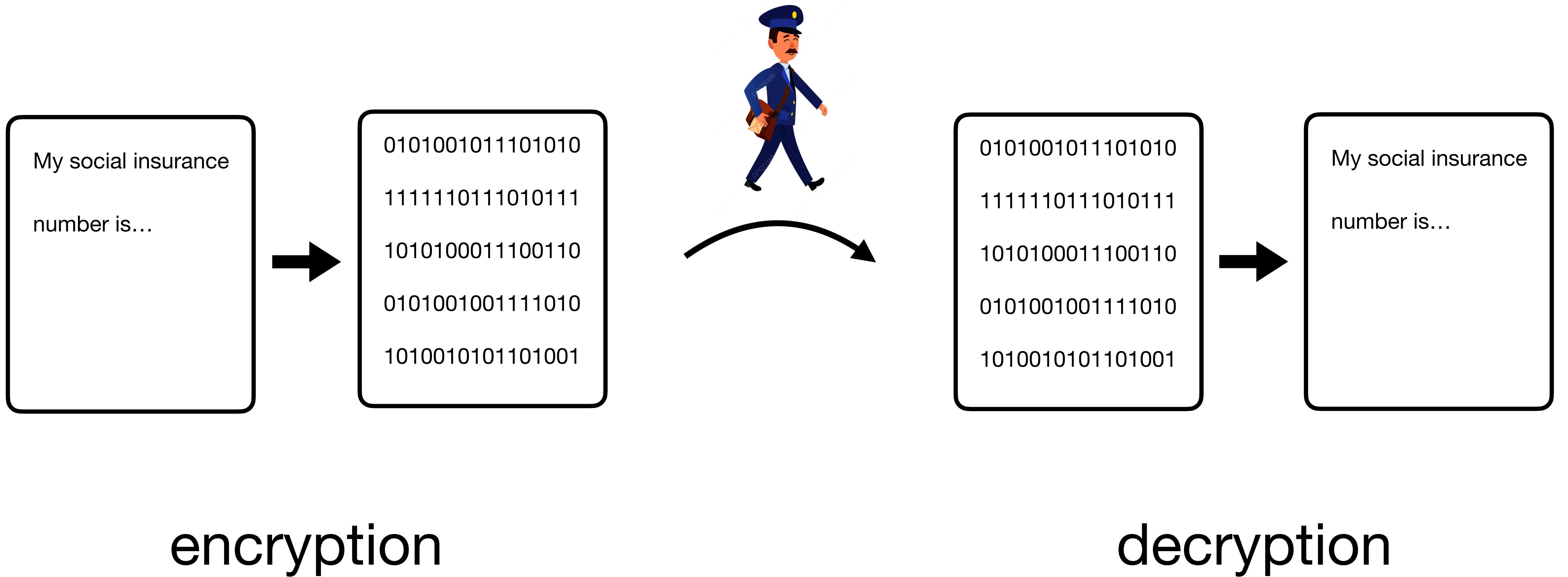


# Story time

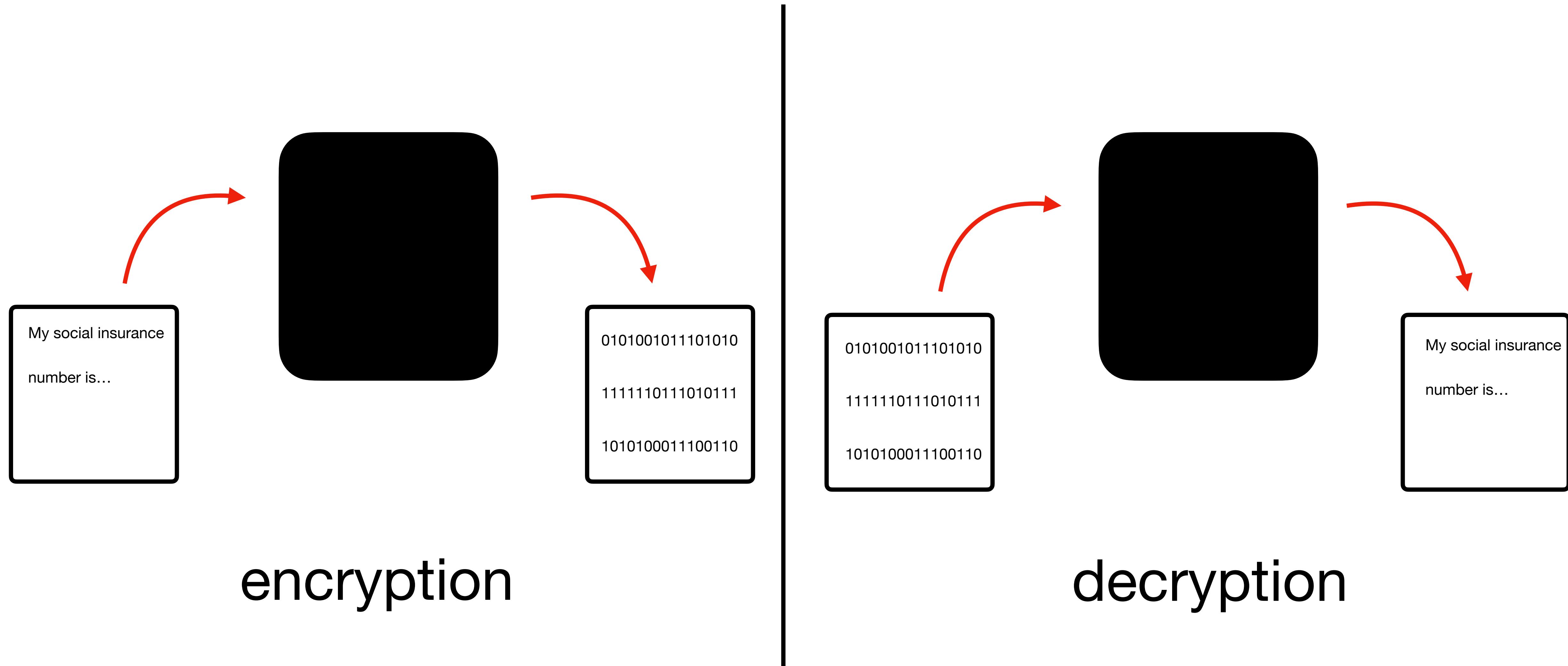


encryption

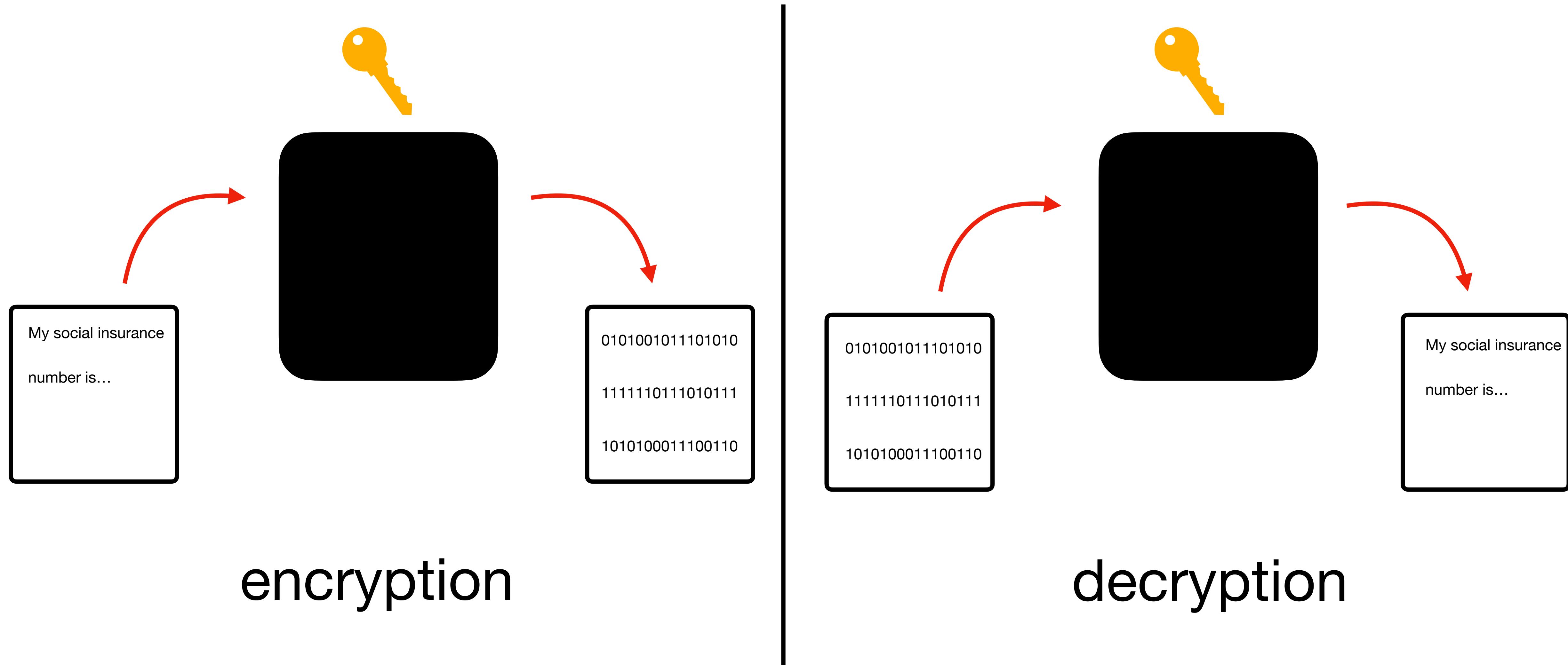
# Story time



# How can we encrypt/decrypt?



# How can we encrypt/decrypt?



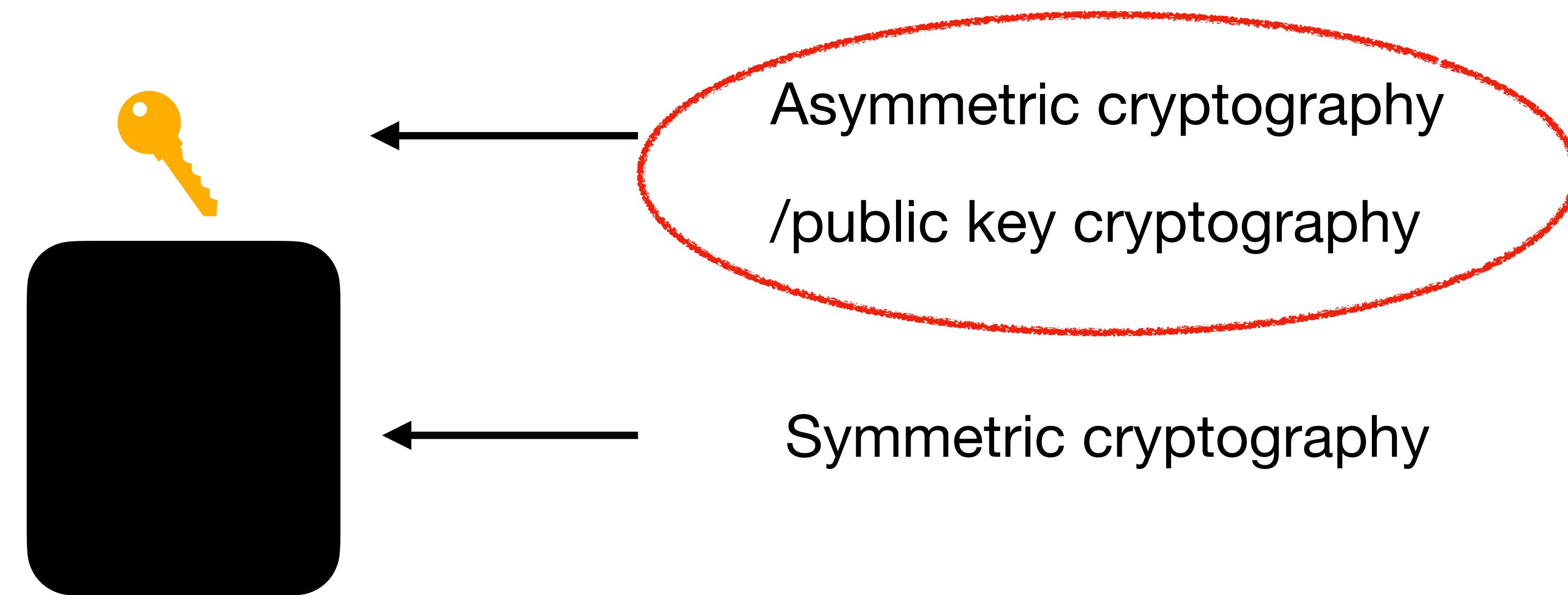
# How can we encrypt/decrypt?



Asymmetric cryptography  
/public key cryptography

Symmetric cryptography

# How can we encrypt/decrypt?



# Diffie-Hellman key exchange



$$G = \langle g \rangle$$

# Diffie-Hellman key exchange



$a \in \mathbb{Z}$



$$G = \langle g \rangle$$



$b \in \mathbb{Z}$

# Diffie-Hellman key exchange



$a \in \mathbb{Z}$

$g^a$



$$G = \langle g \rangle$$

$b \in \mathbb{Z}$



# Diffie-Hellman key exchange



$a \in \mathbb{Z}$

$g^a$



$$G = \langle g \rangle$$

$g^a$



$b \in \mathbb{Z}$

# Diffie-Hellman key exchange



$a \in \mathbb{Z}$

$g^a$



$$G = \langle g \rangle$$

$g^a$



$b \in \mathbb{Z}$

$g^a$

# Diffie-Hellman key exchange



$$a \in \mathbb{Z}$$

$$g^a$$

$$g^b$$



$$G = \langle g \rangle$$

$$g^a \quad g^b$$



$$b \in \mathbb{Z}$$

$$g^b$$

$$g^a$$

# Diffie-Hellman key exchange



$$a \in \mathbb{Z}$$

$$g^a$$

$$g^b$$

$$(g^b)^a$$



$$G = \langle g \rangle$$

$$g^a \quad g^b$$



$$b \in \mathbb{Z}$$

$$g^b$$

$$g^a$$

# Diffie-Hellman key exchange



$$a \in \mathbb{Z}$$

$$g^a$$

$$g^b$$

$$(g^b)^a$$



$$G = \langle g \rangle$$

$$g^a \quad g^b$$



$$b \in \mathbb{Z}$$

$$g^b$$

$$g^a$$

$$(g^a)^b$$

# Diffie-Hellman key exchange



$$a \in \mathbb{Z}$$

$$g^a$$

$$g^b$$

$$(g^b)^a$$



$$G = \langle g \rangle$$

$$g^a \quad g^b$$



$$b \in \mathbb{Z}$$

$$g^b$$

$$g^a$$

$$(g^a)^b$$

# Diffie-Hellman key exchange

Discrete Log

Problem :  $G = \langle g \rangle$ . Given  $g, g^a$ , recover  $a$ .

Computational Diffie-Hellman

Problem :  $G = \langle g \rangle$ . Given  $g, g^a, g^b$ , recover  $g^{ab}$ .

# Diffie-Hellman key exchange

Discrete Log

Problem :  $G = \langle g \rangle$ . Given  $g, g^a$ , recover  $a$ .

Computational Diffie-Hellman

Problem :  $G = \langle g \rangle$ . Given  $g, g^a, g^b$ , recover  $g^{ab}$ .

→ hard for computers (and mailmen!)

# Diffie-Hellman key exchange

Discrete Log

Problem :  $G = \langle g \rangle$ . Given  $g, g^a$ , recover  $a$ .

Computational Diffie-Hellman

Problem :  $G = \langle g \rangle$ . Given  $g, g^a, g^b$ , recover  $g^{ab}$ .

→ hard for computers (and mailmen!)

...NOT HARD for quantum computers

# Group actions

Let  $G$  be a group and  $X$  a set

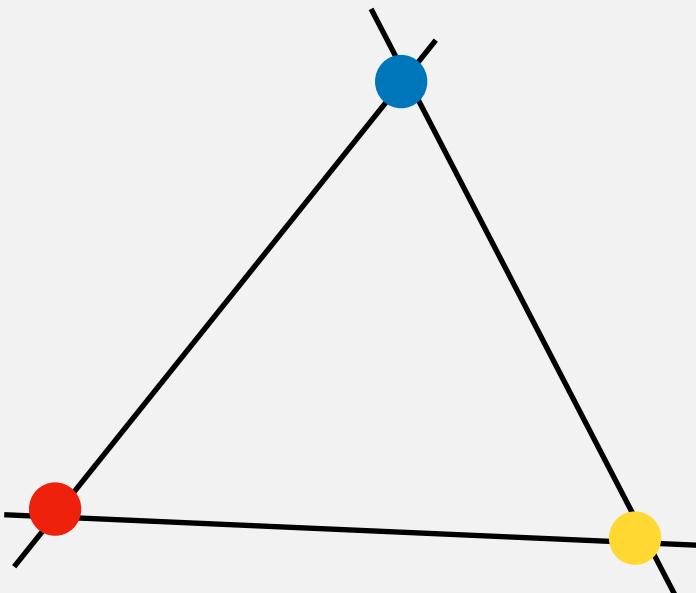
A *group action* is a map  $\star : G \times X \rightarrow X$  that satisfies

1. *Compatibility* :  $\forall g_1, g_2 \in G, x \in X, \quad g_1 \star (g_2 \star x) = g_1 g_2 \star x$
2. *Identity* :  $\exists e \in G : \forall x \in X, \quad e \star x = x$

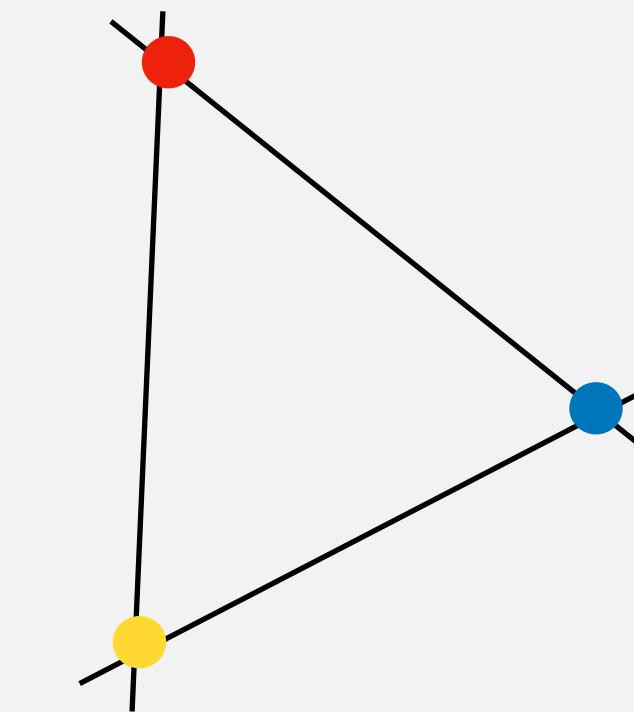
# Group actions

e.g.  $\star : (\text{rotations}) \times (\text{triangles}) \rightarrow (\text{triangles})$

$90^\circ \text{ CW}$



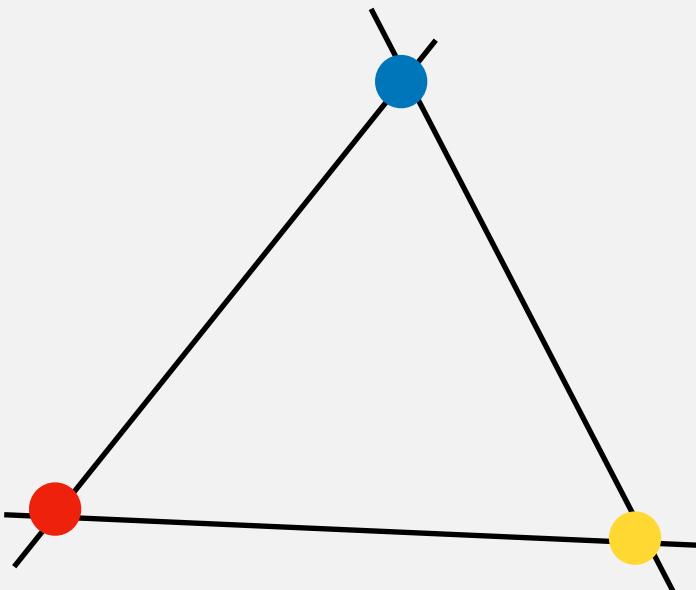
=



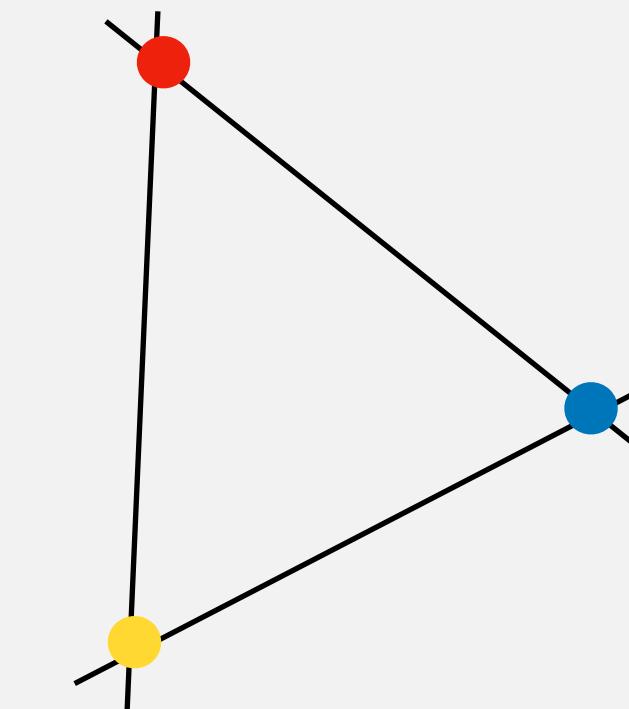
# Group actions

e.g.  $\star : (\text{rotations}) \times (\text{triangles}) \rightarrow (\text{triangles})$

$90^\circ \text{ CW}$



=



-identity:  $0^\circ \text{ CW}$

-compatibility:  $X^\circ \star (Y^\circ \star \triangle) = (X^\circ + Y^\circ) \star \triangle$

# Group actions

Discrete Log

Problem :  $G = \langle g \rangle$ . Given  $g, g^a$ , recover  $a$ .

Computational Diffie-Hellman

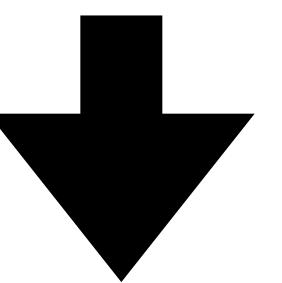
Problem :  $G = \langle g \rangle$ . Given  $g, g^a, g^b$ , recover  $g^{ab}$ .

# Group actions

Discrete Log

Problem :  $G = \langle g \rangle$ . Given  $g, g^a$ , recover  $a$ .

Computational Diffie-Hellman



Vectorization

Problem :  $(\star, G, X)$ . Given  $x, g \star x$ , recover  $g$ .

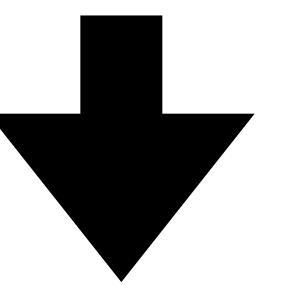
Problem :  $G = \langle g \rangle$ . Given  $g, g^a, g^b$ , recover  $g^{ab}$ .

# Group actions

Discrete Log

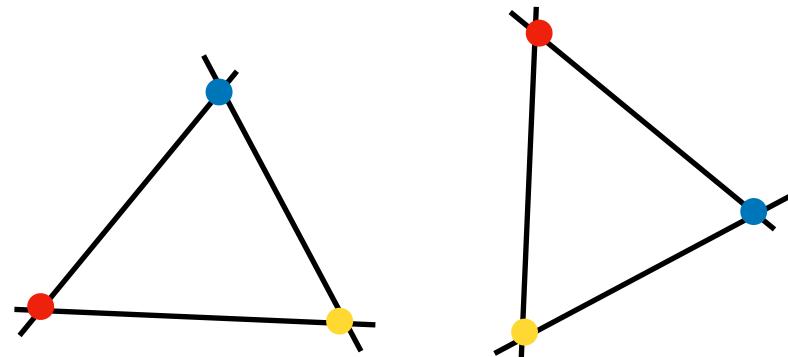
Problem :  $G = \langle g \rangle$ . Given  $g, g^a$ , recover  $a$ .

Computational Diffie-Hellman



Vectorization

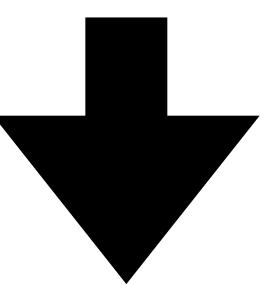
Problem :  $(\star, G, X)$ . Given  $x, g \star x$ , recover  $g$ .



# Group actions

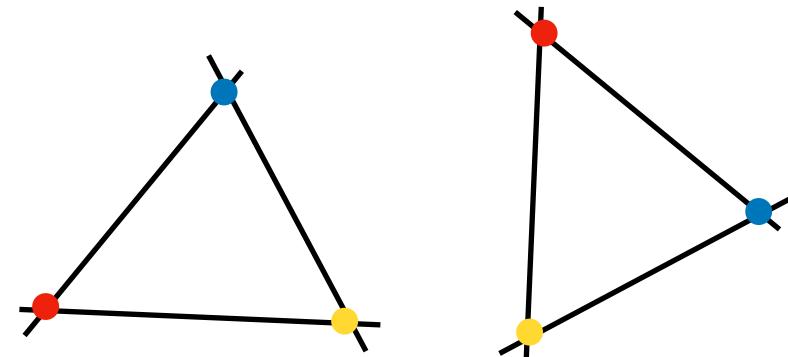
Discrete Log

Problem :  $G = \langle g \rangle$ . Given  $g, g^a$ , recover  $a$ .



Vectorization

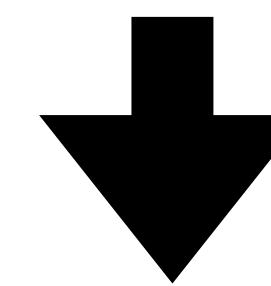
Problem :  $(\star, G, X)$ . Given  $x, g \star x$ , recover  $g$ .



90° CW

Computational Diffie-Hellman

Problem :  $G = \langle g \rangle$ . Given  $g, g^a, g^b$ , recover  $g^{ab}$ .



Parallelization

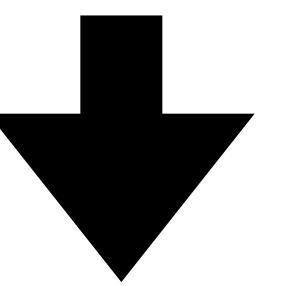
Problem :  $(\star, G, X)$ .

Given  $x, g_A \star x, g_B \star x$ , recover  $g_A g_B \star x$ .

# Group actions

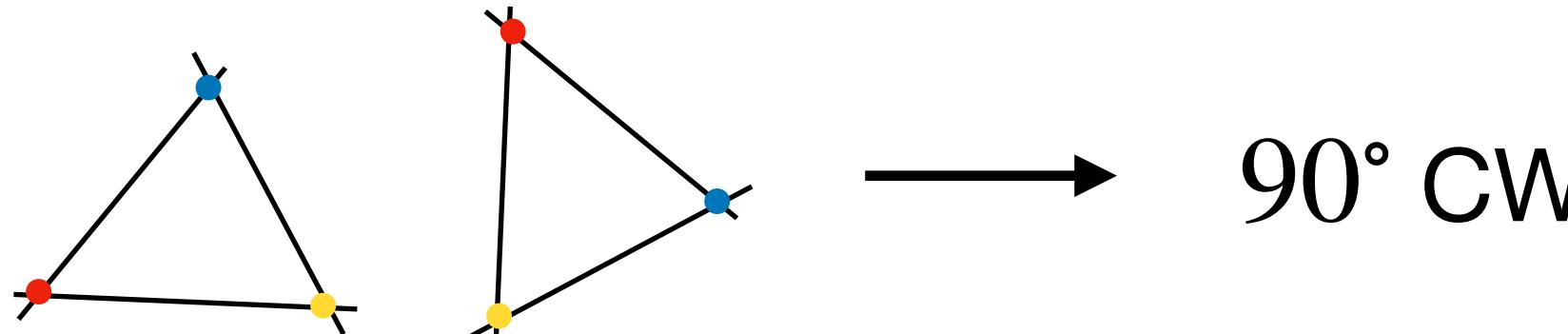
Discrete Log

Problem :  $G = \langle g \rangle$ . Given  $g, g^a$ , recover  $a$ .



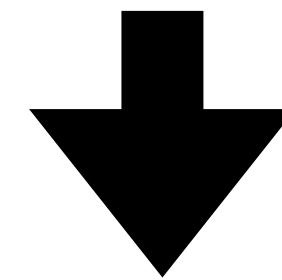
Vectorization

Problem :  $(\star, G, X)$ . Given  $x, g \star x$ , recover  $g$ .



Computational Diffie-Hellman

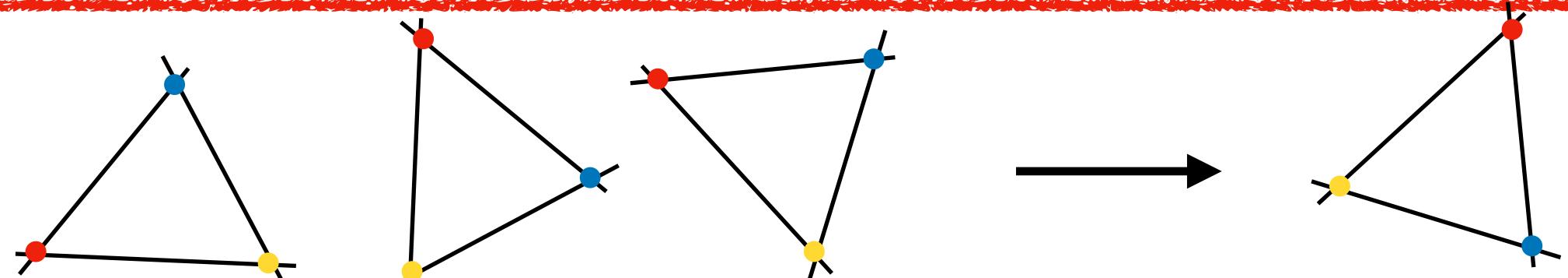
Problem :  $G = \langle g \rangle$ . Given  $g, g^a, g^b$ , recover  $g^{ab}$ .



Parallelization

Problem :  $(\star, G, X)$ .

Given  $x, g_A \star x, g_B \star x$ , recover  $g_A g_B \star x$ .



# Group actions

Let  $\star : G \times X \rightarrow X$  be a (commutative) group action,  $x \in X$

**Alice**

(secret)  $g_a \in G$

**Bob**

(secret)  $g_b \in G$

# Group actions

Let  $\star : G \times X \rightarrow X$  be a (commutative) group action,  $x \in X$

**Alice**

(secret)  $g_a \in G$

$g_a \star x$

**Bob**

(secret)  $g_b \in G$

# Group actions

Let  $\star : G \times X \rightarrow X$  be a (commutative) group action,  $x \in X$

**Alice**

(secret)  $g_a \in G$

$g_a \star x$

**Bob**

(secret)  $g_b \in G$

$g_b \star x$

# Group actions

Let  $\star : G \times X \rightarrow X$  be a (commutative) group action,  $x \in X$

**Alice**

(secret)  $g_a \in G$

$g_a \star x$

**Bob**

(secret)  $g_b \in G$

$g_b \star x$

$g_b \star x$

$g_a \star x$

# Group actions

Let  $\star : G \times X \rightarrow X$  be a (commutative) group action,  $x \in X$

**Alice**

(secret)  $g_a \in G$

$g_a \star x$

$g_b \star x$

$g_a g_b \star x$

**Bob**

(secret)  $g_b \in G$

$g_b \star x$

$g_a \star x$

# Group actions

Let  $\star : G \times X \rightarrow X$  be a (commutative) group action,  $x \in X$

**Alice**

(secret)  $g_a \in G$

$$g_a \star x$$

$$g_b \star x$$

$$g_a g_b \star x$$

**Bob**

(secret)  $g_b \in G$

$$g_b \star x$$

$$g_a \star x$$

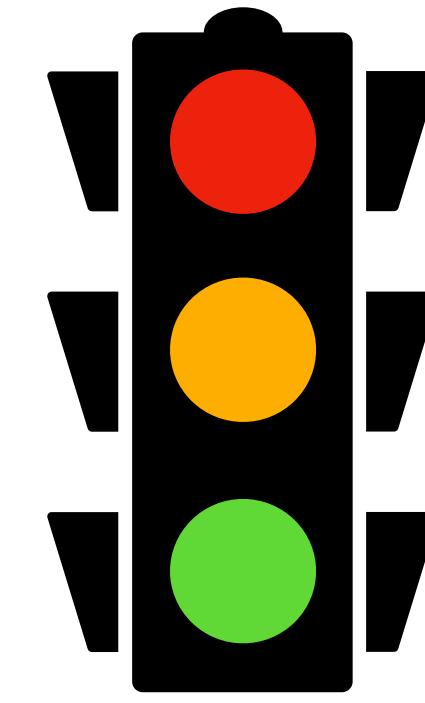
$$g_b g_a \star x$$

# Contributions

# Contributions

group action + an instance from the literature + our contribution

# Contributions



group action + an instance from the literature + our contribution

# Contribution 1

[In submission]

## On the security of two blind signatures from code equivalence problems

Valerie Gilchrist<sup>1</sup> , Laurane Marco<sup>2</sup> , Christophe Petit<sup>1,3</sup>  and Gang Tang<sup>3</sup>

<sup>1</sup> Université Libre de Bruxelles, Belgium

<sup>2</sup> EPFL, Switzerland

<sup>3</sup> University of Birmingham, United Kingdom

**Abstract.** The Linear Code Equivalence (LCE) problem and the Matrix Code Equivalence (MCE) problem are two examples of code-based hard problems that have gained attention as candidates for use in post-quantum cryptography. They are straightforward to implement, can be viewed as group actions, and offer a good trade-off between compactness and performance in the realm of post-quantum group actions.

With the community gaining confidence in the security of these problems, new variants of them will need to be introduced to maintain the functionality of the existing ones.

# Linear code group action

An  $[n, m]$  linear code  $\mathcal{C}$  is an  $m$ -dimensional linear subspace of  $\mathbb{F}_q^n$

# Linear code group action

An  $[n, m]$  linear code  $\mathcal{C}$  is an  $m$ -dimensional linear subspace of  $\mathbb{F}_q^n$

e.g. Consider  $\mathbb{F}_{101}^4$

$$\mathcal{C} = \langle [1,0,0,0], [0,1,0,0] \rangle$$

# Linear code group action

An  $[n, m]$  linear code  $\mathcal{C}$  is an  $m$ -dimensional linear subspace of  $\mathbb{F}_q^n$

e.g. Consider  $\mathbb{F}_{101}^4$

$$\mathcal{C} = \langle [1,0,0,0], [0,1,0,0] \rangle = \{ [a,b,0,0] : a, b \in \mathbb{F}_{101} \}$$

# Linear code group action

An  $[n, m]$  linear code  $\mathcal{C}$  is an  $m$ -dimensional linear subspace of  $\mathbb{F}_q^n$

e.g. Consider  $\mathbb{F}_{101}^4$

$$\mathcal{C} = \langle [1,0,0,0], [0,1,0,0] \rangle = \{ [a, b, 0, 0] : a, b \in \mathbb{F}_{101} \}$$

$$= \{ [a \ b] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} : [a \ b] \in \mathbb{F}_{101}^2 \}$$

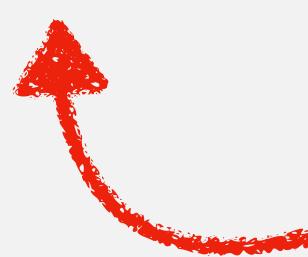
# Linear code group action

An  $[n, m]$  linear code  $\mathcal{C}$  is an  $m$ -dimensional linear subspace of  $\mathbb{F}_q^n$

e.g. Consider  $\mathbb{F}_{101}^4$

$$\mathcal{C} = \langle [1,0,0,0], [0,1,0,0] \rangle = \{ [a, b, 0, 0] : a, b \in \mathbb{F}_{101} \}$$

$$= \{ [a \ b] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} : [a \ b] \in \mathbb{F}_{101}^2 \}$$



*Generator matrix*

# Linear code group action

Let  $G$  be the generator matrix of a code

e.g.  $G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

# Linear code group action

Let  $G$  be the generator matrix of a code

e.g.  $G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

For  $A$  invertible,  $D$  diagonal,  $P$  permutation, the following is a group action:

$$(A, DP) \star G = AGDP$$

# Linear code group action (contribution 1)

[DKQ+25] Blind Signatures from Cryptographic Group Actions. Duong, Khuc, Qiao, Susilo, Zhang. Eprint 2025.

**Problem :**

Given generator matrices  $G_0, G_1, G_2$  such that

$$G_0 \xrightarrow{\star (A_0, D_0 P)} G_1 \xrightarrow{\star (A_1, D_1 P^{-1})} G_2$$

( $D_i$  diagonal,  $P$  permutation), compute  $P$ .

# Linear code group action (contribution 1)

[DKQ+25] Blind Signatures from Cryptographic Group Actions. Duong, Khuc, Qiao, Susilo, Zhang. Eprint 2025.

**Problem :**

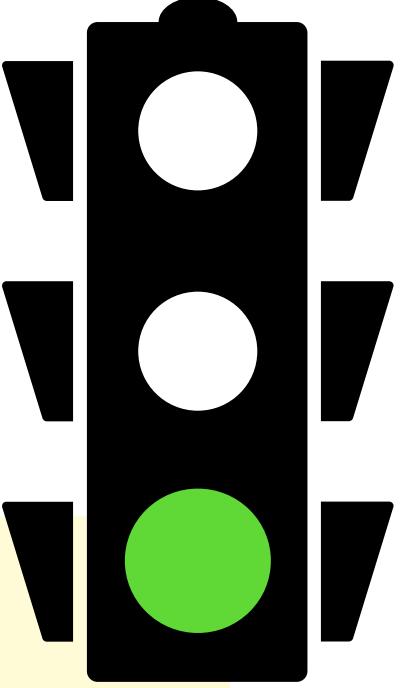
Given generator matrices  $G_0, G_1, G_2$  such that

$$G_0 \xrightarrow{\star (A_0, D_0 P)} G_1 \xrightarrow{\star (A_1, D_1 P^{-1})} G_2$$

( $D_i$  diagonal,  $P$  permutation), compute  $P$ .

-we gave a polynomial time reduction between this problem and pure vectorization

# Linear code group action (contribution 1)



[DKQ+25] Blind Signatures from Cryptographic Group Actions. Duong, Khuc, Qiao, Susilo, Zhang. Eprint 2025.

**Problem :**

Given generator matrices  $G_0, G_1, G_2$  such that

$$G_0 \xrightarrow{\star (A_0, D_0 P)} G_1 \xrightarrow{\star (A_1, D_1 P^{-1})} G_2$$

( $D_i$  diagonal,  $P$  permutation), compute  $P$ .

-we gave a polynomial time reduction between this problem and pure vectorization

# Contribution 2

[In submission]

## On the security of two blind signatures from code equivalence problems

Valerie Gilchrist<sup>1</sup> , Laurane Marco<sup>2</sup> , Christophe Petit<sup>1,3</sup>  and Gang Tang<sup>3</sup>

<sup>1</sup> Université Libre de Bruxelles, Belgium

<sup>2</sup> EPFL, Switzerland

<sup>3</sup> University of Birmingham, United Kingdom

**Abstract.** The Linear Code Equivalence (LCE) problem and the Matrix Code Equivalence (MCE) problem are two examples of code-based hard problems that have gained attention as candidates for use in post-quantum cryptography. They are straightforward to implement, can be viewed as group actions, and offer a good trade-off between compactness and performance in the realm of post-quantum group actions.

With the community gaining confidence in the security of these problems, new variants of them will need to be introduced to maintain a range of options for future applications.

# Matrix code group action

A  $[m \times n, k]$  matrix code  $\mathcal{C}$  is a  $k$ -dimensional  $\mathbb{F}_q$ -linear subspace of  $\mathbb{F}_q^{m \times n}$

# Matrix code group action

A  $[m \times n, k]$  matrix code  $\mathcal{C}$  is a  $k$ -dimensional  $\mathbb{F}_q$ -linear subspace of  $\mathbb{F}_q^{m \times n}$

e.g. Consider  $\mathbb{F}_{101}^{2 \times 2}$

$$\mathcal{C} = \left\langle \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\rangle$$

# Matrix code group action

A  $[m \times n, k]$  matrix code  $\mathcal{C}$  is a  $k$ -dimensional  $\mathbb{F}_q$ -linear subspace of  $\mathbb{F}_q^{m \times n}$

e.g. Consider  $\mathbb{F}_{101}^{2 \times 2}$

$$\mathcal{C} = \left\langle \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\rangle = \left\{ \begin{bmatrix} a & b \\ b & 0 \end{bmatrix} : a, b \in \mathbb{F}_{101} \right\}$$

# Matrix code group action

A  $[m \times n, k]$  matrix code  $\mathcal{C}$  is a  $k$ -dimensional  $\mathbb{F}_q$ -linear subspace of  $\mathbb{F}_q^{m \times n}$

e.g. Consider  $\mathbb{F}_{101}^{2 \times 2}$

$$\mathcal{C} = \left\langle \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\rangle = \left\{ \begin{bmatrix} a & b \\ b & 0 \end{bmatrix} : a, b \in \mathbb{F}_{101} \right\}$$

$$A\mathcal{C} = \left\{ A \begin{bmatrix} a & b \\ b & 0 \end{bmatrix} : a, b \in \mathbb{F}_{101} \right\},$$

# Matrix code group action

A  $[m \times n, k]$  matrix code  $\mathcal{C}$  is a  $k$ -dimensional  $\mathbb{F}_q$ -linear subspace of  $\mathbb{F}_q^{m \times n}$

e.g. Consider  $\mathbb{F}_{101}^{2 \times 2}$

$$\mathcal{C} = \left\langle \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\rangle = \left\{ \begin{bmatrix} a & b \\ b & 0 \end{bmatrix} : a, b \in \mathbb{F}_{101} \right\}$$

$$A\mathcal{C} = \left\{ A \begin{bmatrix} a & b \\ b & 0 \end{bmatrix} : a, b \in \mathbb{F}_{101} \right\}, \quad \mathcal{C}B = \left\{ \begin{bmatrix} a & b \\ b & 0 \end{bmatrix} B : a, b \in \mathbb{F}_{101} \right\}$$

# Matrix code group action

Let  $A, B$  be invertible, then the following is a group action:

# Matrix code group action

Let  $A, B$  be invertible, then the following is a group action:

$$(A, B) \star \mathcal{C} = A\mathcal{C}B$$

# Matrix code group action (contribution 2)

[KLP25] Post-Quantum Blind Signatures from Matrix Code Equivalence. Kuchta, LeGrow, Persichetti. Eprint 2025.

**Problem :** Given matrix codes  $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2$ , such that

$$\mathcal{C}_2 \xleftarrow{\star(A^{-1}, B^{-1})} \mathcal{C}_0 \xrightarrow{\star(A, B)} \mathcal{C}_1$$

for  $A, B$  (anti)symmetric matrices... compute  $A, B$ .

# Matrix code group action (contribution 2)

[KLP25] Post-Quantum Blind Signatures from Matrix Code Equivalence. Kuchta, LeGrow, Persichetti. Eprint 2025.

**Problem :** Given matrix codes  $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2$ , such that

$$\mathcal{C}_2 \xleftarrow{\star(A^{-1}, B^{-1})} \mathcal{C}_0 \xrightarrow{\star(A, B)} \mathcal{C}_1$$

for  $A, B$  (anti)symmetric matrices... compute  $A, B$ .

-we showed that repeating the same secret key even once would leak the secret

# Matrix code group action (contribution 2)

[KLP25] Post-Quantum Blind Signatures from Matrix Code Equivalence. Kuchta, LeGrow, Persichetti. Eprint 2025.

**Problem :** Given matrix codes  $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2$ , such that

$$\mathcal{C}_2 \xleftarrow{\star (A^{-1}, B^{-1})} \mathcal{C}_0 \xrightarrow{\star (A, B)} \mathcal{C}_1$$

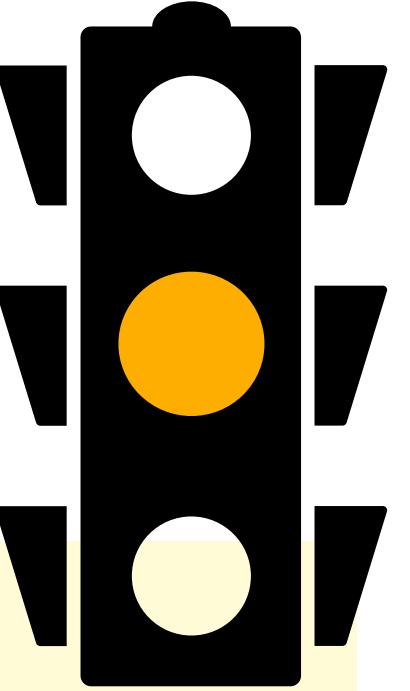
for  $A, B$  (anti)symmetric matrices... compute  $A, B$ .

$$\mathcal{D}_2 \xleftarrow{\star (A^{-1}, B^{-1})} \mathcal{D}_0 \xrightarrow{\star (A, B)} \mathcal{D}_1$$

-we showed that repeating the same secret key even once would leak the secret

# Matrix code group action (contribution 2)

[KLP25] Post-Quantum Blind Signatures from Matrix Code Equivalence. Kuchta, LeGrow, Persichetti. Eprint 2025.



**Problem :** Given matrix codes  $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2$ , such that

$$\mathcal{C}_2 \xleftarrow{\star (A^{-1}, B^{-1})} \mathcal{C}_0 \xrightarrow{\star (A, B)} \mathcal{C}_1$$

for  $A, B$  (anti)symmetric matrices... compute  $A, B$ .

$$\mathcal{D}_2 \xleftarrow{\star (A^{-1}, B^{-1})} \mathcal{D}_0 \xrightarrow{\star (A, B)} \mathcal{D}_1$$

-we showed that repeating the same secret key even once would leak the secret

# Contribution 3

[CRYPTO 2024 ]

## Solving the Tensor Isomorphism Problem for special orbits with low rank points: Cryptanalysis and repair of an Asiacrypt 2023 commitment scheme

Valerie Gilchrist<sup>1</sup>, Laurane Marco<sup>2</sup>, Christophe Petit<sup>1,3</sup>, Gang Tang<sup>4,3</sup>

<sup>1</sup> Université Libre de Bruxelles, Brussels, Belgium

<sup>2</sup> EPFL, Lausanne, Switzerland

<sup>3</sup> University of Birmingham, Birmingham, United Kingdom

<sup>4</sup> University of Technology Sydney, NSW, Australia

**Abstract.** The Tensor Isomorphism Problem (TIP) has been shown equivalent to the matrix code equivalence problem, making it an interesting candidate on which to build post-quantum cryptographic primitives. These hard problems have already been used in protocol development. One of these, MEDS, is currently in Round 1 of NIST's call for additional

# Tensor group action

# Tensor group action

We would like to compute the tensor  $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$  where  $\mathbf{u} = [u_1, u_2]$ ,  $\mathbf{v} = [v_1, v_2]$ ,  $\mathbf{w} = [w_1, w_2]$ .

# Tensor group action

We would like to compute the tensor  $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$  where  $\mathbf{u} = [u_1, u_2]$ ,  $\mathbf{v} = [v_1, v_2]$ ,  $\mathbf{w} = [w_1, w_2]$ .

We proceed by first expanding the matrix  $\mathbf{v} \cdot \mathbf{w}^T$ :

$$\mathbf{v} \cdot \mathbf{w}^T = \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}$$

# Tensor group action

We would like to compute the tensor  $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$  where  $\mathbf{u} = [u_1, u_2]$ ,  $\mathbf{v} = [v_1, v_2]$ ,  $\mathbf{w} = [w_1, w_2]$ .

We proceed by first expanding the matrix  $\mathbf{v} \cdot \mathbf{w}^T$ :

$$\mathbf{v} \cdot \mathbf{w}^T = \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}$$

Now we multiply this matrix by each entry of  $\mathbf{u}$ , storing them in a list as we go:

$$u_1 \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}, u_2 \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}$$

# Tensor group action

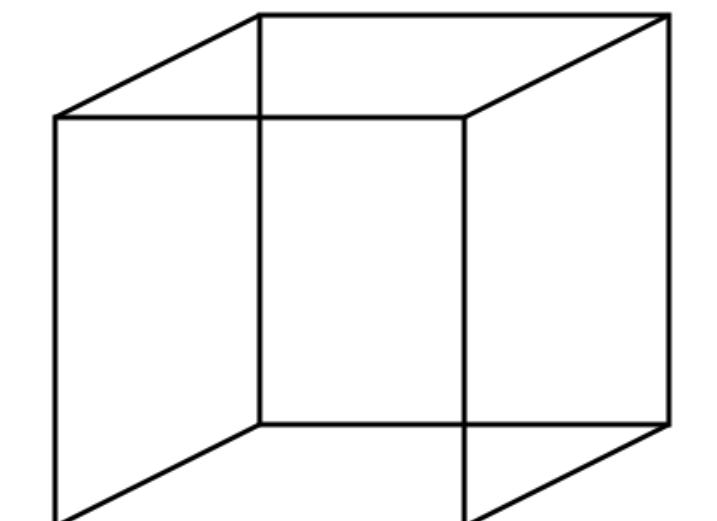
We would like to compute the tensor  $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$  where  $\mathbf{u} = [u_1, u_2]$ ,  $\mathbf{v} = [v_1, v_2]$ ,  $\mathbf{w} = [w_1, w_2]$ .

We proceed by first expanding the matrix  $\mathbf{v} \cdot \mathbf{w}^T$ :

$$\mathbf{v} \cdot \mathbf{w}^T = \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}$$

Now we multiply this matrix by each entry of  $\mathbf{u}$ , storing them in a list as we go:

$$u_1 \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}, u_2 \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v_2 w_2 \end{bmatrix}$$



# Tensor group action

Let  $\mathbf{V}$  be the tensor space  $\sum_i u_i \otimes v_i \otimes w_i$

# Tensor group action

Let  $\mathbf{V}$  be the tensor space  $\sum_i u_i \otimes v_i \otimes w_i = \sum_{ijl} v(i,j,l) e_i \otimes e_j \otimes e_l$

# Tensor group action

Let  $\mathbf{V}$  be the tensor space  $\sum_i u_i \otimes v_i \otimes w_i = \sum_{ijl} v(i,j,l) e_i \otimes e_j \otimes e_l$

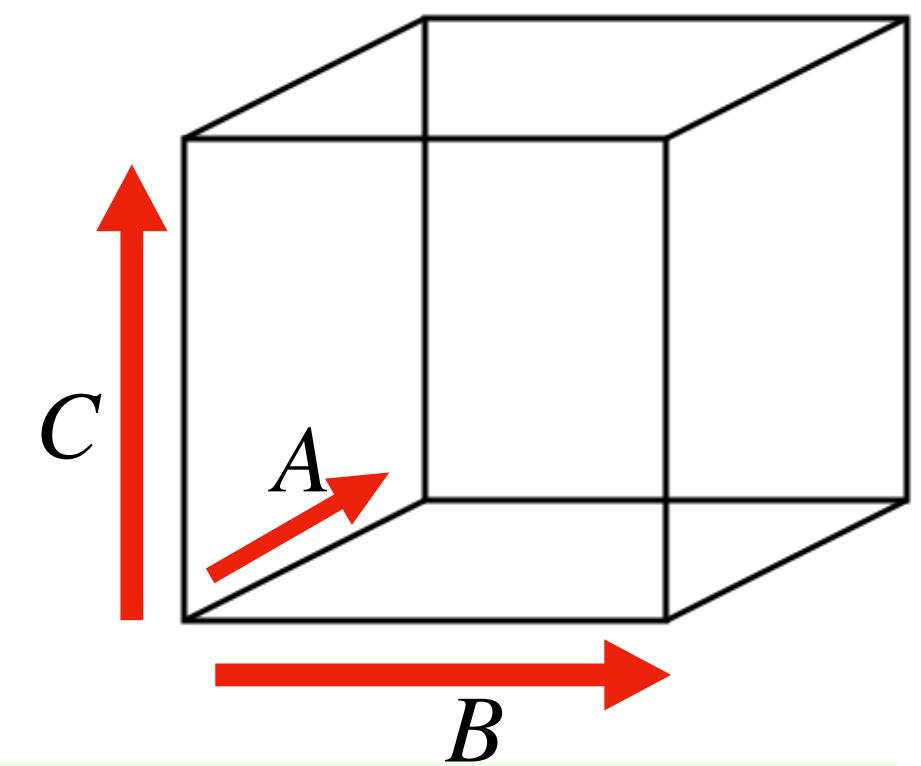
Then the following is a group action :

$$(A, B, C) \star \sum_{i,j,l=1}^{k,m,n} v(i,j,l) e_i \otimes e_j \otimes e_l = \sum_{i,j,l=1}^{k,m,n} v(i,j,l) Ae_i \otimes Be_j \otimes Ce_l$$

# Tensor group action

Let  $\mathbf{V}$  be the tensor space  $\sum_i u_i \otimes v_i \otimes w_i = \sum_{ijl} v(i,j,l) e_i \otimes e_j \otimes e_l$

Then the following is a group action :

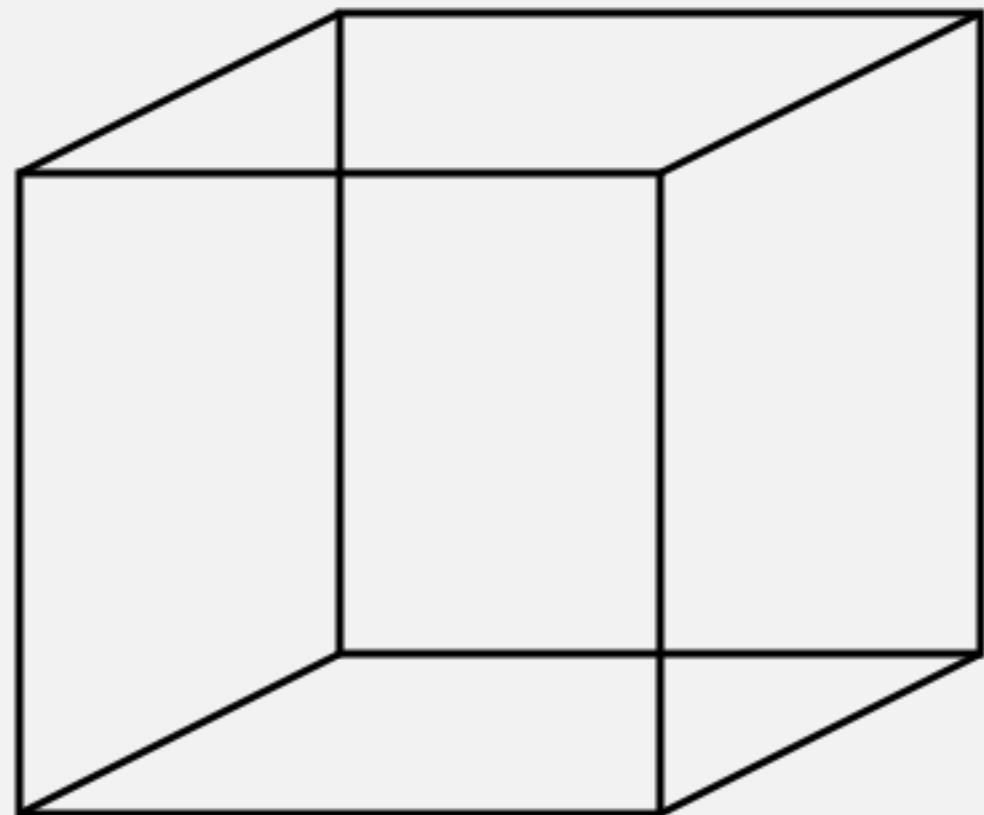


$$(A, B, C) \star \sum_{i,j,l=1}^{k,m,n} v(i,j,l) e_i \otimes e_j \otimes e_l = \sum_{i,j,l=1}^{k,m,n} v(i,j,l) Ae_i \otimes Be_j \otimes Ce_l$$

# Tensor group action

An example over  $\mathbb{F}_7$  :

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

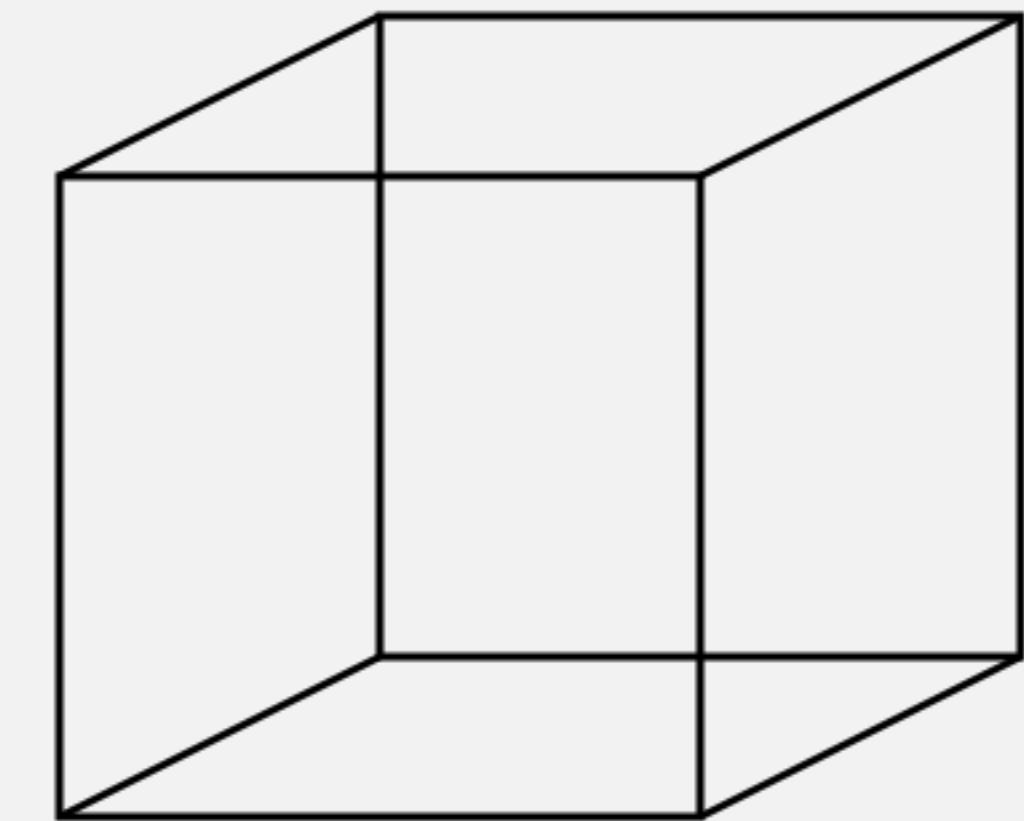


# Tensor group action

An example over  $\mathbb{F}_7$  :

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix},$$



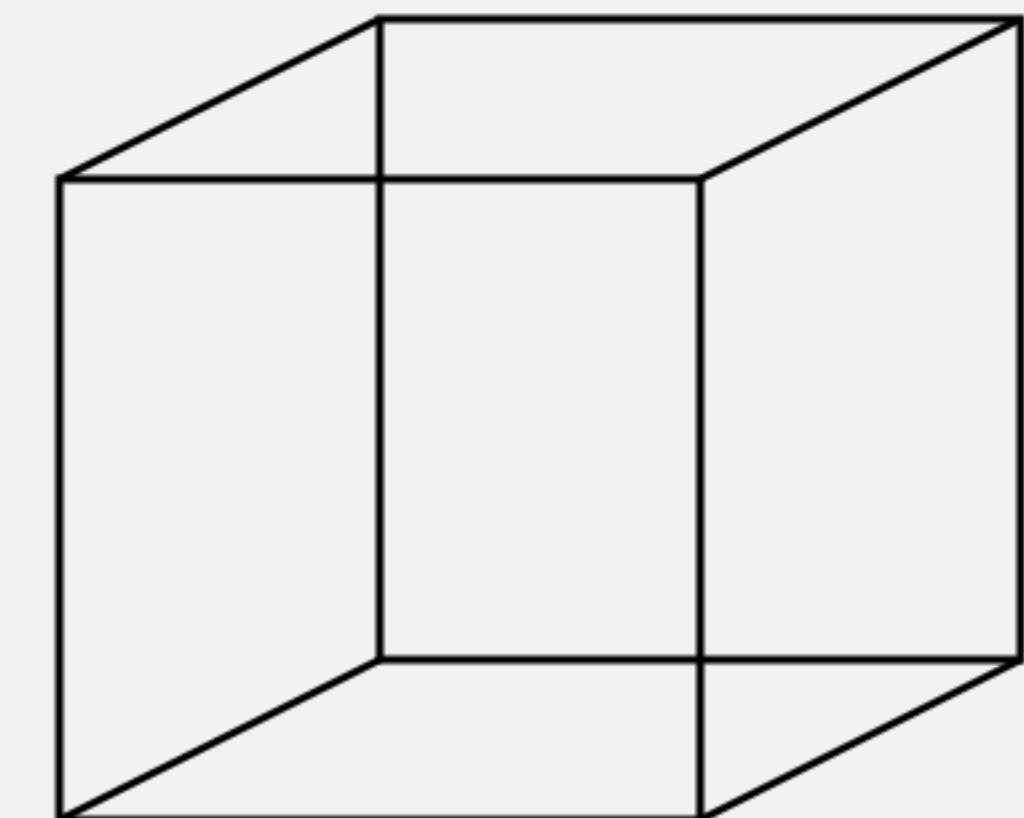
# Tensor group action

An example over  $\mathbb{F}_7$  :

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix},$$

$$(I, A, I) \star t := \sum_{i,j,k} e_i \otimes Ae_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix},$$



# Tensor group action

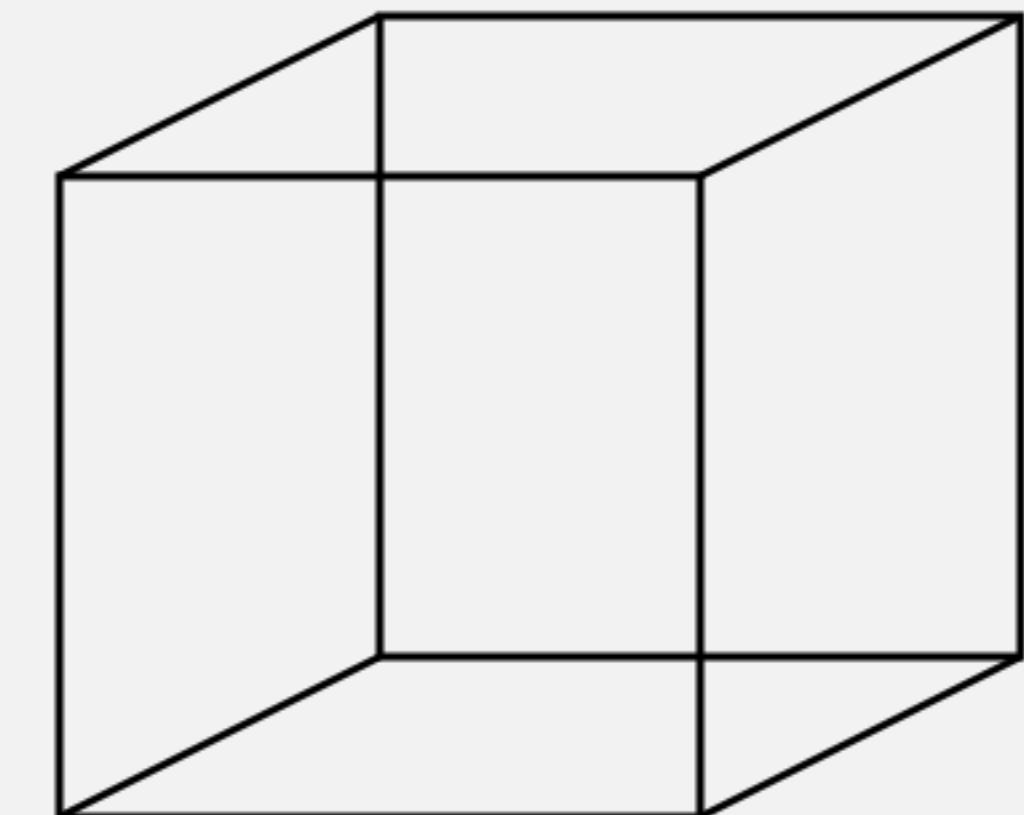
An example over  $\mathbb{F}_7$  :

$$t = \sum_{i,j,k} e_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 6 & 3 & 6 \\ 5 & 6 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

$$(A, I, I) \star t := \sum_{i,j,k} Ae_i \otimes e_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix},$$

$$(I, A, I) \star t := \sum_{i,j,k} e_i \otimes Ae_j \otimes e_k = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 5 & 5 & 5 \end{bmatrix},$$

$$(I, I, A) \star t := \sum_{i,j,k} e_i \otimes e_j \otimes Ae_k = \begin{bmatrix} 1 & 0 & 5 \\ 1 & 0 & 5 \\ 1 & 0 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 5 \\ 1 & 0 & 5 \\ 1 & 0 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 5 \\ 1 & 0 & 5 \\ 1 & 0 & 5 \end{bmatrix},$$



# Tensor group action (contribution 3)

[DFG23] Non-Interactive Commitment from Non-Transitive Group Actions. D'Alconzo, Flamini, Gangemi. Asiacrypt 2023.

**Problem :**

$$\text{Let } t_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Given  $(A, B, C) \star t_b$  where  $b \in \{0,1\}$ ,

recover  $b$ .

# Tensor group action (contribution 3)

[DFG23] Non-Interactive Commitment from Non-Transitive Group Actions. D'Alconzo, Flamini, Gangemi. Asiacrypt 2023.

**Problem :**

$$\text{Let } t_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Given  $(A, B, C) \star t_b$  where  $b \in \{0,1\}$ ,

recover  $b$ .

$t_0$

$t_1$

# Tensor group action (contribution 3)

[DFG23] Non-Interactive Commitment from Non-Transitive Group Actions. D'Alconzo, Flamini, Gangemi. Asiacrypt 2023.

**Problem :**

Let  $t_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Given  $(A, B, C) \star t_b$  where  $b \in \{0,1\}$ ,

recover  $b$ .

$t_0$

$T$

$t_1$

# Tensor group action (contribution 3)

[DFG23] Non-Interactive Commitment from Non-Transitive Group Actions. D'Alconzo, Flamini, Gangemi. Asiacrypt 2023.

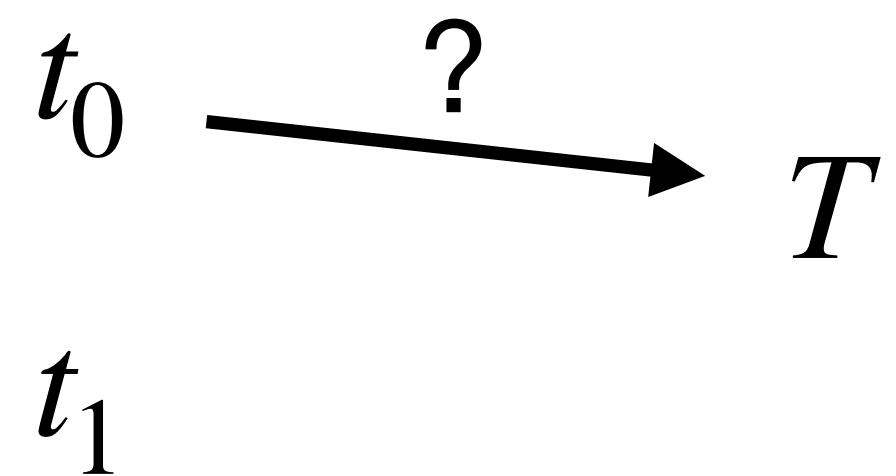
**Problem :**

Let  $t_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Given  $(A, B, C) \star t_b$  where  $b \in \{0,1\}$ ,

recover  $b$ .



# Tensor group action (contribution 3)

[DFG23] Non-Interactive Commitment from Non-Transitive Group Actions. D'Alconzo, Flamini, Gangemi. Asiacrypt 2023.

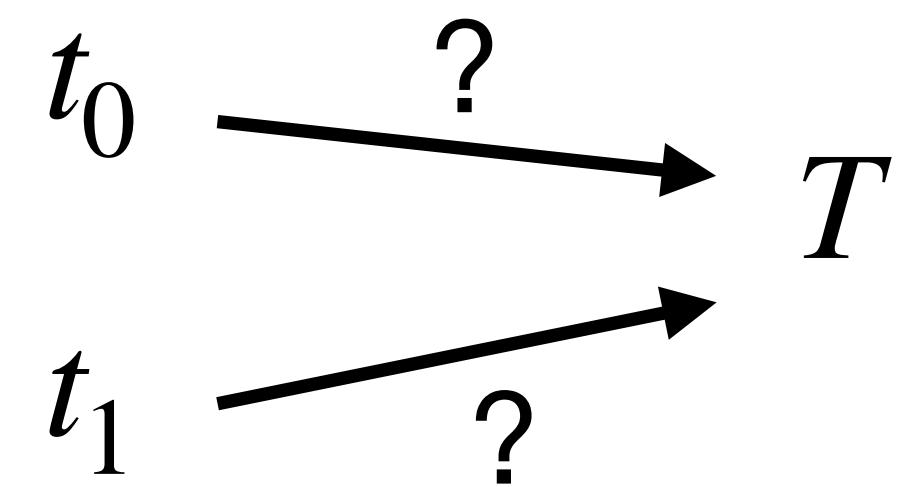
**Problem :**

Let  $t_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Given  $(A, B, C) \star t_b$  where  $b \in \{0,1\}$ ,

recover  $b$ .



# Tensor group action (contribution 3)

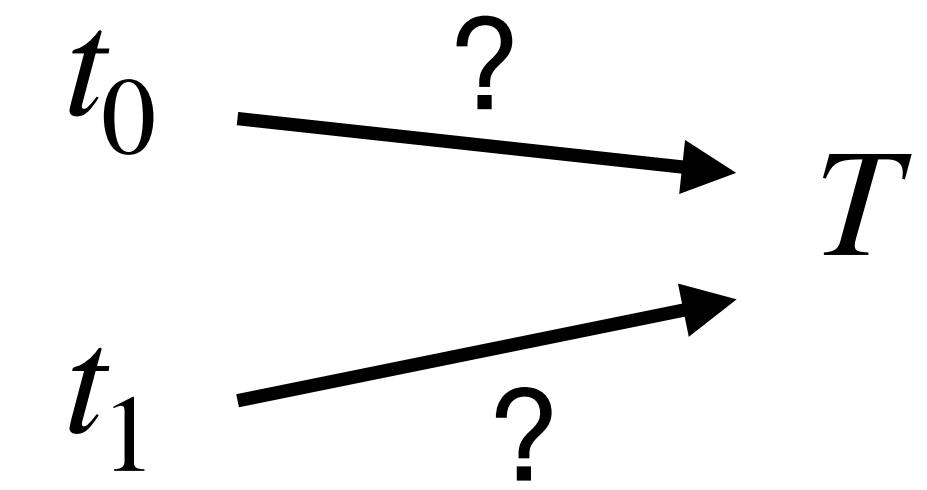
[DFG23] Non-Interactive Commitment from Non-Transitive Group Actions. D'Alconzo, Flamini, Gangemi. Asiacrypt 2023.

**Problem :**

Let  $t_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

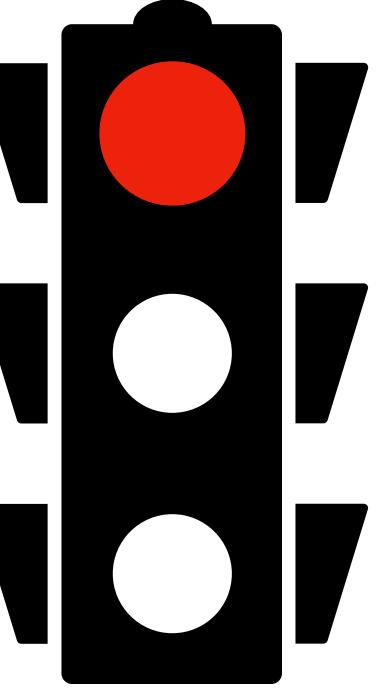
$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Given  $(A, B, C) \star t_b$  where  $b \in \{0,1\}$ ,  
recover  $b$ .



- we gave a polynomial time attack
- that recovers  $b$
- we showed how to recover
- $A, B, C$
- we offered a repair to the scheme
- and described a compatible ZKP

# Tensor group action (contribution 3)



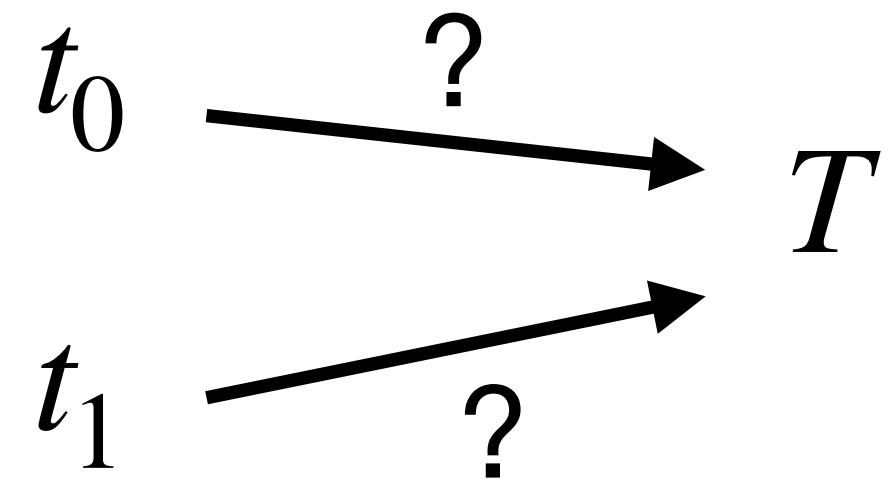
[DFG23] Non-Interactive Commitment from Non-Transitive Group Actions. D'Alconzo, Flamini, Gangemi. Asiacrypt 2023.

**Problem :**

Let  $t_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Given  $(A, B, C) \star t_b$  where  $b \in \{0,1\}$ ,  
recover  $b$ .



- we gave a polynomial time attack
- that recovers  $b$
- we showed how to recover
- $A, B, C$
- we offered a repair to the scheme
- and described a compatible ZKP

# Tensor group action (contribution 3)

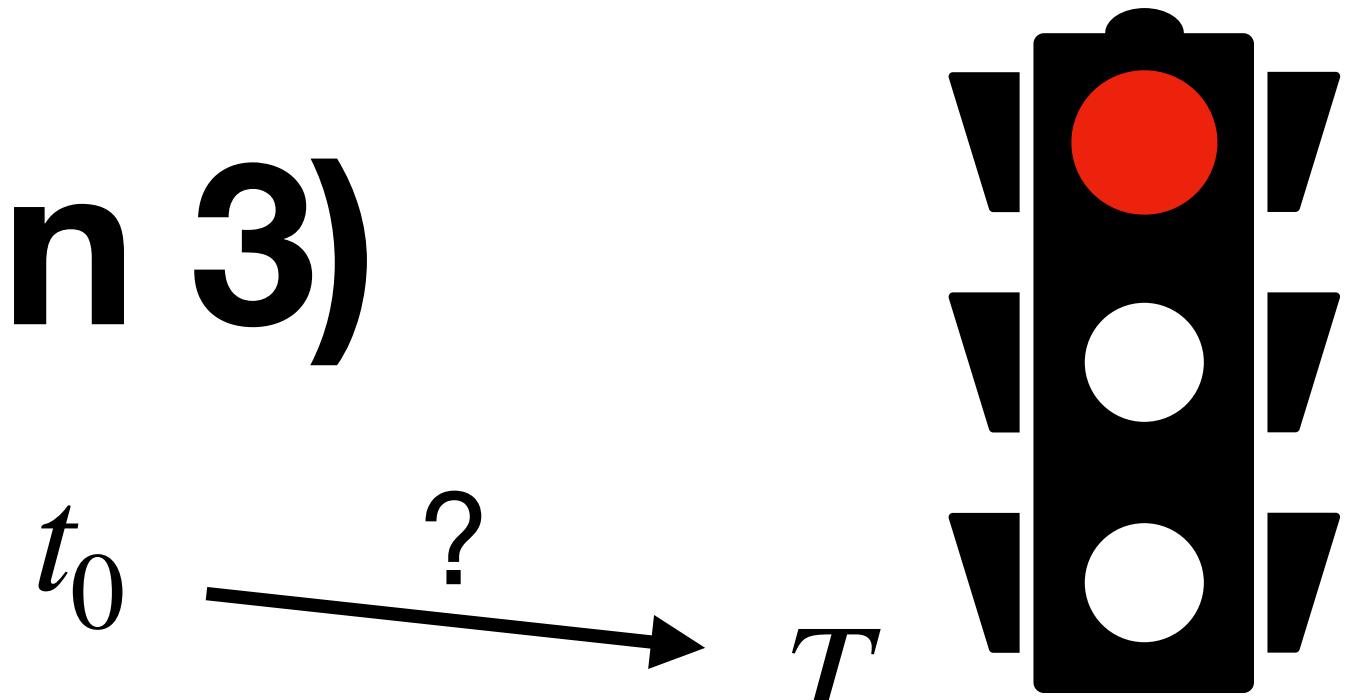
[DFG23] Non-Interactive Commitment from Non-Transitive Group Actions. D'Alconzo, Flamini, Gangemi. Asiacrypt 2023.

**Problem :**

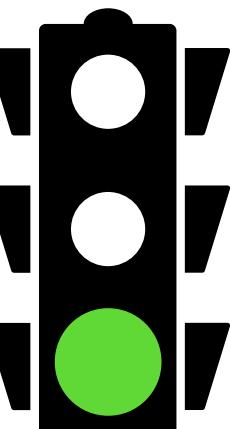
Let  $t_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$t_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Given  $(A, B, C) \star t_b$  where  $b \in \{0,1\}$ ,  
recover  $b$ .

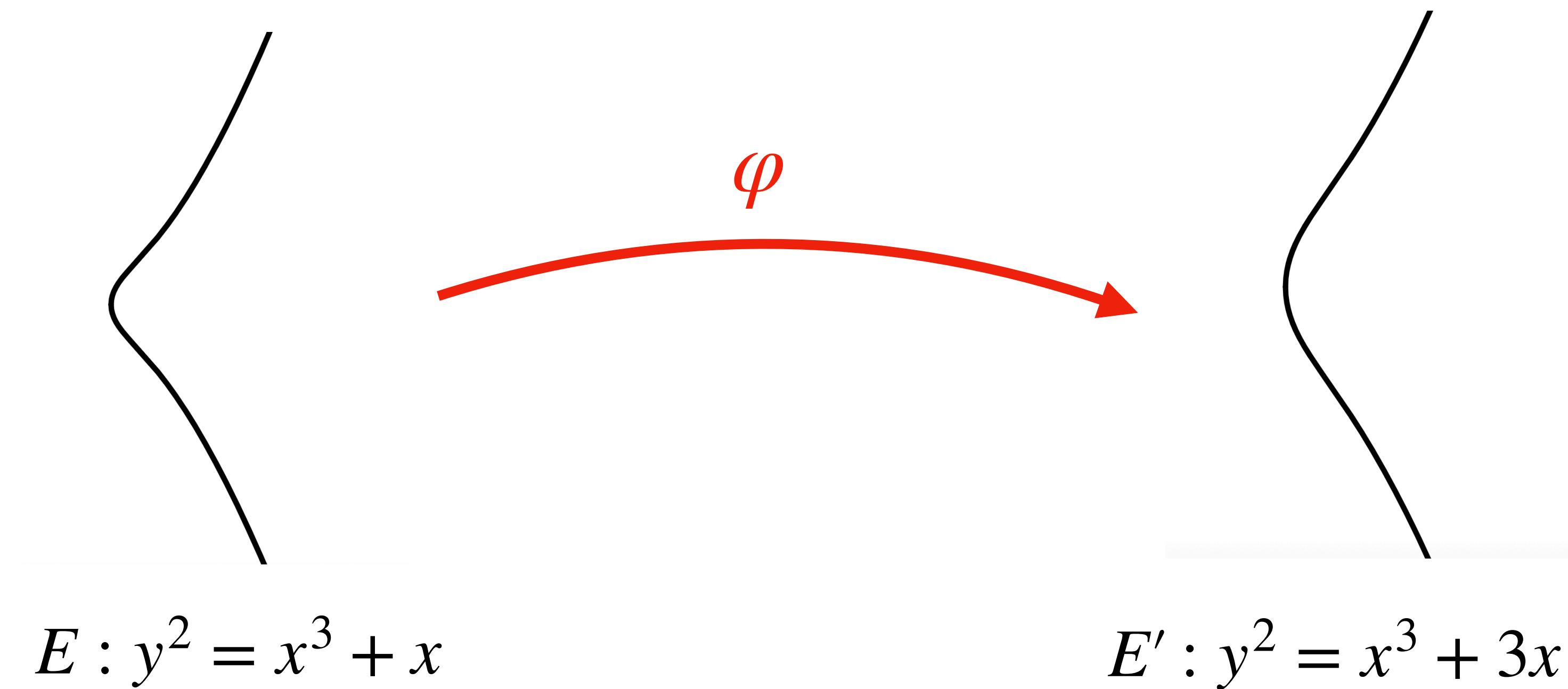


- we gave a polynomial time attack
- that recovers  $b$
- we showed how to recover
- $A, B, C$
- we offered a repair to the scheme
- and described a compatible ZKP



# Isogeny group action

An isogeny  $\varphi : E \rightarrow E'$  is a non-constant rational map between two elliptic curves

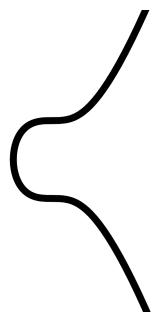


# Isogeny group action

An “ $\ell$ -isogeny” = an isogeny of degree  $\ell$

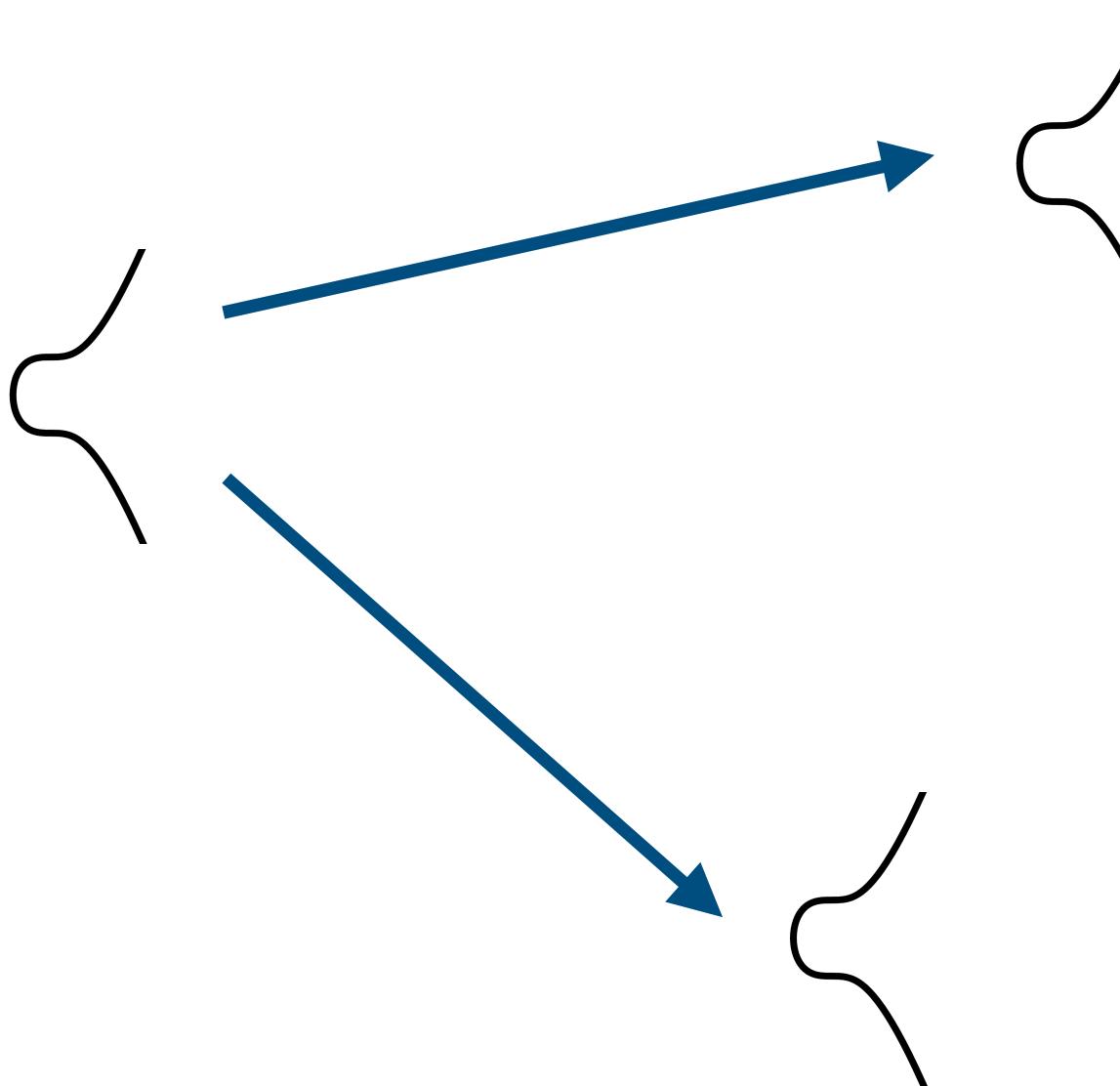
# Isogeny group action

An “ $\ell$ -isogeny” = an isogeny of degree  $\ell$



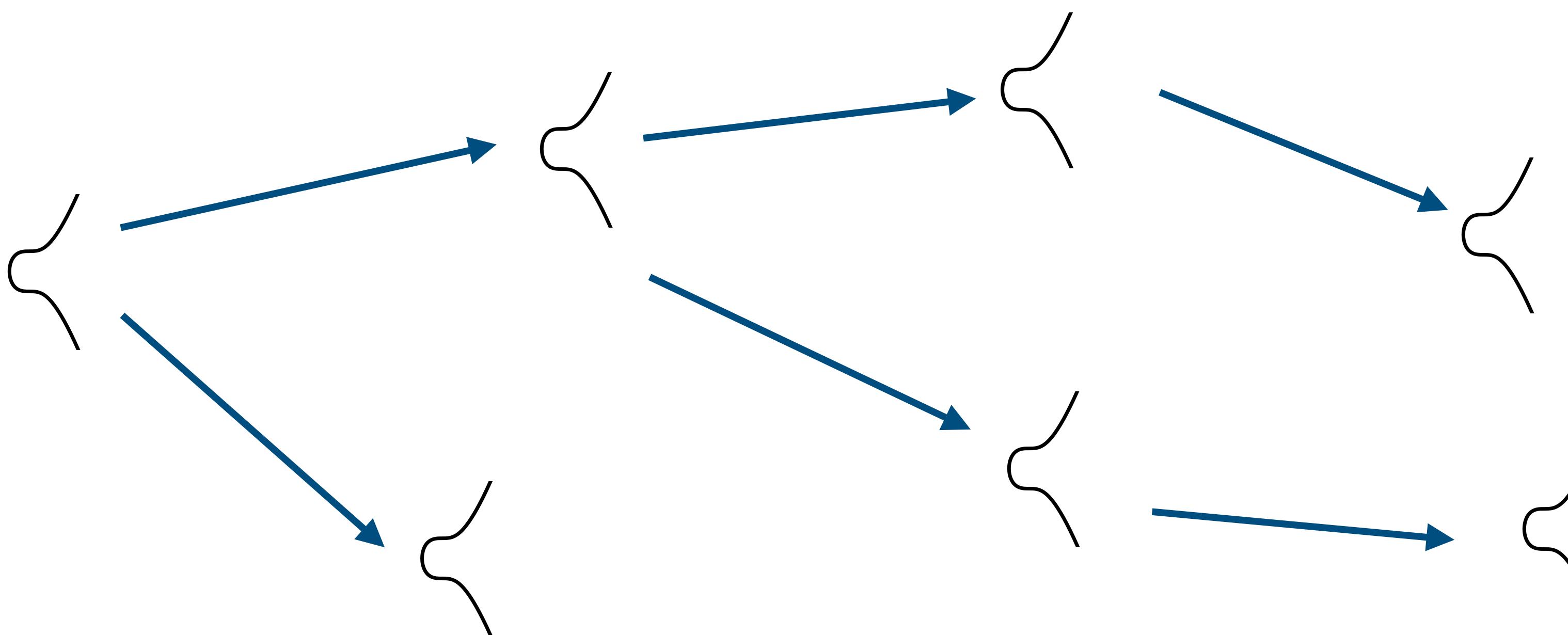
# Isogeny group action

An “ $\ell$ -isogeny” = an isogeny of degree  $\ell$



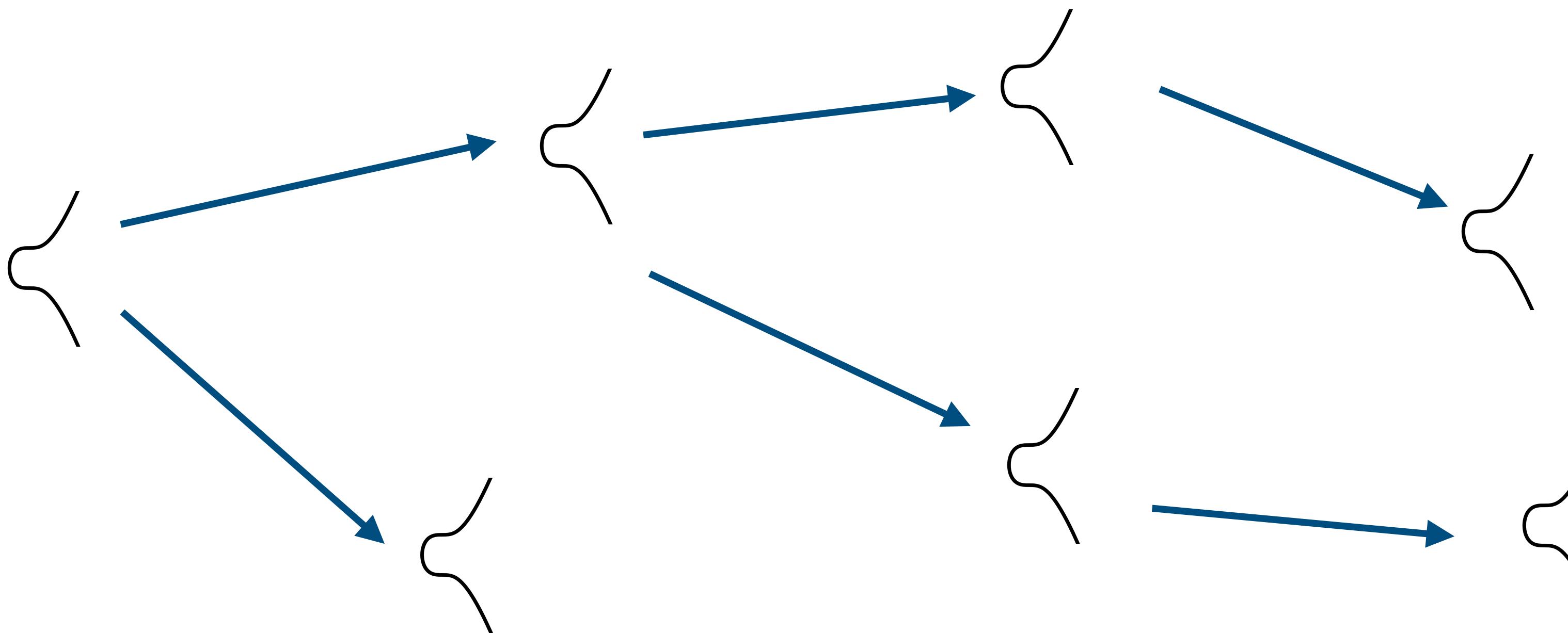
# Isogeny group action

An “ $\ell$ -isogeny” = an isogeny of degree  $\ell$



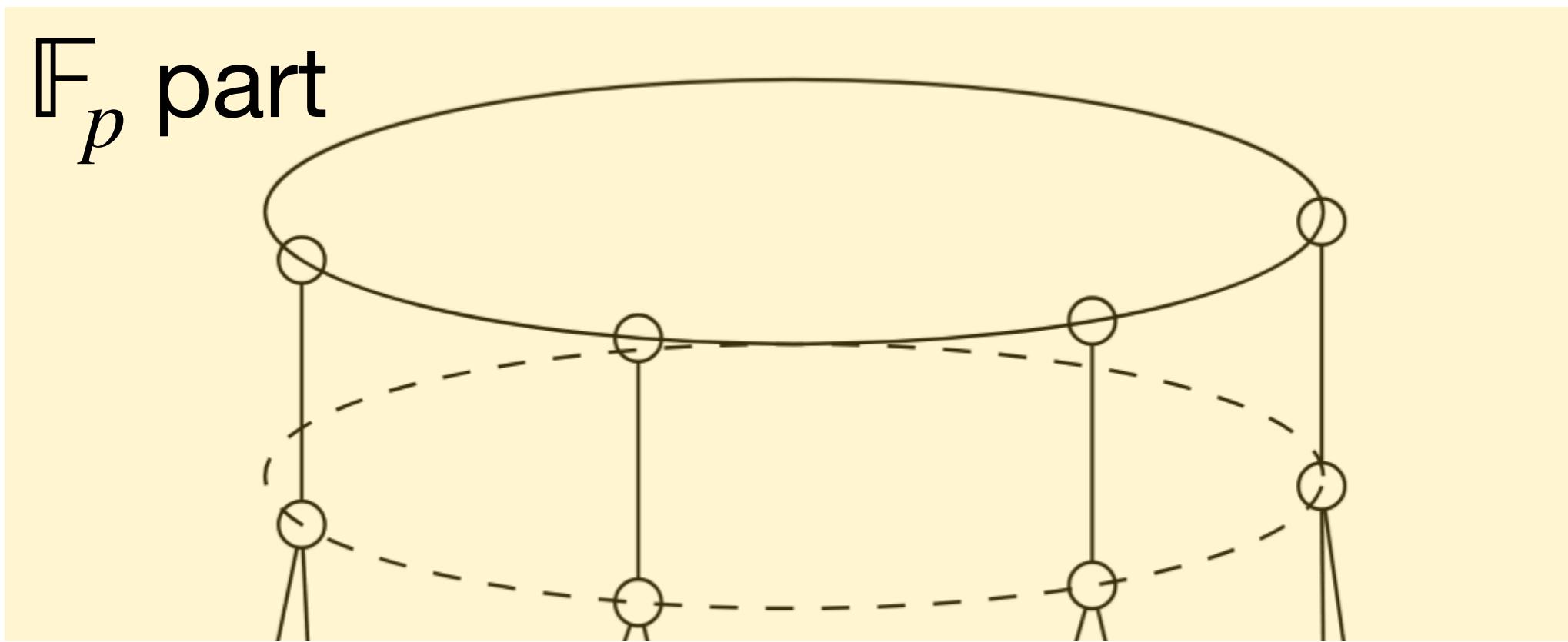
# Isogeny group action

An “ $\ell$ -isogeny” = an isogeny of degree  $\ell$



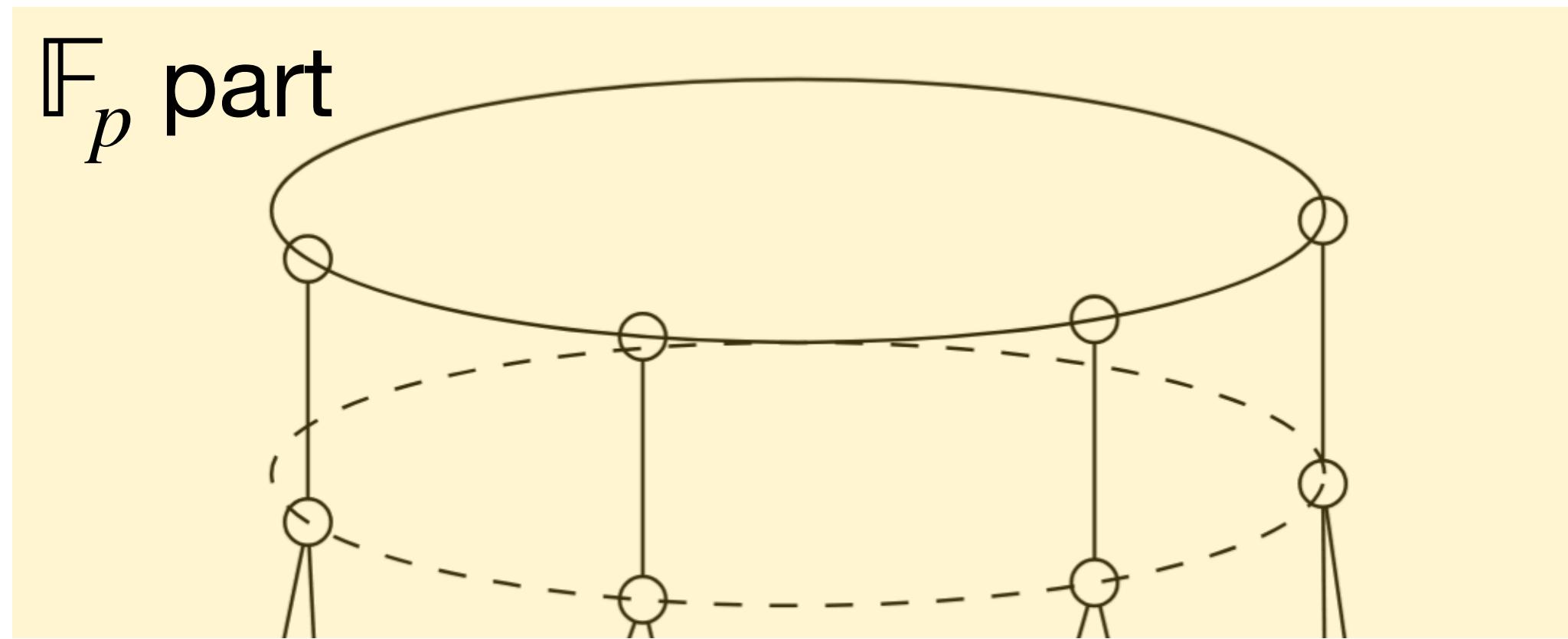
Vertices = elliptic curves, edges =  $\ell$ -isogenies

# Isogeny group action



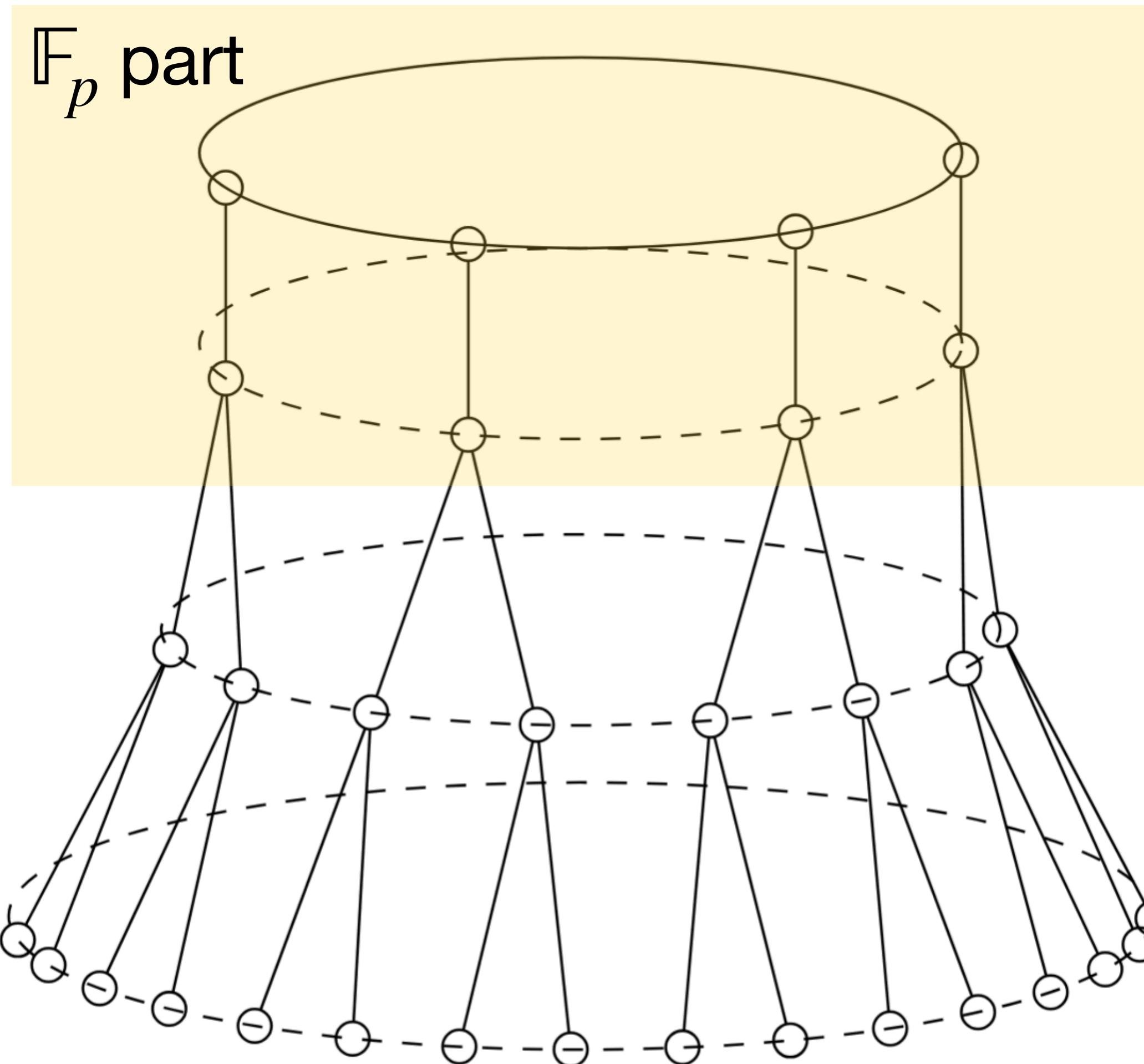
- restrict to isogenies of degree  $\ell$
- over  $\mathbb{F}_p$  we get an isogeny **volcano**

# Isogeny group action



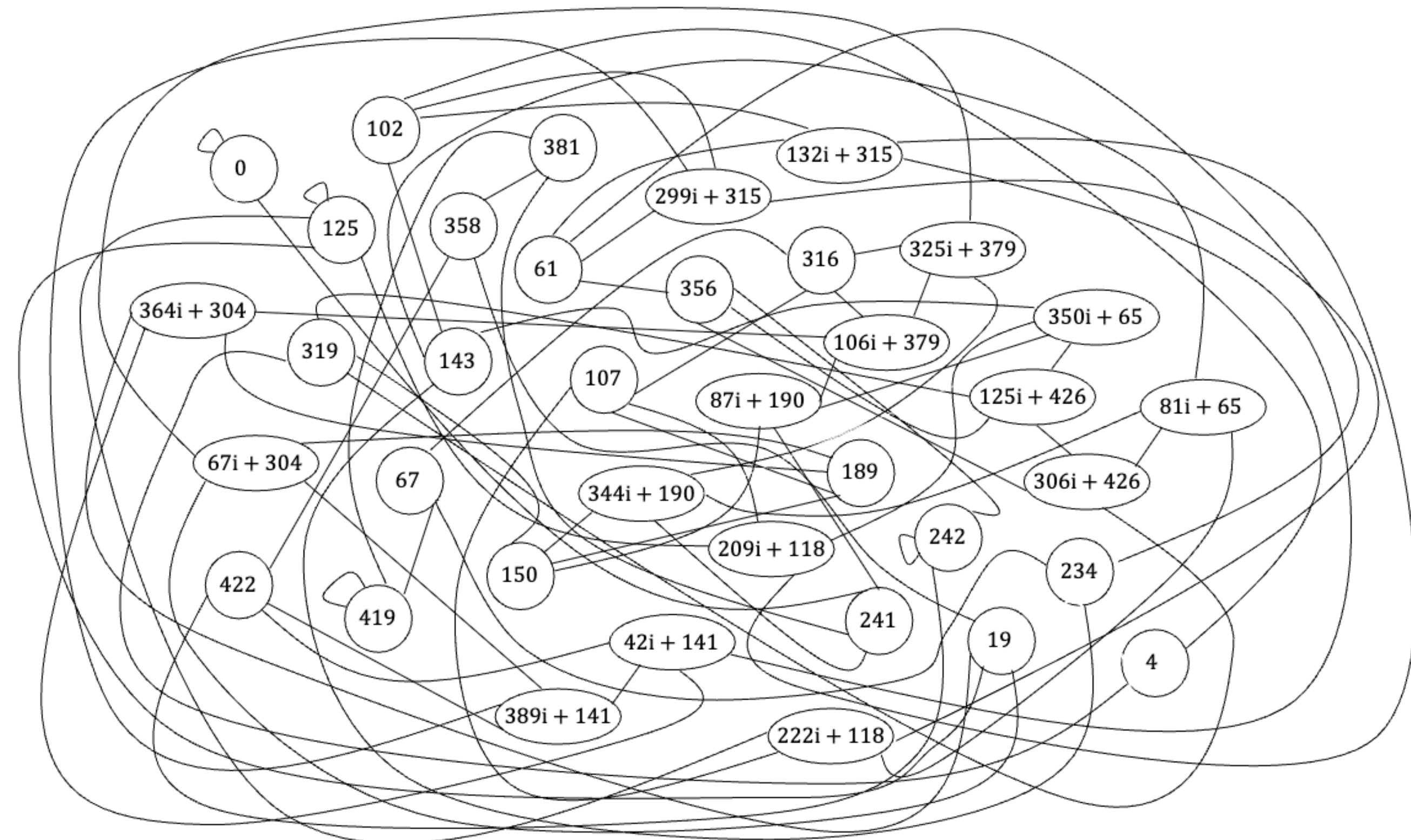
- restrict to isogenies of degree  $\ell$
- over  $\mathbb{F}_p$  we get an isogeny **volcano**
- over  $\mathbb{F}_{p^k}, k > 1$  only some graphs will be volcanoes (ordinary curves)

# Isogeny group action



- restrict to isogenies of degree  $\ell$
- over  $\mathbb{F}_p$  we get an isogeny **volcano**
- over  $\mathbb{F}_{p^k}, k > 1$  only *some* graphs will be volcanoes (ordinary curves)

# Isogeny group action



Supersingular graph  
[image by Craig Costello]

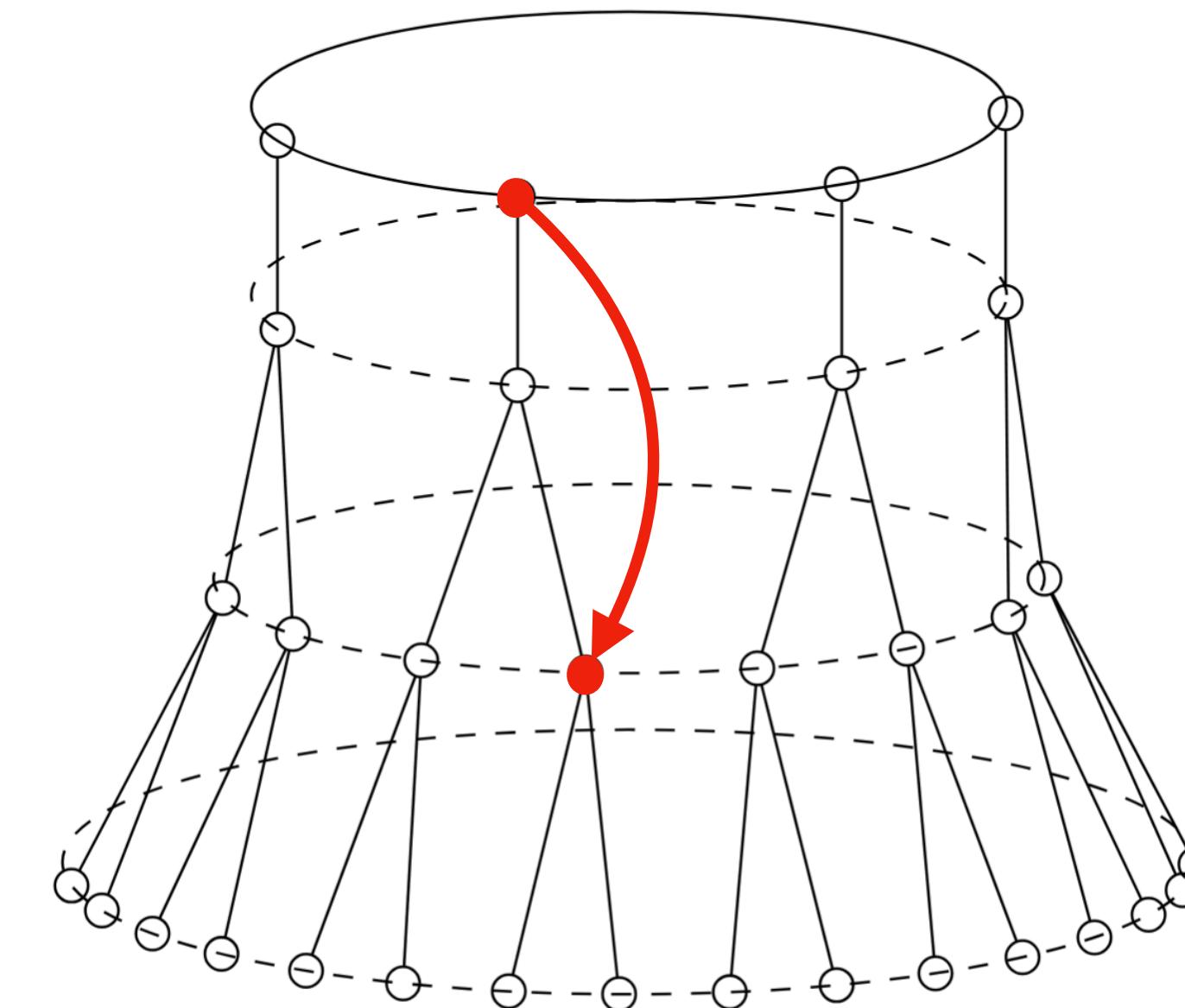
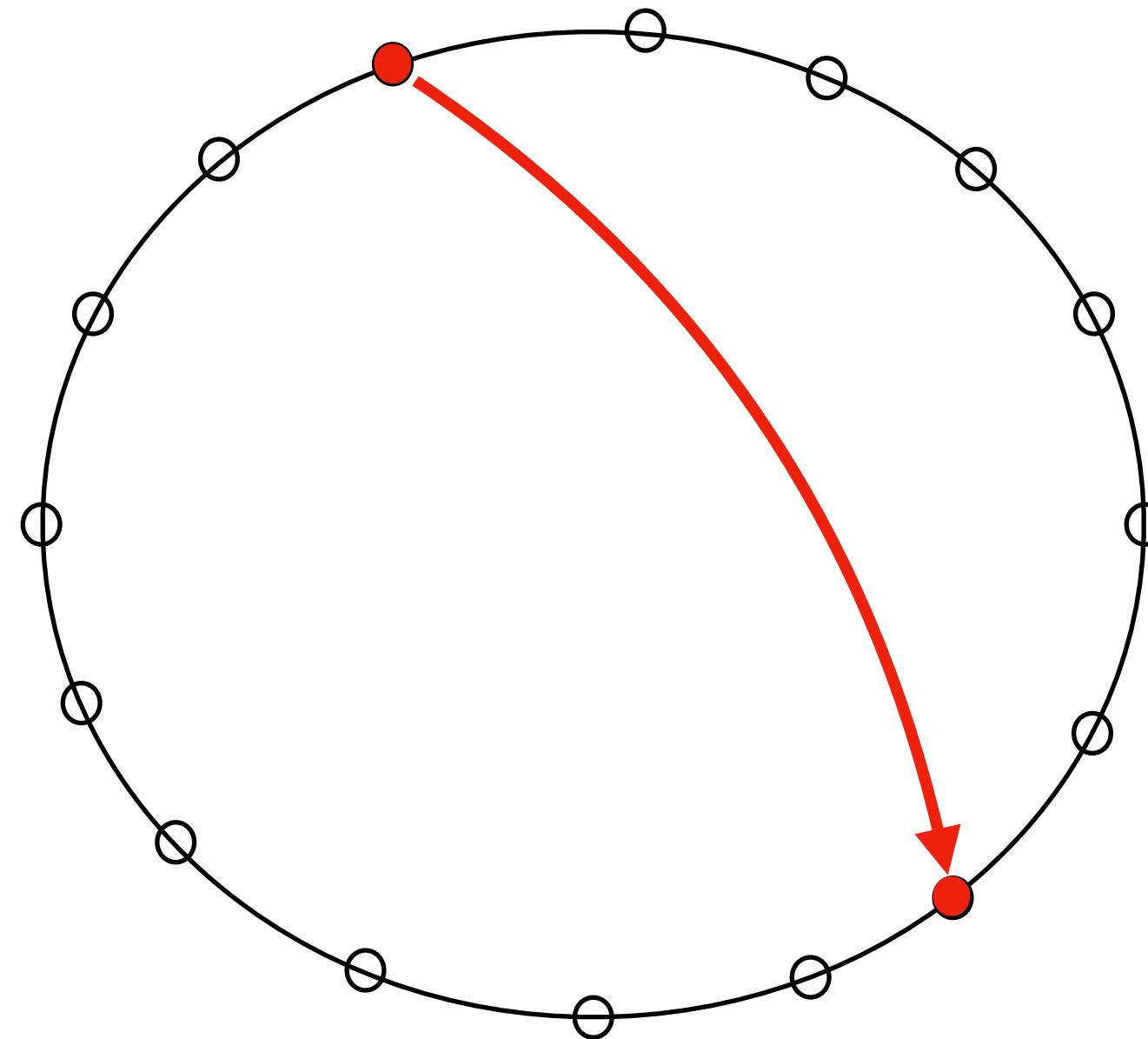
# Isogeny group action



Supersingular graph  
[image by Craig Costello]

# Isogeny group action

★: “walks” in the graph    × (some particular) elliptic curves    → (some particular) elliptic curves



# Contribution 4

[In submission]

## Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs

Paul Frixons<sup>1</sup>, Valerie Gilchrist<sup>1</sup>, Péter Kutas<sup>2,3</sup>, Simon-Philipp Merz<sup>4</sup>,  
Christophe Petit<sup>1,3</sup>, Lam L. Pham<sup>5</sup>

<sup>1</sup> Université Libre de Bruxelles, Belgium

<sup>2</sup> Eötvös Loránd University, Hungary

<sup>3</sup> University of Birmingham, United Kingdom

<sup>4</sup> ETH Zürich, Switzerland

<sup>5</sup> Ghent University, Belgium

**Abstract.** Cryptographic group actions provide a basis for simple post-quantum generalizations of many cryptographic protocols based on the discrete logarithm problem (DLP). However, many advanced group action-based protocols do not solely rely on the core group action problem (the so-called vectorization problem), but also on *variants* of this problem, to either improve efficiency or enable new functionalities. For example, the security of the CSI-SharK threshold signature protocol relies on the hardness of the *Vectorization Problem with Shifted Inputs* where (in DLP

# Isogeny group action (contribution 4)

[ABCP23] CSI-SharK: CSI-FiSh with Sharing-friendly Keys. Atapoor, Baghery, Cozzo, Pedersen. ACISP 2023.

**Problem :**

compute  $z$ .

# Isogeny group action (contribution 4)

[ABCP23] CSI-SharK: CSI-FiSh with Sharing-friendly Keys. Atapoor, Baghery, Cozzo, Pedersen. ACISP 2023.

**Problem :**

$$E_0$$

compute  $z$ .

# Isogeny group action (contribution 4)

[ABCP23] CSI-SharK: CSI-FiSh with Sharing-friendly Keys. Atapoor, Baghery, Cozzo, Pedersen. ACISP 2023.

**Problem :**

$$E_0 \xrightarrow{\star g^z} E_1$$

compute  $z$ .

# Isogeny group action (contribution 4)

[ABCP23] CSI-SharK: CSI-FiSh with Sharing-friendly Keys. Atapoor, Baghery, Cozzo, Pedersen. ACISP 2023.

**Problem :**

$$E_0 \xrightarrow{\star g^z} E_1 \xrightarrow{\star g^z} E_2$$

compute  $z$ .

# Isogeny group action (contribution 4)

[ABCP23] CSI-SharK: CSI-FiSh with Sharing-friendly Keys. Atapoor, Baghery, Cozzo, Pedersen. ACISP 2023.

**Problem :**

$$E_0 \xrightarrow{\star g^z} E_1 \xrightarrow{\star g^z} E_2 \xrightarrow{\star g^z} E_3$$

compute  $z$ .

# Isogeny group action (contribution 4)

[ABCP23] CSI-SharK: CSI-FiSh with Sharing-friendly Keys. Atapoor, Baghery, Cozzo, Pedersen. ACISP 2023.

**Problem :**

$$E_0 \xrightarrow{\star g^z} E_1 \xrightarrow{\star g^z} E_2 \xrightarrow{\star g^z} E_3 \dots \xrightarrow{\star g^z} E_M$$

compute  $z$ .

# Isogeny group action (contribution 4)

[ABCP23] CSI-SharK: CSI-FiSh with Sharing-friendly Keys. Atapoor, Baghery, Cozzo, Pedersen. ACISP 2023.

**Problem :**

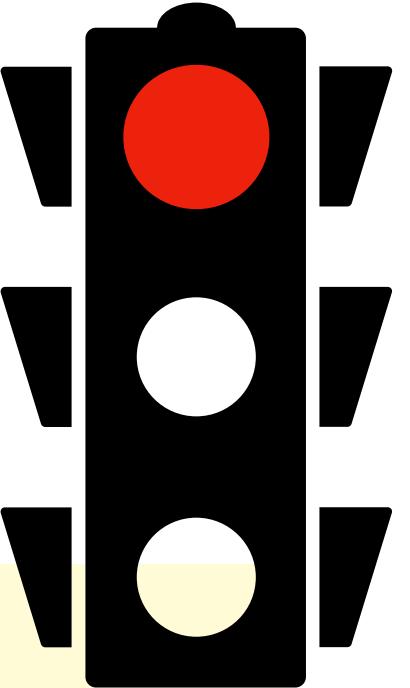
$$E_0 \xrightarrow{\star g^z} E_1 \xrightarrow{\star g^z} E_2 \xrightarrow{\star g^z} E_3 \dots \xrightarrow{\star g^z} E_M$$

compute  $z$ .

- we presented the quantum algorithm from Childs and van Dam for use in cryptography and used it to analyze the security of this problem
- we specialized relevant subroutines in order to obtain a concrete security estimate
- this approach improved upon the state-of-the-art (Kuperberg on one instance)

# Isogeny group action (contribution 4)

[ABCP23] CSI-SharK: CSI-FiSh with Sharing-friendly Keys. Atapoor, Baghery, Cozzo, Pedersen. ACISP 2023.



**Problem :**

$$E_0 \xrightarrow{\star g^z} E_1 \xrightarrow{\star g^z} E_2 \xrightarrow{\star g^z} E_3 \dots \xrightarrow{\star g^z} E_M$$

compute  $z$ .

- we presented the quantum algorithm from Childs and van Dam for use in cryptography and used it to analyze the security of this problem
- we specialized relevant subroutines in order to obtain a concrete security estimate
- this approach improved upon the state-of-the-art (Kuperberg on one instance)

# Contribution 5

[Latincrypt 2025]

## Improved algorithms for ascending isogeny volcanoes, and applications

Steven D. Galbraith<sup>1</sup>, Valerie Gilchrist<sup>2</sup>, Damien Robert<sup>3</sup>

<sup>1</sup> University of Auckland, Auckland, New Zealand

<sup>2</sup> Université Libre de Bruxelles, Brussels, Belgium

<sup>3</sup> Inria Bordeaux, Institut de Mathématiques de Bordeaux, France

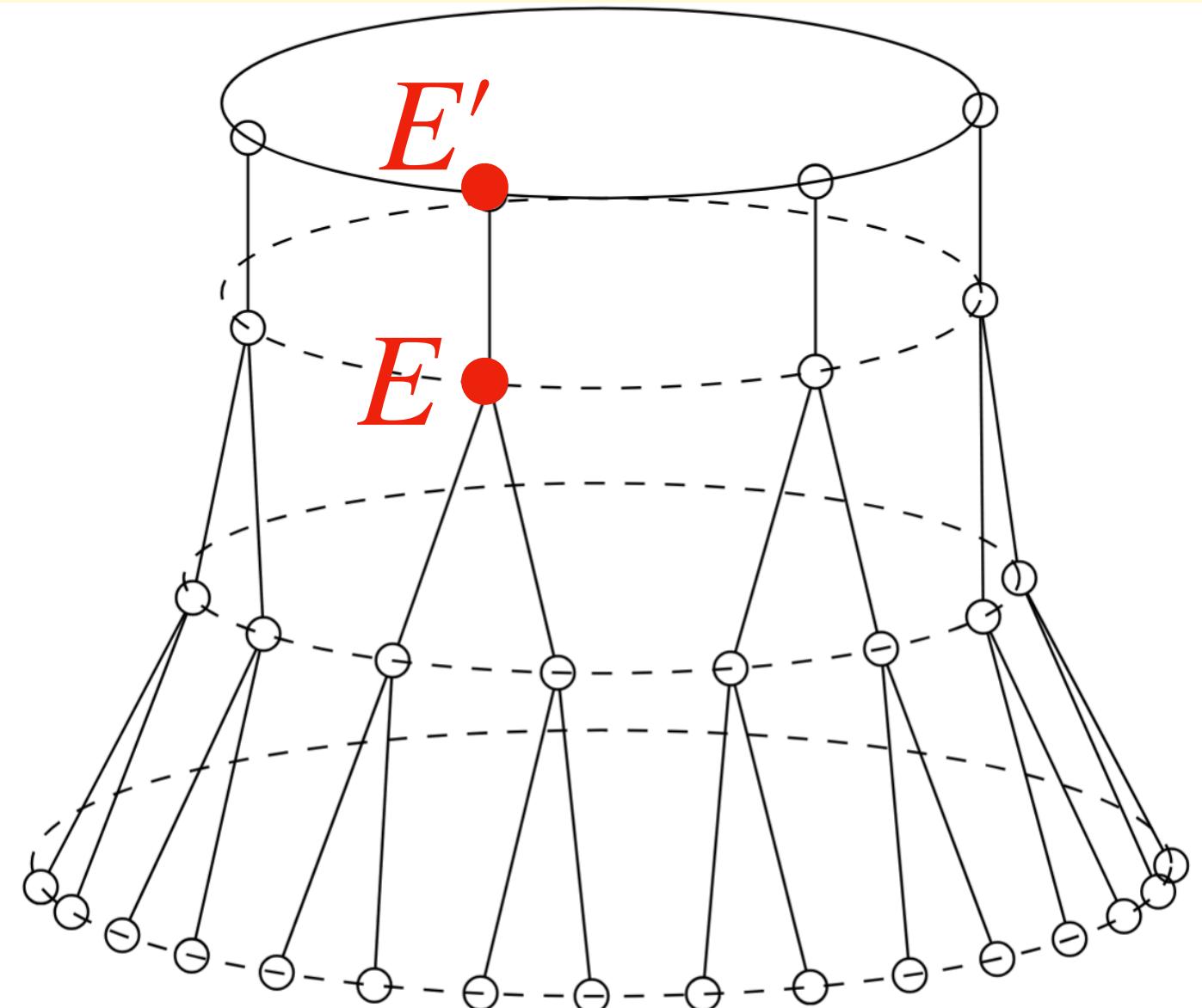
**Abstract.** Given two elliptic curves over  $\mathbb{F}_q$ , computing an isogeny mapping one to the other is conjectured to be classically and quantumly hard. This problem plays an important role in the security of elliptic curve cryptography. In 2024, Galbraith applied recently developed techniques for isogenies to improve the state-of-the-art for this problem.

In this work, we focus on computing ascending isogenies with respect to an orientation. Our results apply to both ordinary and supersingular curves. We give a simplified framework for computing self-pairings, and show how they can be used to improve upon the approach from Galbraith to recover these ascending isogenies and eliminate a heuristic

# Isogeny group action (contribution 5)

**Problem :**

Given  $E, E' = \alpha \star E$ , where  $\alpha$  corresponds to a vertical isogeny of known degree,  
compute  $\alpha$ .

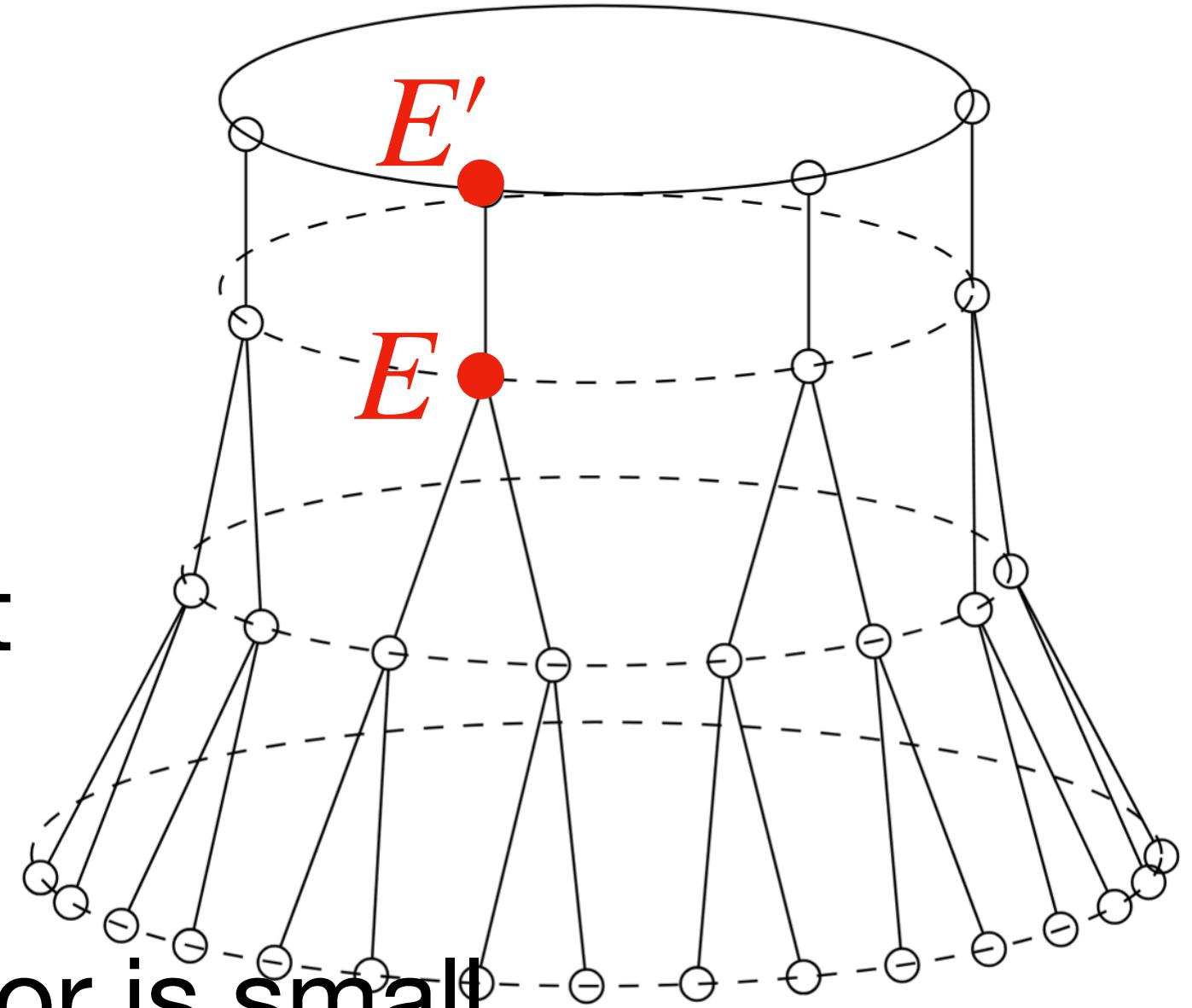


# Isogeny group action (contribution 5)

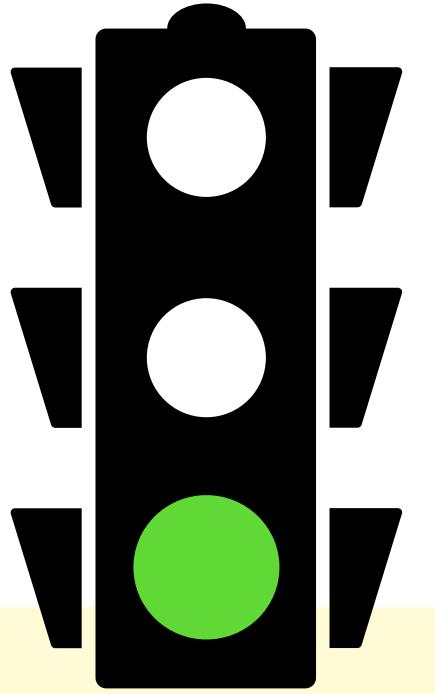
**Problem :**

Given  $E, E' = \alpha \star E$ , where  $\alpha$  corresponds to a vertical isogeny of known degree,  
compute  $\alpha$ .

- we considered special cases of the worst case of this problem
- we removed a heuristic assumption from the state-of-the-art when the conductor is large
- we gave an improved complexity analysis when the conductor is small



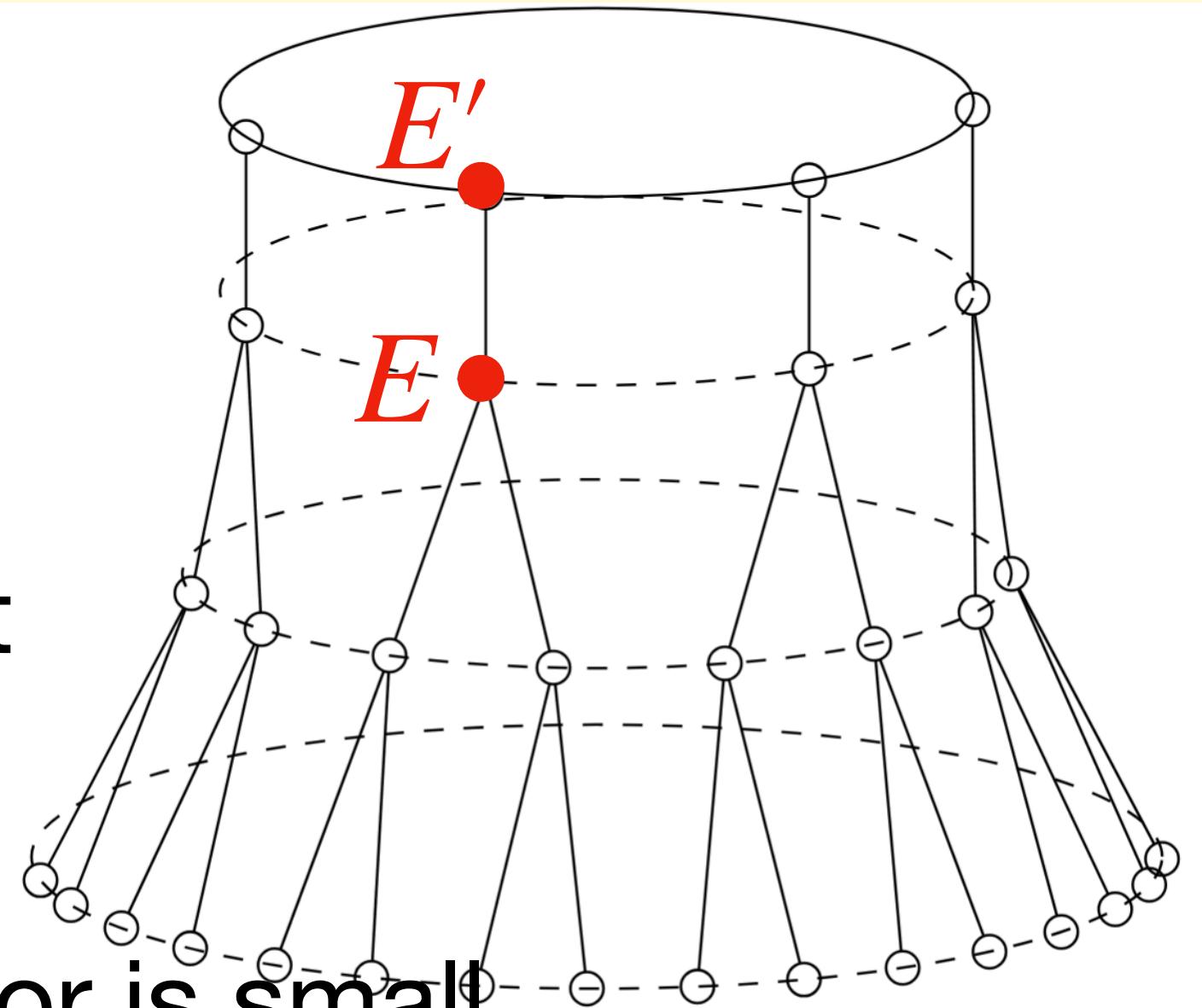
# Isogeny group action (contribution 5)



**Problem :**

Given  $E, E' = \alpha \star E$ , where  $\alpha$  corresponds to a vertical isogeny of known degree,  
compute  $\alpha$ .

- we considered special cases of the worst case of this problem
- we removed a heuristic assumption from the state-of-the-art when the conductor is large
- we gave an improved complexity analysis when the conductor is small



# Contribution 6

[Latincrypt 2023]

## Fast and Frobenius: Rational Isogeny Evaluation over Finite Fields

Gustavo Banegas<sup>1</sup>, Valerie Gilchrist<sup>2</sup>, Anaëlle Le Dévéhat<sup>3</sup>, Benjamin Smith<sup>3</sup>

<sup>1</sup> Qualcomm France SARL, Valbonne, France

<sup>2</sup> Université Libre de Bruxelles and FRIA, Brussels, Belgium

<sup>3</sup> Inria and Laboratoire d’Informatique de l’École polytechnique, Institut Polytechnique de Paris, Palaiseau, France

**Abstract.** Consider the problem of efficiently evaluating isogenies  $\phi : \mathcal{E} \rightarrow \mathcal{E}/H$  of elliptic curves over a finite field  $\mathbb{F}_q$ , where the kernel  $H = \langle G \rangle$  is a cyclic group of odd (prime) order: given  $\mathcal{E}$ ,  $G$ , and a point (or several points)  $P$  on  $\mathcal{E}$ , we want to compute  $\phi(P)$ . This problem is at the heart of efficient implementations of group-action- and isogeny-based post-quantum cryptosystems such as CSIDH. Algorithms based on Vélu’s formulæ give an efficient solution when the kernel generator  $G$  is defined over  $\mathbb{F}_q$ , but for general isogenies  $G$  is only defined over some extension  $\mathbb{F}_{q^k}$ , even though  $\langle G \rangle$  as a whole (and thus  $\phi$ ) is defined over the base field  $\mathbb{F}_q$ ; and the performance of Vélu-style algorithms degrades rapidly as  $k$  grows. In this article we revisit isogeny evaluation with a special focus on the case when  $k=1/2$ . We propose a Miller-like algorithm

# Isogeny group action (contribution 6)

**Problem :**

Given  $E$  and a kernel generator  $G$  (possibly defined over an extension field),  
evaluate the isogeny  $\phi : E \rightarrow E'$  with kernel  $\langle G \rangle$  on some point  $P$ .

# Isogeny group action (contribution 6)

**Problem :**

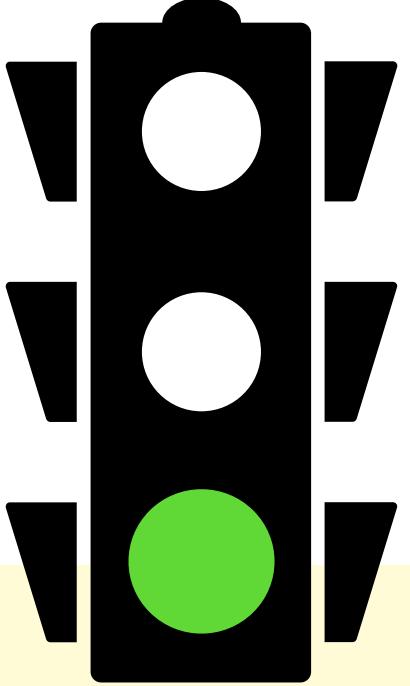
Given  $E$  and a kernel generator  $G$  (possibly defined over an extension field),  
evaluate the isogeny  $\phi : E \rightarrow E'$  with kernel  $\langle G \rangle$  on some point  $P$ .

-we describe an algorithm for computing the kernel polynomial of  $\phi$

that improves upon the number of arithmetic operations necessary

-we show how to achieve further savings when  $G$  is defined over some field  
extension up to degree 12 using the Frobenius map

# Isogeny group action (contribution 6)



**Problem :**

Given  $E$  and a kernel generator  $G$  (possibly defined over an extension field),  
evaluate the isogeny  $\phi : E \rightarrow E'$  with kernel  $\langle G \rangle$  on some point  $P$ .

-we describe an algorithm for computing the kernel polynomial of  $\phi$

that improves upon the number of arithmetic operations necessary

-we show how to achieve further savings when  $G$  is defined over some field extension up to degree 12 using the Frobenius map

# Contribution 7

[Mathcrypt 2022]

## Efficient supersingularity testing over $\mathbb{F}_p$ and CSIDH key validation

Gustavo Banegas<sup>1</sup>, Valerie Gilchrist<sup>2,1,\*</sup>, Benjamin Smith<sup>1</sup>

<sup>1</sup>Inria and Laboratoire d’Informatique de l’École polytechnique, Institut Polytechnique de Paris, Palaiseau, France

<sup>2</sup>University of Waterloo, Canada

**Abstract** Many public-key cryptographic protocols, notably non-interactive key exchange (NIKE), require incoming public keys to be validated to mitigate some adaptive attacks. In CSIDH, an isogeny-based post-quantum NIKE, a key is deemed legitimate if the given Montgomery coefficient specifies a supersingular elliptic curve over the prime field. In this work, we survey the current supersingularity tests used for CSIDH key validation, and implement and measure two new alternative algorithms. Our implementation shows that we can determine supersingularity substantially faster, and using less memory, than the state-of-the-art.

**Keywords:** Isogenies, Key validation, Supersingularity, Elliptic Curves

### 1 INTRODUCTION

The security of many public-key cryptosystems assumes that public keys are honestly generated: that is, that public keys have not been manipulated by adversaries. *Key validation* is the process of determining whether an incoming public key was plausibly constructed following the protocol.

The simplest example of this is in non-interactive key exchange (NIKE). Consider classic static Diffie–Hellman in a finite field: the system parameters fix a prime modulus  $p$  and a generator  $g$  in  $\mathbb{F}_p$  for a subgroup  $G = \langle g \rangle \subset \mathbb{F}_p^\times$  of prime order  $r$ . Alice samples a long-term secret integer  $a$ , and binds to the corresponding public key  $A = g^a$ . An honest Bob computes his keypair  $(B = g^b, b)$ , and the shared secret is  $S = A^b = B^a$ . However, if  $G$  is a proper subgroup of  $\mathbb{F}_p^\times$ , then a dishonest Bob can choose some  $h$  in  $\mathbb{F}_p^\times \setminus G$ , of order  $s \mid (p - 1)/r$ , and transmit the malformed public key  $B' = B \cdot h$ . The shared secret as computed by Alice is now  $S' = (B')^a = S \cdot h^a$ , while Bob derives the original  $S = A^b$ . The success or failure of subsequent encrypted communication tells Bob whether  $S = S'$ , and hence whether  $a$  is  $0 \pmod{s}$ .

To avoid leaking information on her long-term private key to adaptive adversaries, then, Alice must validate incoming public keys as being honestly generated. In the example above, this amounts to checking that Bob’s

# Isogeny group action (contribution 7)

**Problem :**

Given an elliptic curve  $E$  decide whether  $E$  is supersingular or ordinary

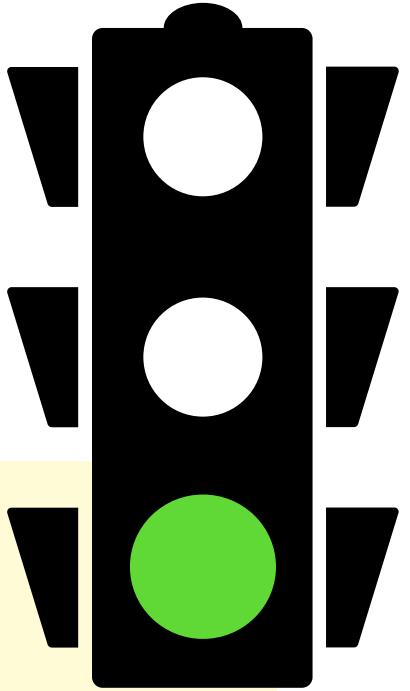
# Isogeny group action (contribution 7)

## Problem :

Given an elliptic curve  $E$  decide whether  $E$  is supersingular or ordinary

- we presented two algorithms from the literature for use in cryptography
- we improved upon these algorithms to give improvements over the state-of-the-art from CSIDH for the cases when  $E$  is expected to be supersingular or ordinary

# Isogeny group action (contribution 7)



## Problem :

Given an elliptic curve  $E$  decide whether  $E$  is supersingular or ordinary

- we presented two algorithms from the literature for use in cryptography
- we improved upon these algorithms to give improvements over the state-of-the-art from CSIDH for the cases when  $E$  is expected to be supersingular or ordinary

# Conclusion

# **Improved algorithms of post-quantum cryptographic group actions**

# **Improved algorithms of post-quantum cryptographic group actions**

# Improved algorithms of post-quantum cryptographic group actions

# Improved algorithms of

**post-quantum  
cryptographic  
group actions**

# Improved algorithms of post-quantum cryptographic group actions