27 May 2024, Eurocrypt 2024, Zurich, Switzerland

The supersingular Endomorphism Ring and One Endomorphism problems are equivalent

(speaker)

Aurel Page
Inria and Université de Bordeaux

Benjamin Wesolowski
CNRS and ENS de Lyon

Elliptic curves

An **elliptic curve** over $\mathbb{F}_q$ is:

a curve of the form

$$y^2 = x^3 + ax + b$$

Elliptic curves

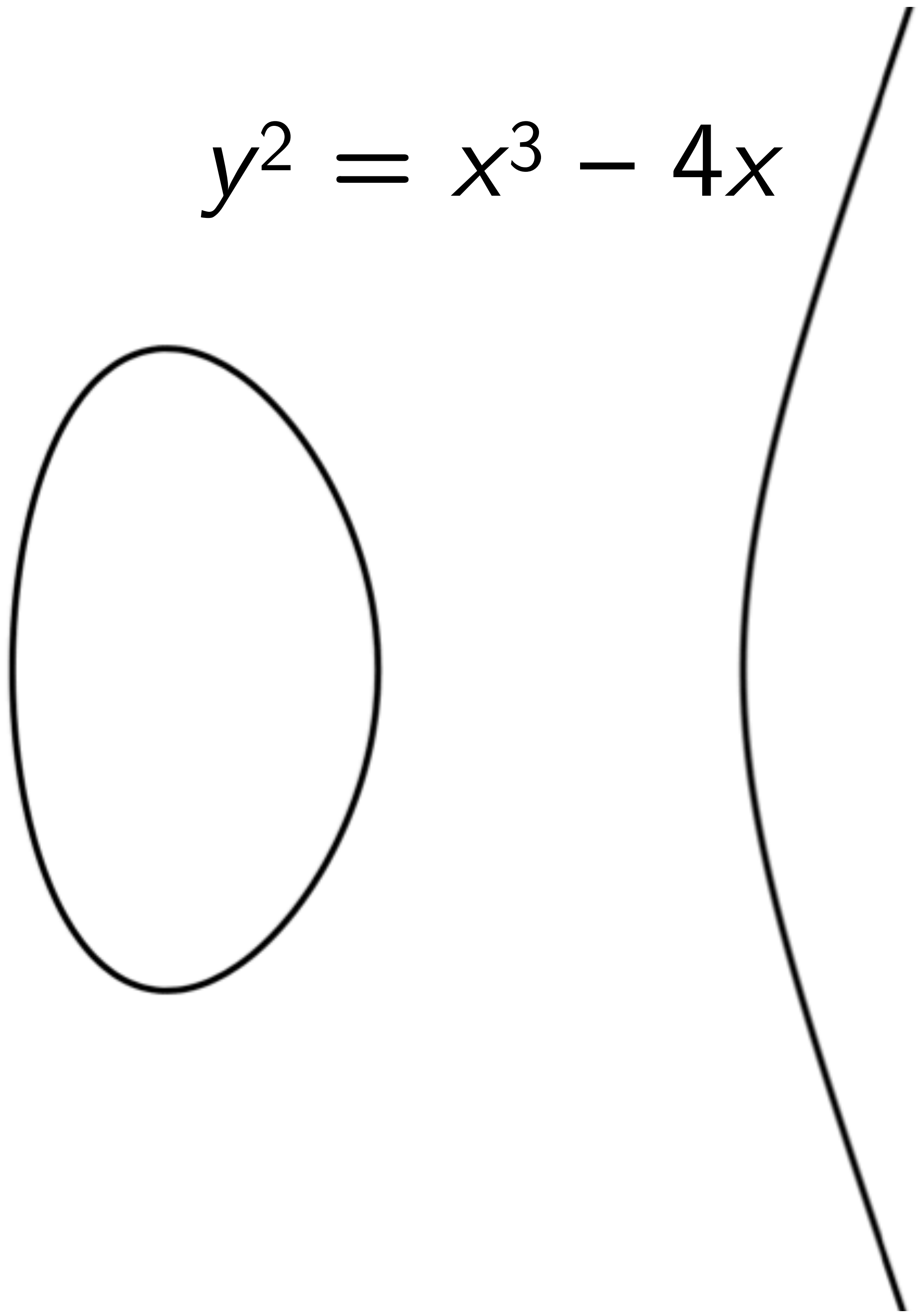
An **elliptic curve** over $\mathbb{F}_q$ is:

$y^2 = x^3 + x$

a curve of the form

$$y^2 = x^3 + ax + b$$

$y^2 = x^3 - 4x$



Isogenies

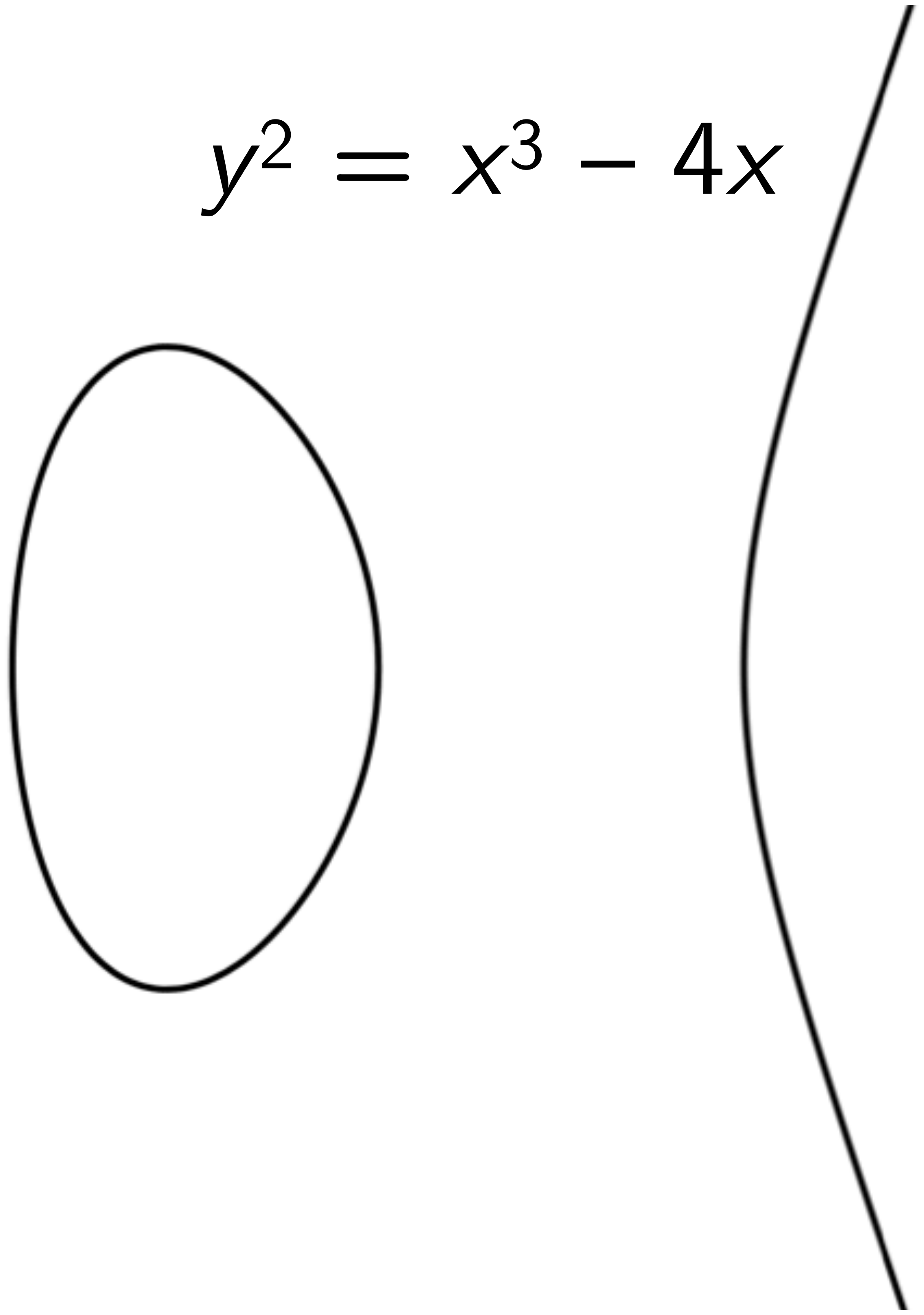
An **isogeny** is:

$y^2 = x^3 + x$

a map between two curves

$\varphi : E_1 \rightarrow E_2$

$y^2 = x^3 - 4x$

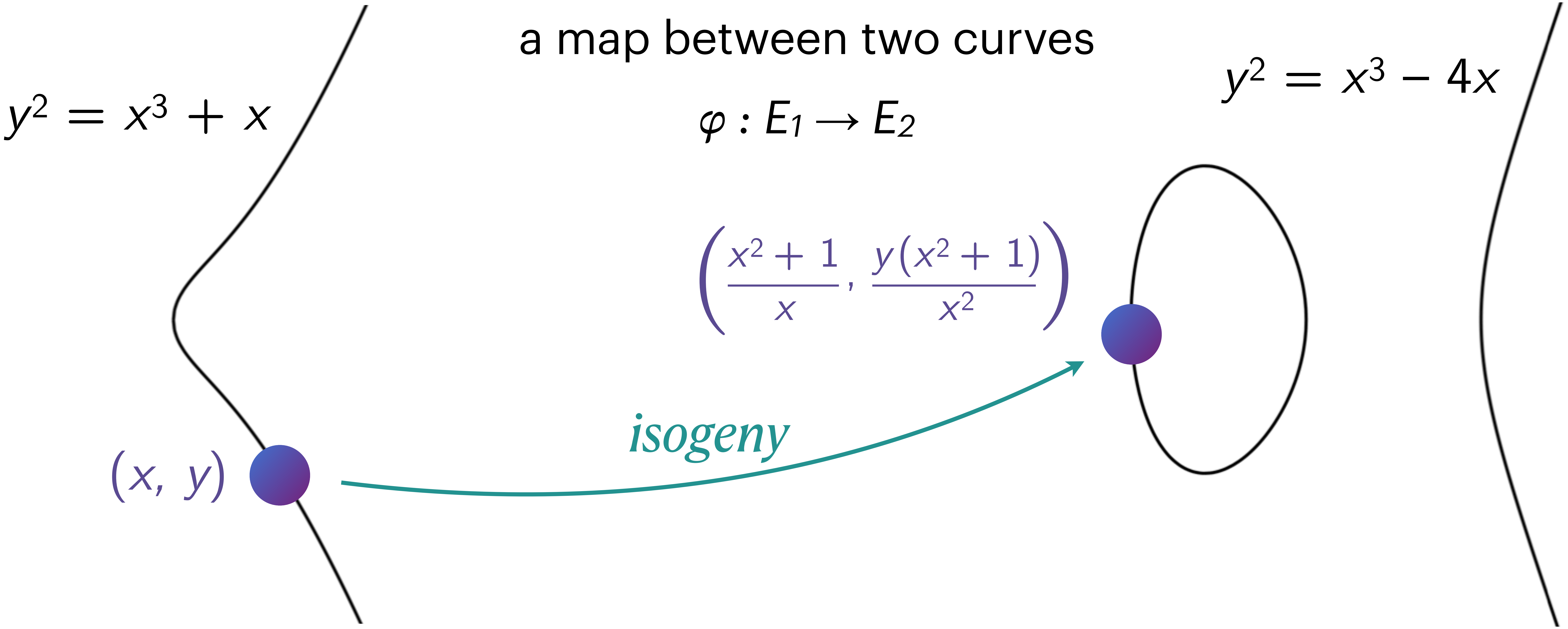


Isogenies

An **isogeny** is:

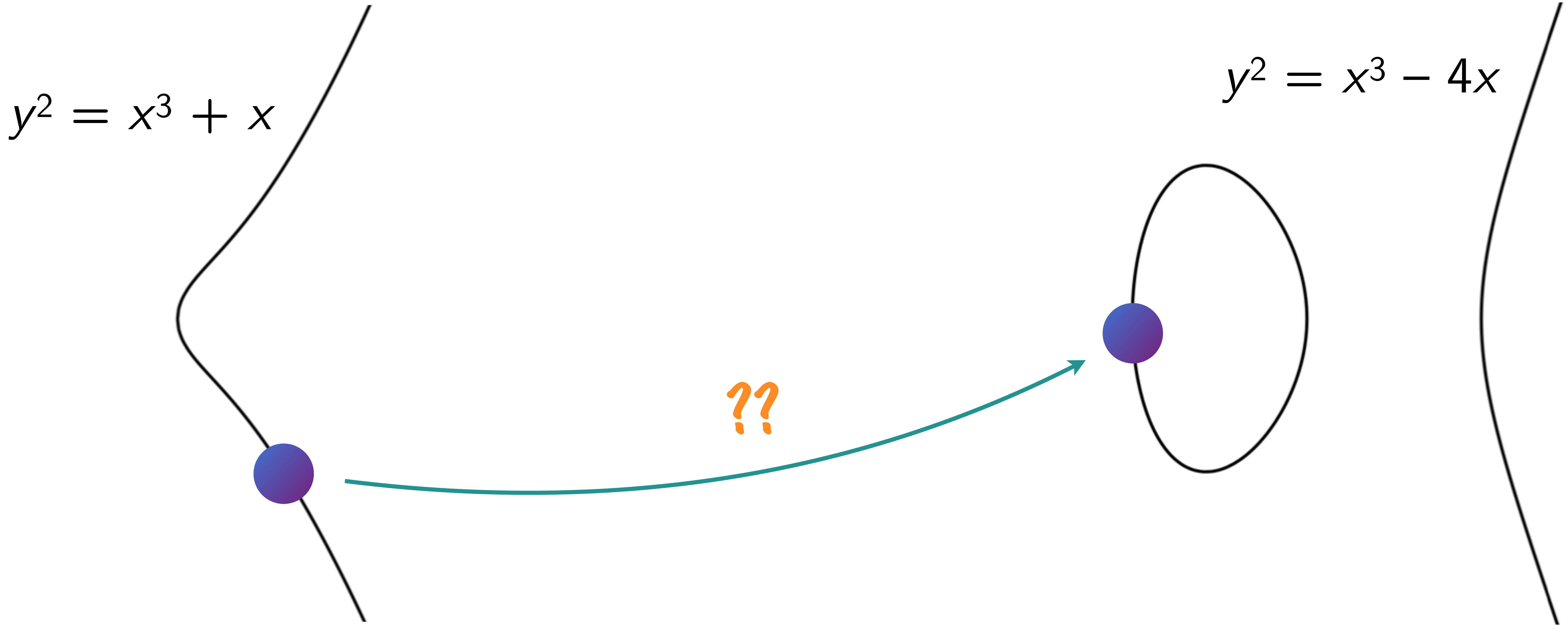
a map between two curves

$$\varphi : E_1 \longrightarrow E_2$$



The isogeny problem

Isogeny problem: Given two elliptic curves E_1 and E_2, find an isogeny $\varphi : E_1 \to E_2$



The isogeny problem

Isogeny problem: Given two elliptic curves E_1 and E_2, find an isogeny $\varphi : E_1 \to E_2$

A useful specialization:

ℓ-IsogenyPath: Given two elliptic curves E_1 and E_2, find an $\boldsymbol{\ell}$-isogeny paths $E_1 \to E_2$

The isogeny problem

Isogeny problem: Given two elliptic curves E_1 and E_2, find an isogeny $\varphi : E_1 \to E_2$

A useful specialization:

ℓ-IsogenyPath: Given two elliptic curves E_1 and E_2, find an ℓ-isogeny paths $E_1 \to E_2$

Special kind of easy-to-work-with isogenies

Isogeny-based cryptography

Expectations: cryptosystems as secure as **ℓ-IsogenyPath** is hard

ℓ-IsogenyPath = **Security of cryptosystems**

Hard even for quantum algorithms

Post-quantum cryptography

Isogeny-based cryptography

Reality: upper and lower bounds

??	$\leq$	Security of cryptosystems	$\leq$	ℓ-IsogenyPath

Isogeny-based cryptography

Reality: upper and lower bounds

??	$\leq$	**Security of cryptosystems**	$\leq$	**ℓ-IsogenyPath**

ℓ-IsogenyPath = CGL hash function (preimage)

Isogeny-based cryptography

Reality: upper and lower bounds

??	$\leq$	Security of cryptosystems	$\leq$	ℓ-IsogenyPath

ℓ-IsogenyPath = CGL hash function (preimage)

OneEnd $\leq$ CGL hash function (collision)

Isogeny-based cryptography

Reality: upper and lower bounds

??	$\leq$	Security of cryptosystems	$\leq$	ℓ-IsogenyPath

ℓ-IsogenyPath	=	CGL hash function (preimage)
OneEnd	$\leq$	CGL hash function (collision)
OneEnd	$\leq$	SQIsign (soundness)

Endomorphisms

OneEnd and EndRing

Endomorphism ring

An **endomorphism** of E is an isogeny $\varphi : E \to E$

Endomorphism ring

An **endomorphism** of E is an isogeny $\varphi : E \to E$

The **endomorphism ring** of E is $\mathrm{End}(E) = \{\varphi : E \to E\}$

Endomorphism ring

An **endomorphism** of E is an isogeny $\varphi : E \to E$

The **endomorphism ring** of E is $\mathrm{End}(E) = \{\varphi : E \to E\}$

- It contains $\mathbb{Z} \subset \mathrm{End}(E)$ (1 = identity, 2 = point doubling, -1 = negation...)

Endomorphism ring

An **endomorphism** of E is an isogeny $\varphi : E \to E$

The **endomorphism ring** of E is $\mathrm{End}(E) = \{\varphi : E \to E\}$

- It contains $\mathbb{Z} \subset \mathrm{End}(E)$ (1 = identity, 2 = point doubling, -1 = negation...)
- $(\mathrm{End}(E), +)$ is a **lattice** of dimension 2 or 4

Endomorphism ring

An **endomorphism** of E is an isogeny $\varphi : E \to E$

The **endomorphism ring** of E is $\mathrm{End}(E) = \{\varphi : E \to E\}$

- It contains $\mathbb{Z} \subset \mathrm{End}(E)$ (1 = identity, 2 = point doubling, -1 = negation...)
- $(\mathrm{End}(E), +)$ is a **lattice** of dimension 2 or 4

Endomorphism ring

An **endomorphism** of E is an isogeny $\varphi : E \to E$

The **endomorphism ring** of E is $\mathrm{End}(E) = \{\varphi : E \to E\}$

- It contains $\mathbb{Z} \subset \mathrm{End}(E)$ (1 = identity, 2 = point doubling, -1 = negation...)
- $(\mathrm{End}(E), +)$ is a **lattice** of dimension 2 or 4

Endomorphism ring

An **endomorphism** of E is an isogeny $\varphi : E \to E$

The **endomorphism ring** of E is $\mathrm{End}(E) = \{\varphi : E \to E\}$

- It contains $\mathbb{Z} \subset \mathrm{End}(E)$ (1 = identity, 2 = point doubling, -1 = negation...)
- $(\mathrm{End}(E), +)$ is a **lattice** of dimension 2 or 4

Ordinary elliptic curve

Supersingular elliptic curve

The endomorphism ring problem

EndRing: Given a supersingular E, find four endomorphisms generating $\mathrm{End}(E)$

The endomorphism ring problem

EndRing: Given a supersingular E, find four endomorphisms generating $\mathrm{End}(E)$

Theorem [W. – FOCS 2021]: **EndRing** is equivalent to ℓ-**IsogenyPath**
(assuming the Generalised Riemann Hypothesis)

Earlier **heuristic** reductions in:

[Petit, Lauter – preprint 2017] *Hard and Easy Problems for Supersingular Isogeny Graphs.*

[Eisenträger, Hallgren, Lauter, Morrison, Petit – Eurocrypt 2018] *Supersingular isogeny graphs and endomorphism rings: Reductions and solutions.*

The [one] endomorphism [ring] problem

EndRing: Given a supersingular E, find four endomorphisms generating $\mathrm{End}(E)$

OneEnd: Given a supersingular E, find a single endomorphism $\alpha \in \mathrm{End}(E) \setminus \mathbb{Z}$

The [one] endomorphism [ring] problem

EndRing: Given a supersingular E, find four endomorphisms generating $\mathrm{End}(E)$

OneEnd: Given a supersingular E, find a single endomorphism $\alpha \in \mathrm{End}(E) \setminus \mathbb{Z}$

Clearly, **OneEnd** $\leq$ **EndRing**...

The [one] endomorphism [ring] problem

EndRing: Given a supersingular E, find four endomorphisms generating $\mathrm{End}(E)$

OneEnd: Given a supersingular E, find a single endomorphism $\alpha \in \mathrm{End}(E) \setminus \mathbb{Z}$

Clearly, **OneEnd** $\leq$ **EndRing**...

Theorem (main result of this work): OneEnd is equivalent to **EndRing**, under probabilistic polynomial time reductions

Applications

of OneEnd = EndRing

New security reductions

Reality: upper and lower bounds

?? $\leq$ **Security of cryptosystems** $\leq$ **ℓ-IsogenyPath**

ℓ-IsogenyPath	$=$	CGL hash function (preimage)
OneEnd	$\leq$	CGL hash function (collision)
OneEnd	$\leq$	SQIsign (soundness)

New security reductions

Reality: upper and lower bounds

??	$\leq$	**Security of cryptosystems**	$\leq$	**ℓ-IsogenyPath**

= EndRing

ℓ-IsogenyPath	$=$	CGL hash function (preimage)
OneEnd	$\leq$	CGL hash function (collision)
OneEnd	$\leq$	SQIsign (soundness)

New security reductions

Reality: upper and lower bounds

??	$\leq$	Security of cryptosystems	$\leq$	ℓ-IsogenyPath

= EndRing

ℓ-IsogenyPath	=	CGL hash function (preimage)
EndRing = **OneEnd**	$\leq$	CGL hash function (collision)
OneEnd	$\leq$	SQIsign (soundness)

New security reductions

Reality: upper and lower bounds

??	$\leq$	**Security of cryptosystems**	$\leq$	**ℓ-IsogenyPath**

= EndRing

	ℓ-IsogenyPath	$=$	CGL hash function (preimage)
EndRing =	**OneEnd**	$\leq$	CGL hash function (collision)
	OneEnd	$\leq$	SQIsign (soundness)

Theorem (Application 1): CGL is **collision-resistant** if and only if **EndRing** is hard

Theorem (Application 2): SQIsign is **sound** if and only if **EndRing** is hard

EndRing is equivalent to Isogeny

Theorem (Application 3): EndRing is equivalent to the **Isogeny** problem

EndRing is equivalent to Isogeny

Theorem (Application 3): EndRing is equivalent to the **Isogeny** problem

Previous work:

- **Isogeny** ≤ **EndRing**: already known (assuming GRH **[W. – FOCS 2021]**)
- **EndRing** ≤ **Isogeny**: only known for special case **ℓ-IsogenyPath** (assuming GRH)

EndRing is equivalent to Isogeny

Theorem (Application 3): EndRing is equivalent to the **Isogeny** problem

Previous work:

- **Isogeny** ≤ **EndRing**: already known (assuming GRH **[W. – FOCS 2021]**)
- **EndRing** ≤ **Isogeny**: only known for special case **ℓ-IsogenyPath** (assuming GRH)

Idea of the proof: Suffices to show that **OneEnd** ≤ **Isogeny**

- Given E (an instance of **OneEnd**), sample random isogeny $\varphi : E \to F$, solve **Isogeny** to find $\psi : F \to E$, and return $\psi \circ \varphi$ (a solution of **OneEnd**)
- No need to assume GRH!

Solving EndRing

Theorem (Application 4): There is an algorithm for **EndRing** in time $\tilde{O}(p^{1/2})$

Solving EndRing

Theorem (Application 4): There is an algorithm for **EndRing** in time $\tilde{O}(p^{1/2})$

Previous work :

- Only known under GRH (see **[W. – FOCS 2021]**, or **[Fuselier, Iezzi, Kozek, Morrison, Namoijam – preprint 2023]**)
- Unconditionally, best known was $\tilde{O}(p)$ **[Kohel – PhD thesis 1996]**

Solving EndRing

Theorem (Application 4): There is an algorithm for **EndRing** in time $\tilde{O}(p^{1/2})$

Previous work :

- Only known under GRH (see **[W. – FOCS 2021]**, or **[Fuselier, Iezzi, Kozek, Morrison, Namoijam – preprint 2023]**)
- Unconditionally, best known was $\tilde{O}(p)$ **[Kohel – PhD thesis 1996]**

Idea of the proof:

- By the previous application, **EndRing** ≤ **Isogeny** (unconditionally!)
- Meet-in-the-middle solves **Isogeny** with complexity $\tilde{O}(p^{1/2})$

Sketch of the proof

Main ideas and obstacles

Reducing EndRing to OneEnd

Suppose we have an oracle $\mathcal{O}$ solving **OneEnd**

Let E be an instance of **EndRing**: we wish to find generators of $\mathrm{End}(E)$

Reducing EndRing to OneEnd

Suppose we have an oracle $\mathcal{O}$ solving **OneEnd**

Let E be an instance of **EndRing**: we wish to find generators of $\mathrm{End}(E)$

Idea 0: *Sample until you make it…*

1. For $i = 1, 2, \ldots$ call $\mathcal{O}(E)$, which returns some $\alpha_i \in \mathrm{End}(E) \setminus \mathbb{Z}$
2. As soon as $(\alpha_i)_i$ generates $\mathrm{End}(E)$, extract a basis and return it

Reducing EndRing to OneEnd

Suppose we have an oracle $\mathcal{O}$ solving **OneEnd**

Let E be an instance of **EndRing**: we wish to find generators of End(E)

Idea 0: *Sample until you make it…*

1. For i = 1, 2,… call $\mathcal{O}(E)$, which returns some $\alpha_i \in \mathrm{End}(E) \setminus \mathbb{Z}$
2. As soon as $(\alpha_i)_i$ generates End(E), extract a basis and return it 👍 Efficient linear algebra!

Reducing EndRing to OneEnd

Suppose we have an oracle $\mathcal{O}$ solving **OneEnd**

Let E be an instance of **EndRing**: we wish to find generators of $\mathrm{End}(E)$

Idea 0: *Sample until you make it…*

1. For $i = 1, 2, \ldots$ call $\mathcal{O}(E)$, which returns some $\alpha_i \in \mathrm{End}(E) \setminus \mathbb{Z}$ 👎 What if $\mathcal{O}(E)$ always returns the same α?
2. As soon as $(\alpha_i)_i$ generates $\mathrm{End}(E)$, extract a basis and return it 👍 Efficient linear algebra!

Reducing EndRing to OneEnd

Suppose we have an oracle $\mathcal{O}$ solving **OneEnd**

Let E be an instance of **EndRing**: we wish to find generators of $\mathrm{End}(E)$

Idea 0: *Sample until you make it…*

1. For $i = 1, 2, \ldots$ call $\mathcal{O}(E)$, which returns some $\alpha_i \in \mathrm{End}(E) \setminus \mathbb{Z}$

What if $\mathcal{O}(E)$ always returns the same α?

2. As soon as $(\alpha_i)_i$ generates $\mathrm{End}(E)$, extract a basis and return it

Efficient linear algebra!

Idea 1 [Eisenträger, Hallgren, Lauter, Morrison, Petit – Eurocrypt 2018]**:**

Randomize the oracle…

Enriching the oracle

Idea 1: Randomize the oracle

We construct a new oracle $\texttt{Rich}^{\mathcal{O}}$

Enriching the oracle

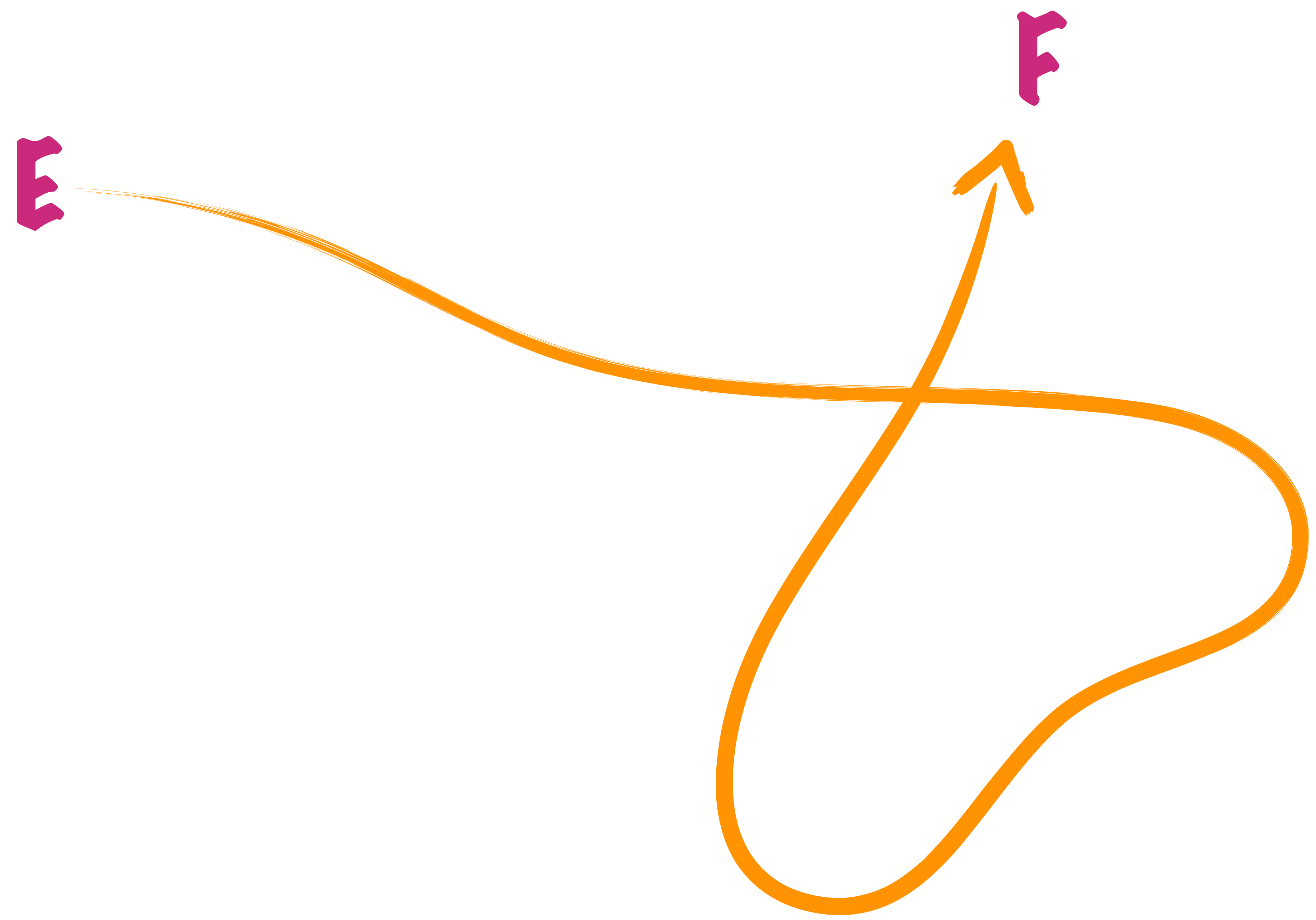
E

Idea 1: Randomize the oracle

We construct a new oracle $\texttt{Rich}^{\mathcal{O}}$

On input E:

Enriching the oracle



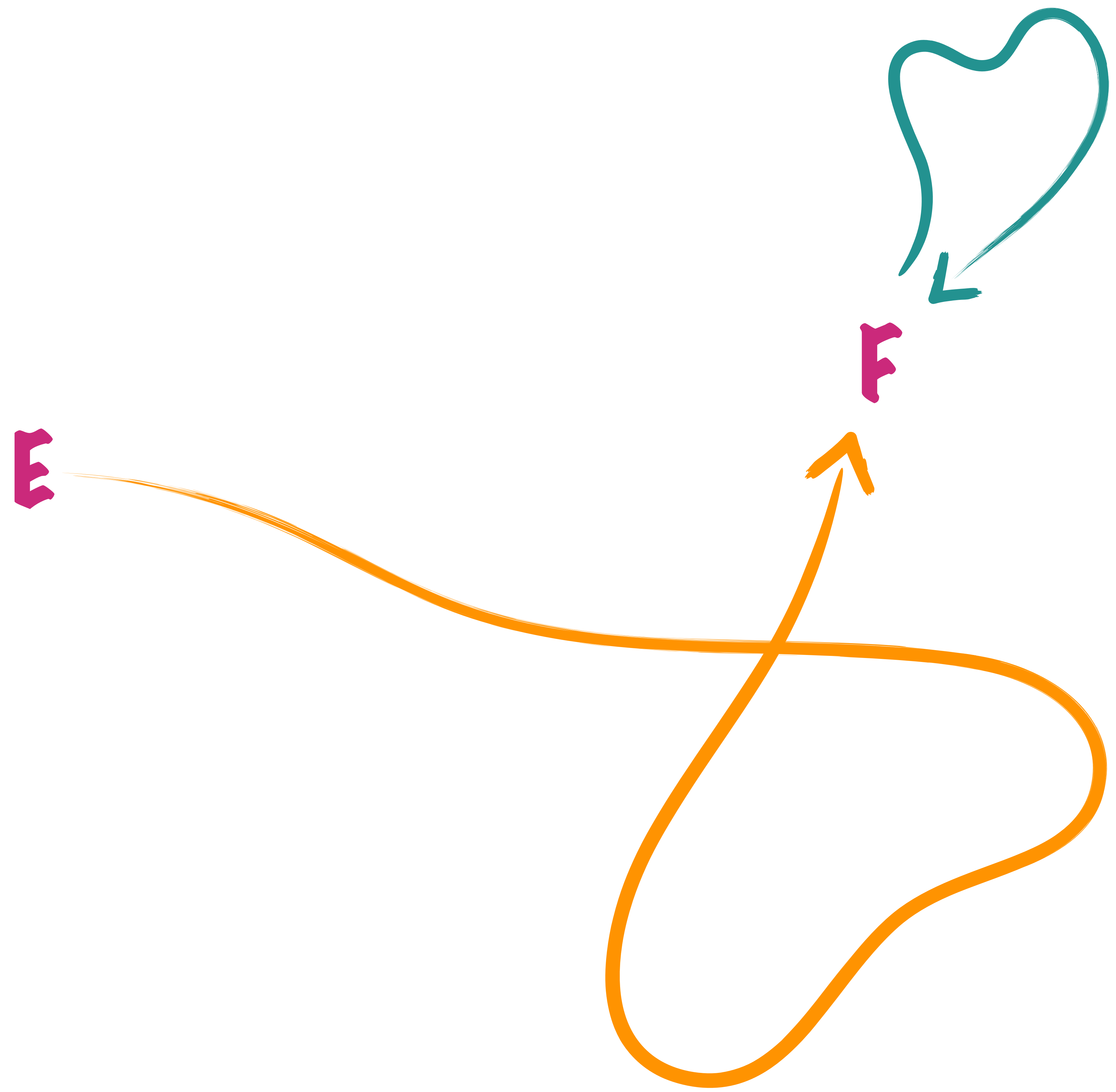
Idea 1: Randomize the oracle

We construct a new oracle $\mathtt{Rich}^{\mathcal{O}}$

On input E:

1. Sample a random isogeny $\varphi : E \to F$

Enriching the oracle



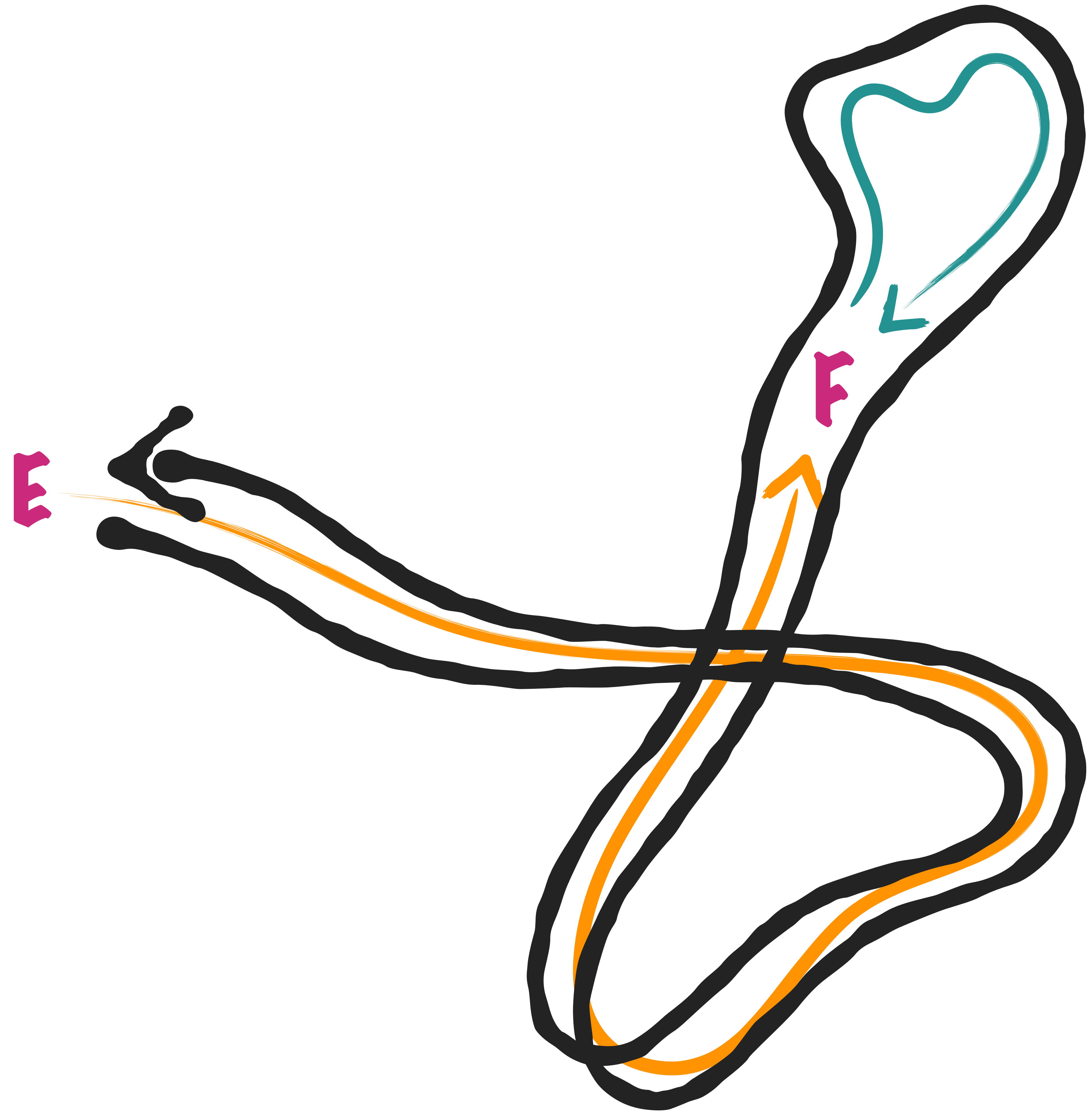
Idea 1: Randomize the oracle

We construct a new oracle $\texttt{Rich}^{\mathcal{O}}$

On input E:

1. Sample a random isogeny $\varphi : E \to F$
2. Call $\mathcal{O}(F)$ which returns $\alpha \in \mathrm{End}(F) \setminus \mathbb{Z}$

Enriching the oracle



Idea 1: Randomize the oracle

We construct a new oracle $\texttt{Rich}^{\mathcal{O}}$

On input E:

1. Sample a random isogeny $\varphi : E \to F$
2. Call $\mathcal{O}(F)$ which returns $\alpha \in \mathrm{End}(F) \setminus \mathbb{Z}$
3. Return $\hat{\varphi} \circ \alpha \circ \varphi \in \mathrm{End}(E) \setminus \mathbb{Z}$

Reducing EndRing to OneEnd

Idea 1: Randomize the oracle

1. For $i = 1, 2, \ldots$ call $\texttt{Rich}^{\mathcal{O}}(E)$, which returns some $\alpha_i \in \mathrm{End}(E) \setminus \mathbb{Z}$
2. As soon as $(\alpha_i)_i$ generates $\mathrm{End}(E)$, extract a basis and return it

Reducing EndRing to OneEnd

Idea 1: Randomize the oracle

1. For $i = 1, 2, \ldots$ call **`Rich`**$^{\mathcal{O}}(E)$, which returns some $\alpha_i \in \mathrm{End}(E) \setminus \mathbb{Z}$
2. As soon as $(\alpha_i)_i$ generates $\mathrm{End}(E)$, extract a basis and return it

Heuristic claim [Eisenträger, Hallgren, Lauter, Morrison, Petit – Eurocrypt 2018]**:**
Rich$^{\mathcal{O}}$ is "random enough": it rapidly produces a generating set

Reducing EndRing to OneEnd

Idea 1: Randomize the oracle

1. For $i = 1, 2, \ldots$ call $\mathtt{Rich}^{\mathcal{O}}(E)$, which returns some $\alpha_i \in \mathrm{End}(E) \setminus \mathbb{Z}$

2. As soon as $(\alpha_i)_i$ generates $\mathrm{End}(E)$, extract a basis and return it

Heuristic claim [Eisenträger, Hallgren, Lauter, Morrison, Petit – Eurocrypt 2018]**:**
$\mathrm{Rich}^{\mathcal{O}}$ is "random enough": it rapidly produces a generating set

Problem: It **fails**. There exist oracles $\mathcal{O}$ for which the algorithm does not terminate

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

Theorem 1: The probability distribution of $\mathtt{Rich}^{\mathcal{O}}(E)$ is stable under conjugation

In essence: any output α is as likely as any conjugate $\beta^{-1}\alpha\beta$

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

Theorem 1: The probability distribution of $\mathtt{Rich}^{\mathcal{O}}(E)$ is stable under conjugation

In essence: any output α is as likely as any conjugate $\beta^{-1}\alpha\beta$

Theorem 2: Subrings of $\mathrm{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \mathrm{End}(E)$ for $M \in \mathbb{Z}$

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

Theorem 1: The probability distribution of $\mathtt{Rich}^{\mathcal{O}}(E)$ is stable under conjugation

In essence: any output α is as likely as any conjugate $\beta^{-1}\alpha\beta$

Theorem 2: Subrings of $\mathrm{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \mathrm{End}(E)$ for $M \in \mathbb{Z}$

Conclusion: The algorithm **eventually** generates a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

Theorem 1: The probability distribution of $\mathtt{Rich}^{\mathcal{O}}(E)$ is stable under conjugation

In essence: any output α is as likely as any conjugate $\beta^{-1}\alpha\beta$

Theorem 2: Subrings of $\mathrm{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \mathrm{End}(E)$ for $M \in \mathbb{Z}$

Conclusion: The algorithm **eventually** generates a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$

From a generating set of $\mathbb{Z} + M \cdot \mathrm{End}(E)$, one can find a basis of $\mathrm{End}(E)$ 👍

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

Theorem 1: The probability distribution of $\mathtt{Rich}^{\mathcal{O}}(E)$ is stable under conjugation

In essence: any output α is as likely as any conjugate $\beta^{-1}\alpha\beta$

Theorem 2: Subrings of $\mathrm{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \mathrm{End}(E)$ for $M \in \mathbb{Z}$

Conclusion: The algorithm **eventually** generates a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$

From a generating set of $\mathbb{Z}$ + $M \cdot \mathrm{End}(E)$, one can find a basis of $\mathrm{End}(E)$ 👍

"Eventually" = exponential time 👎

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

The tough part!

Theorem 1: The probability distribution of $\mathtt{Rich}^{\mathcal{O}}(E)$ is stable under conjugation

In essence: any output α is as likely as any conjugate $\beta^{-1}\alpha\beta$

Theorem 2: Subrings of $\mathrm{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \mathrm{End}(E)$ for $M \in \mathbb{Z}$

Conclusion: The algorithm **eventually** generates a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$

From a generating set of $\mathbb{Z}$ + M·End(E), one can find a basis of End(E) 👍

"Eventually" = exponential time 👎

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

The tough part!

Theorem 1: The probability distribution of $\texttt{Rich}^{\mathcal{O}}(E)$ is stable under conjugation

In essence: any output α is as likely as any conjugate $\beta^{-1}\alpha\beta$

Deuring correspondence

Theorem 2: Subrings of $\mathrm{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \mathrm{End}(E)$ for $M \in \mathbb{Z}$

Conclusion: The algorithm **eventually** generates a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$

From a generating set of $\mathbb{Z}$ + M·End(E), one can find a basis of End(E) 👍

"Eventually" = exponential time 👎

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

The tough part!

Theorem 1: The probability distribution of $\mathtt{Rich}^{\mathcal{O}}(E)$ is stable under conjugation

In essence: any output α is as likely as any conjugate $\beta^{-1}\alpha\beta$

Deuring correspondence

+

Jacquet-Langlands correspondence

Theorem 2: Subrings of $\mathrm{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \mathrm{End}(E)$ for $M \in \mathbb{Z}$

Conclusion: The algorithm **eventually** generates a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$

From a generating set of $\mathbb{Z}$ + M·End(E), one can find a basis of End(E) 👍

"Eventually" = exponential time 👎

Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

The tough part!

Theorem 1: The probability distribution of $\mathbf{Rich}^{\mathcal{O}}(E)$ is stable under conjugation

In essence: any output α is as likely as any conjugate $\beta^{-1}\alpha\beta$

Deuring correspondence

+

Theorem 2: Subrings of $\mathrm{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \mathrm{End}(E)$ for $M \in \mathbb{Z}$

Jacquet-Langlands correspondence

+

Conclusion: The algorithm **eventually** generates a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$

Deligne's bound on coefficients of modular forms

From a generating set of $\mathbb{Z}$ + M·End(E), one can find a basis of End(E) 👍

"Eventually" = exponential time 👎

Reducing EndRing to OneEnd

Outline of the reduction:

1. Initialize $S = \{ 1 \}$
2. While S does not generate a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$, do:
    3. Sample $\alpha \leftarrow$ **`Rich`**$^{\mathcal{O}}(E)$
    4. $\alpha \leftarrow$ **`LazyReduce`**(α) **(Idea 3)**
    5. Add α to S
6. Extract from S a basis of $\mathrm{End}(E)$, and return it

Reducing EndRing to OneEnd

Outline of the reduction:

1. Initialize $S = \{ 1 \}$
2. While S does not generate a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$, do:
   3. Sample $\alpha \leftarrow \texttt{Rich}^{\mathcal{O}}(E)$
   4. $\alpha \leftarrow \texttt{LazyReduce}(\alpha)$ **(Idea 3)**
   5. Add α to S
6. Extract from S a basis of $\mathrm{End}(E)$, and return it

Terminates! 👍

Reducing EndRing to OneEnd

Outline of the reduction:

1. Initialize $S = \{ 1 \}$
2. While S does not generate a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$, do:
    3. Sample $\alpha \leftarrow \texttt{Rich}^{\mathcal{O}}(E)$
    4. $\alpha \leftarrow \texttt{LazyReduce}(\alpha)$ (Idea 3)
    5. Add α to S
6. Extract from S a basis of $\mathrm{End}(E)$, and return it

Terminates! 👍

In exponential time... 👎

Reducing EndRing to OneEnd

Outline of the reduction:

1. Initialize $S = \{ 1 \}$
2. While S does not generate a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$, do:
    3. Sample $\alpha \leftarrow \texttt{Rich}^{\mathcal{O}}(E)$
    4. $\alpha \leftarrow \texttt{LazyReduce}(\alpha)$ **(Idea 3)**
    5. Add α to S
6. Extract from S a basis of $\mathrm{End}(E)$, and return it

Reducing EndRing to OneEnd

Outline of the reduction:

1. Initialize $S = \{ 1 \}$
2. While S does not generate a ring of the form $\mathbb{Z} + M \cdot \mathrm{End}(E)$, do:
    3. Sample $\alpha \leftarrow \mathtt{Rich}^{\mathcal{O}}(E)$
    4. $\alpha \leftarrow \mathtt{LazyReduce}(\alpha)$ **(Idea 3)**
    5. Add α to S
6. Extract from S a basis of $\mathrm{End}(E)$, and return it

Polynomial time!

OneEnd to find them all