Computational foundations of

Isogeny-based cryptography

Benjamin Wesolowski, CNRS and ENS Lyon

October 2024, *25th Workshop on Elliptic Curve Cryptography*, Taipei, Taiwan

Isogenies

Elliptic curves, isogenies, isogeny graphs

Elliptic curves

- In crypto, we use elliptic curves over a **finite field**
- Elliptic curves are **groups**: you can add points together!

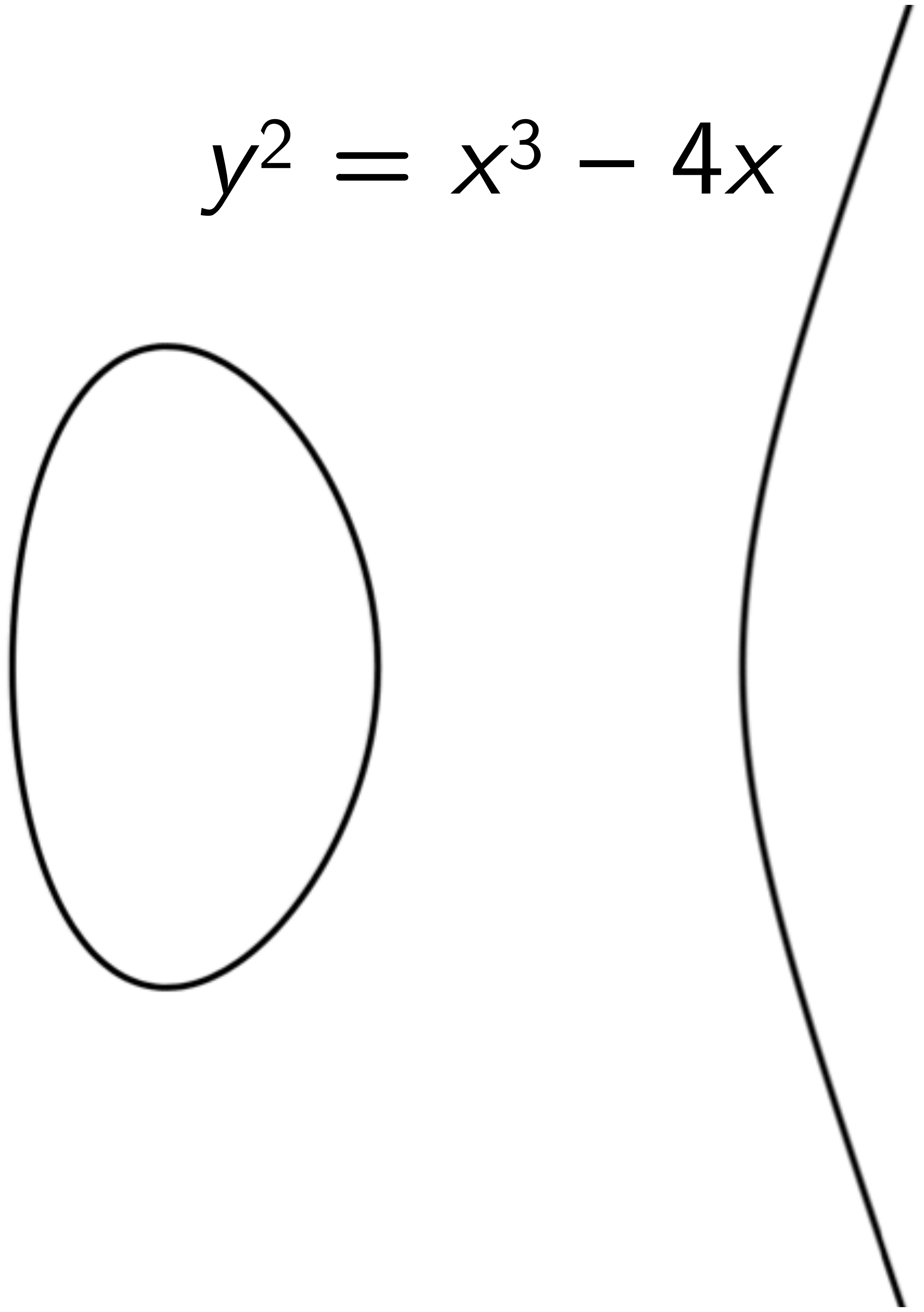
equations of the form

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

$y^2 = x^3 + x$

E_1

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



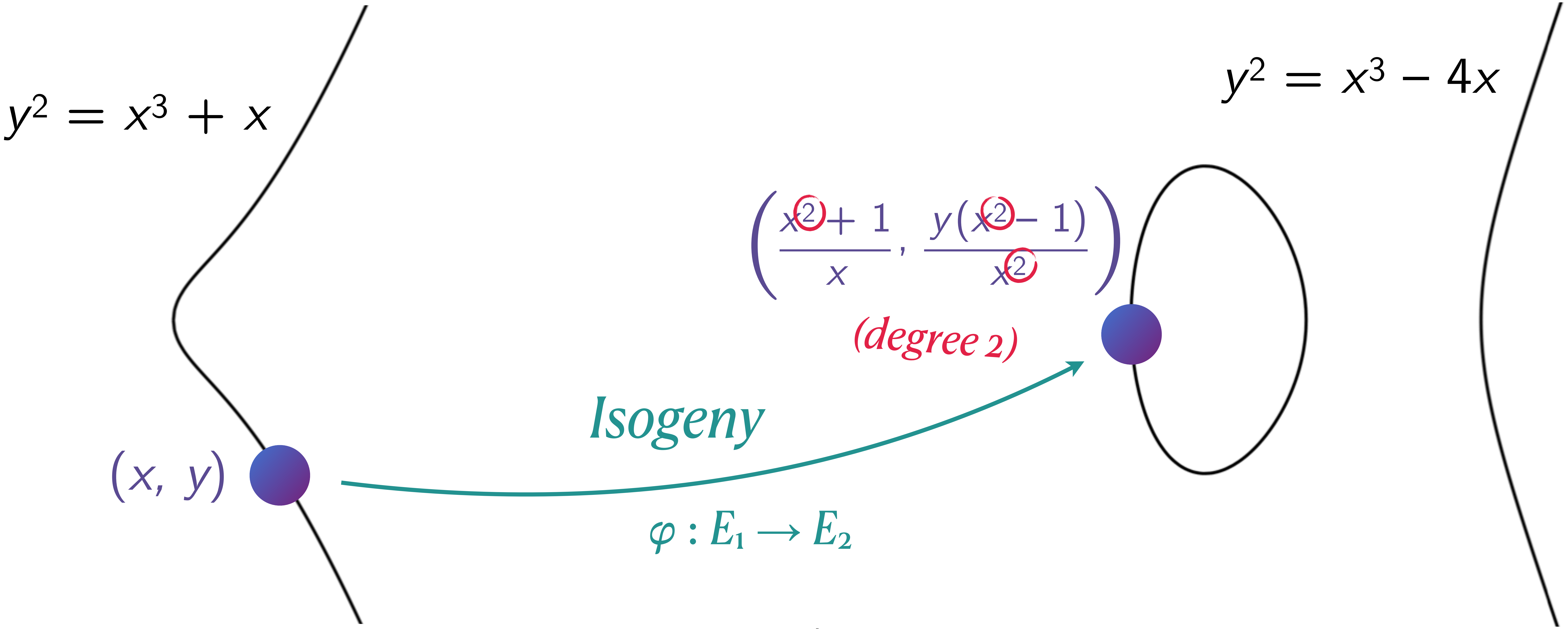
E_2

Elliptic curves

- In crypto, we use elliptic curves over a **finite field**
- Elliptic curves are **groups**: you can add points together!

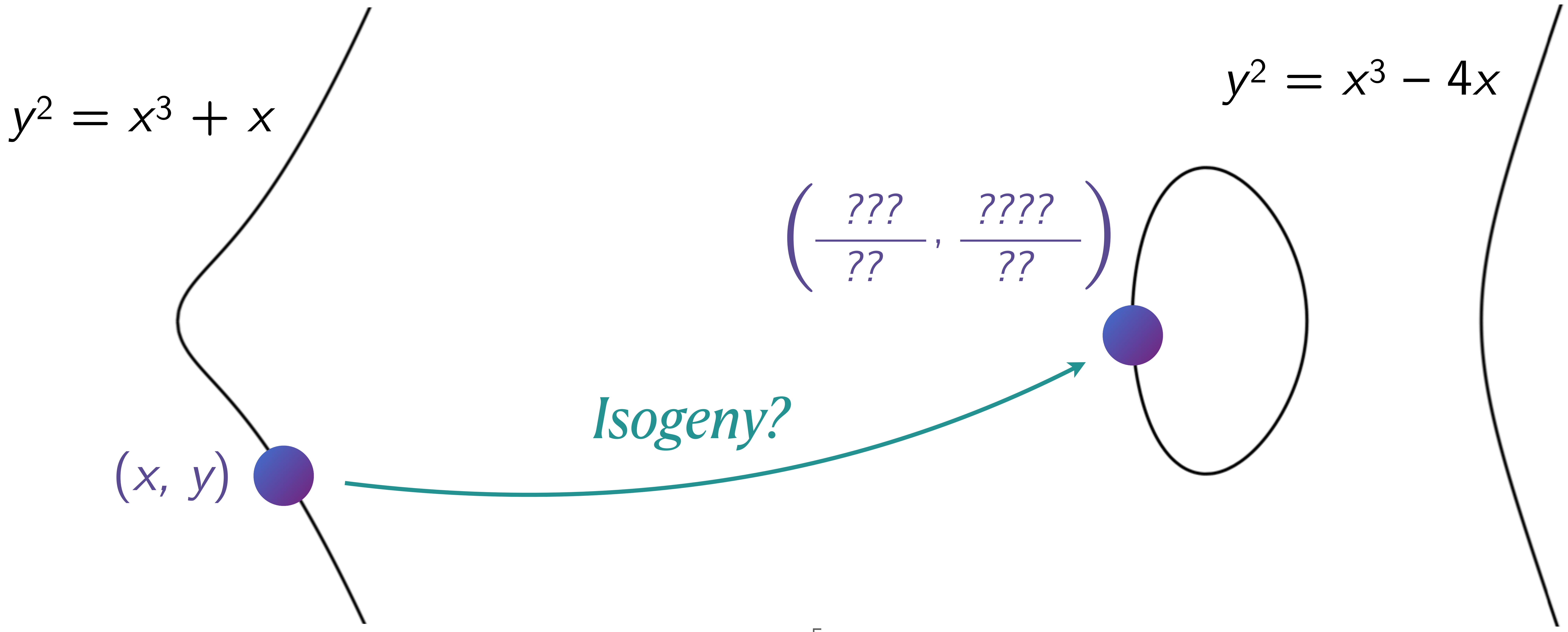
- Isogenies are group **homomorphisms**
- **Degree** = size of kernel

Sometimes, there is a formula to transform solutions from one equation to another



The *Isogeny* problem

Given E_1 and E_2 find an isogeny $\varphi : E_1 \to E_2$



The *Isogeny* problem

Given E_1 and E_2 find an isogeny $\varphi : E_1 \to E_2$

Hope: cryptosystems as secure as isogeny problem is hard

The isogeny problem = **Security of cryptosystems**

Hard even for quantum algorithms

Post-quantum cryptography

The *Isogeny* problem

Given E_1 and E_2 find an isogeny $\varphi : E_1 \to E_2$

solution typically has degree $\approx 2^{256}$

- The solution φ is an isogeny...
- How to represent an isogeny?

$$(x,\ y) \longmapsto \left(\frac{x^2+1}{x}, \frac{y(x^2-1)}{x^2}\right)$$

(degree 2)

fine for small degree...

- Build "big" isogenies as *formal compositions* of "small" ones

$$\deg(\varphi \circ \psi) = \deg(\varphi) \cdot \deg(\psi)$$

The *Isogeny* problem

Given E_1 and E_2 find an isogeny $\varphi : E_1 \to E_2$

solution typically has degree ≈ 2^{256}

- The solution φ is an isogeny...
- How to represent an isogeny?

$$(x,\ y) \quad \longmapsto \quad \left(\frac{x^2+1}{x},\ \frac{y(x^2-1)}{x^2} \right)$$

(degree 2)

fine for small degree...

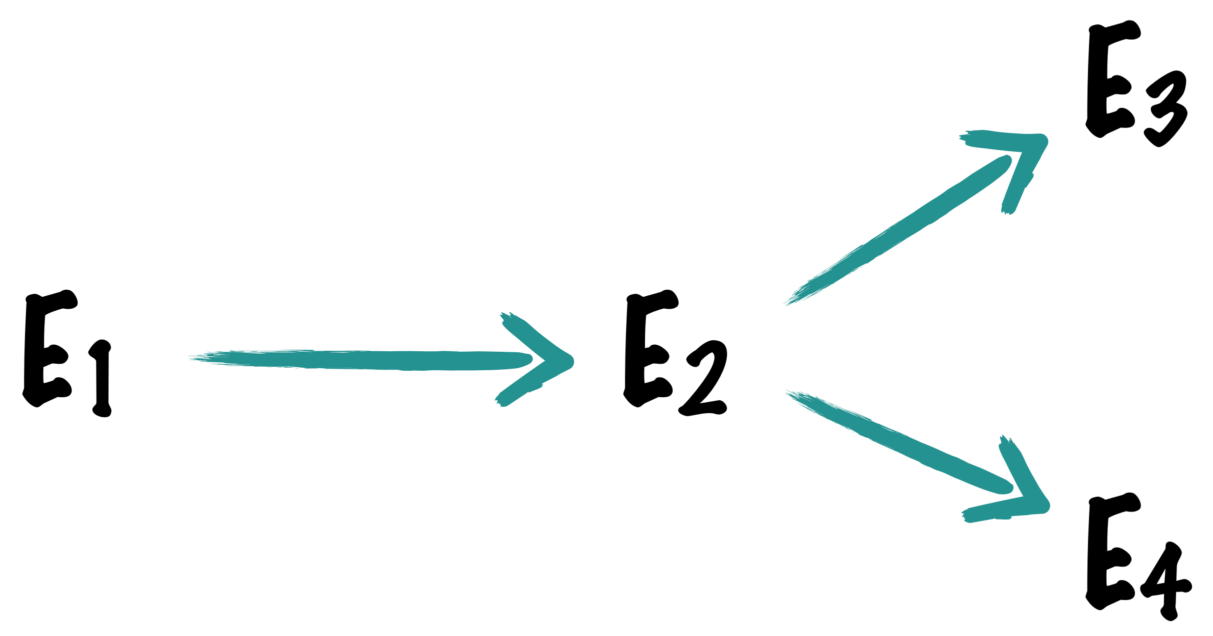
- Build "big" isogenies as *formal compositions* of "small" ones

$$E_1 \xrightarrow{2} E_2 \xrightarrow{2} E_3 \xrightarrow{2} \dots \xrightarrow{2} E_{257}$$

degree 2^{256}

Isogeny graph

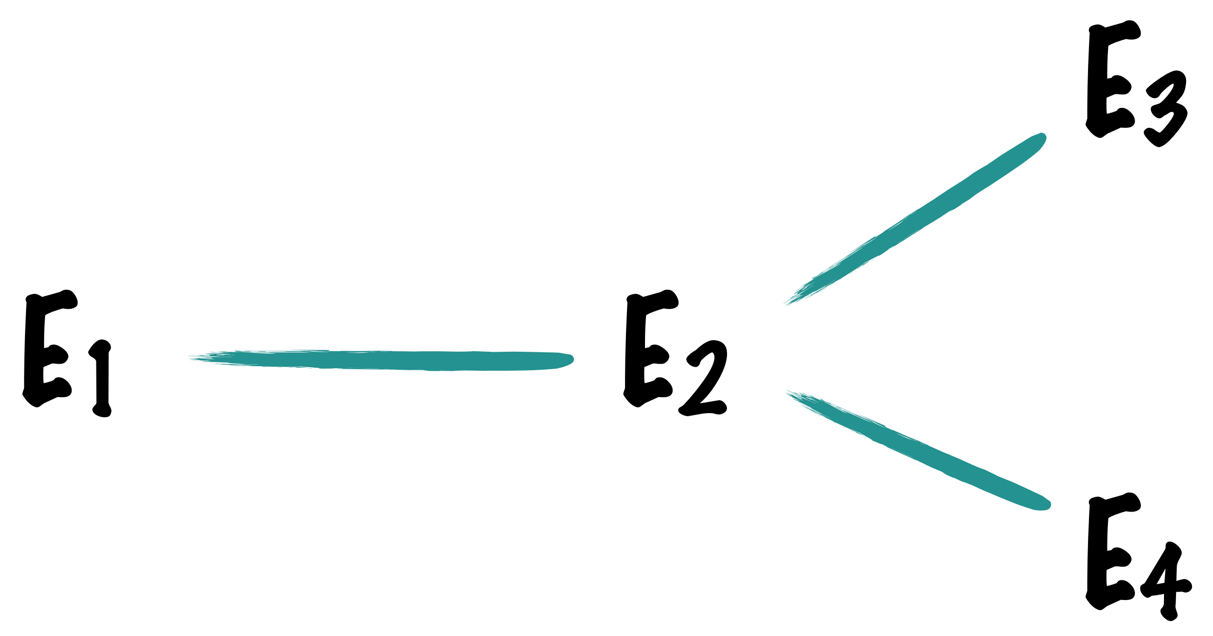
- Fix small ℓ (say, $\ell = 2$). Can easily compute ℓ-isogenies



an isogeny of degree ℓ = an edge in a graph

Isogeny graph

- Fix small ℓ (say, $\ell = 2$). Can easily compute ℓ-isogenies

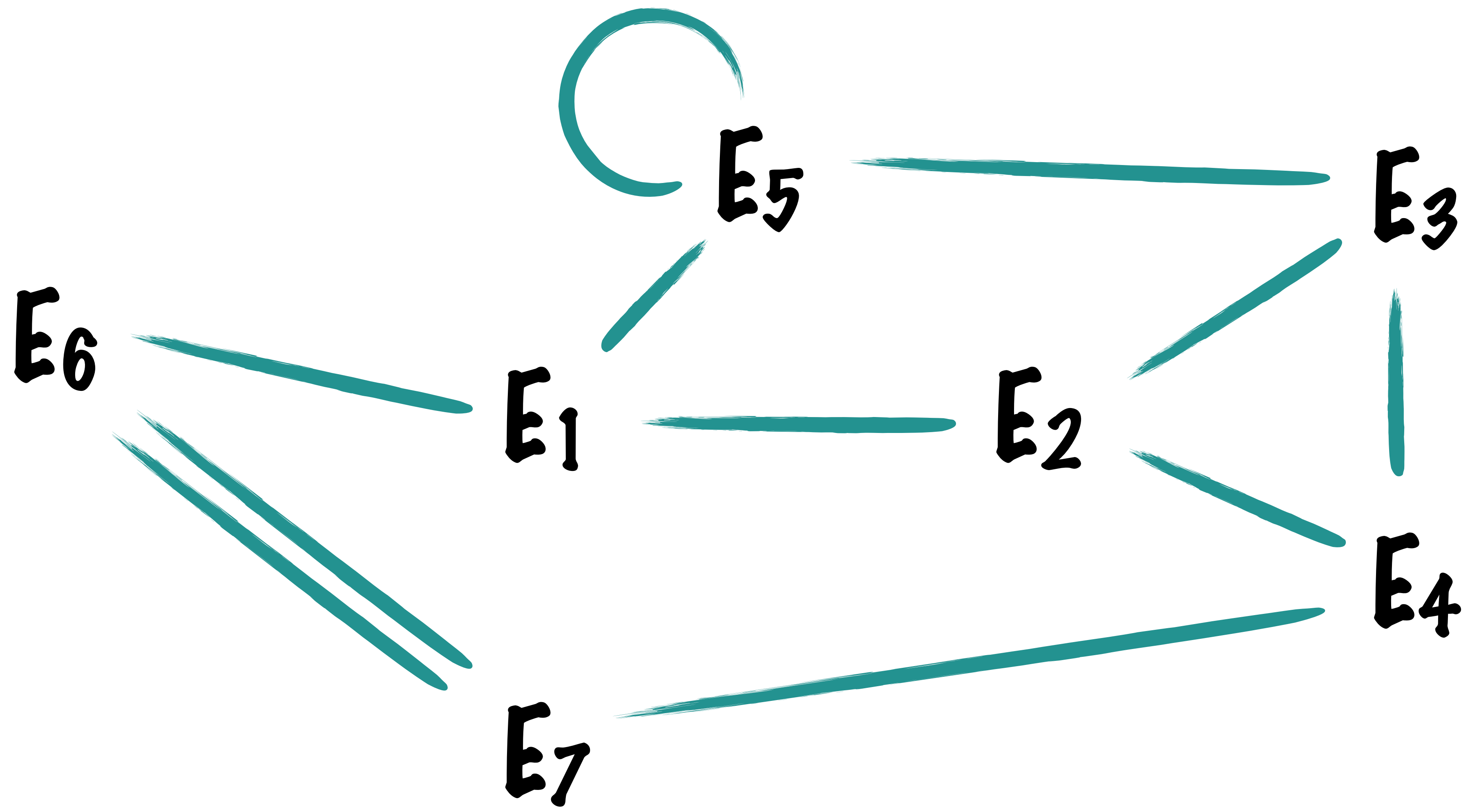


an isogeny of degree ℓ = an edge in a graph

$\exists\ \ell$-isogeny $E_1 \to E_2 \Rightarrow \exists\ \ell$-isogeny $E_2 \to E_1$

Isogeny graph

- Fix small ℓ (say, $\ell = 2$). Can easily compute ℓ-isogenies
- **The (supersingular...) ℓ-isogeny graph**



- $(\ell + 1)$-regular, **connected**, finite (all supersingular curves are defined over $\mathbb{F}_{p^2}$)

205
206
166
165
164
167
163
168
162
202
169
161
201
103
102
101
100
99
98
160
170
104
105
97
96
210
171
106
95
159
94
55
54
53
52
172
200
158
51
107
56
173
93
57
50
157
108
58
28
27
92
174
109
49
26
91
156
59
90
110
13
30
48
155
60
111
175
14
25
89
199
154
12
112
61
31
6
47
88
198
15
113
153
24
114
62
87
152
5
46
176
115
2
11
151
32
7
86
63
116
16
23
45
85
150
1
3
177
117
33
64
44
84
196
149
4
22
118
10
195
178

The ℓ-isogeny path problem

The ℓ-*IsogenyPath* problem

Given E_1 and E_2 (supersingular) find an ℓ-isogeny path from E_1 to E_2

- Path finding in a graph of size ≈ $p/12$
- Hard for supersingular curves! Best known algorithm = generic graph algorithm
- Typical meaning of ***"the isogeny problem"***

Isogeny-based cryptography

Computational problems and cryptosystems

Isogeny-based cryptography

Hope: cryptosystems as secure as isogeny problem is hard

The isogeny problem = **Security of cryptosystems**

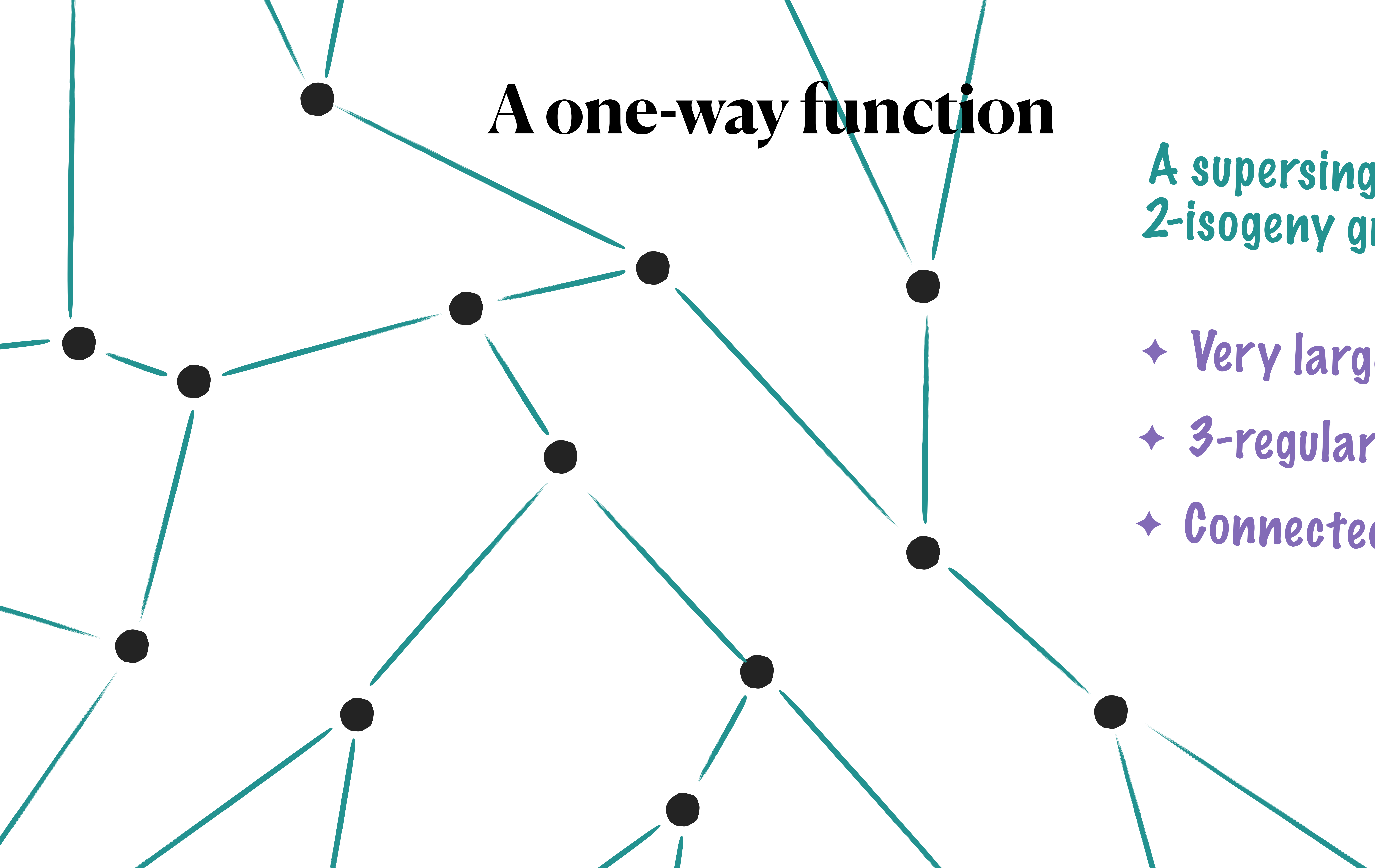
Hard even for quantum algorithms

Post-quantum cryptography

A one-way function

- **One-way function:** a function $f : X \to Y$ which is
  - ➡ **Easy to evaluate:** given $x \in X$, it is easy to compute $f(x)$
  - ➡ **Hard to invert:** given $y \in Y$, it is hard to find some $x \in X$ such that $f(x) = y$

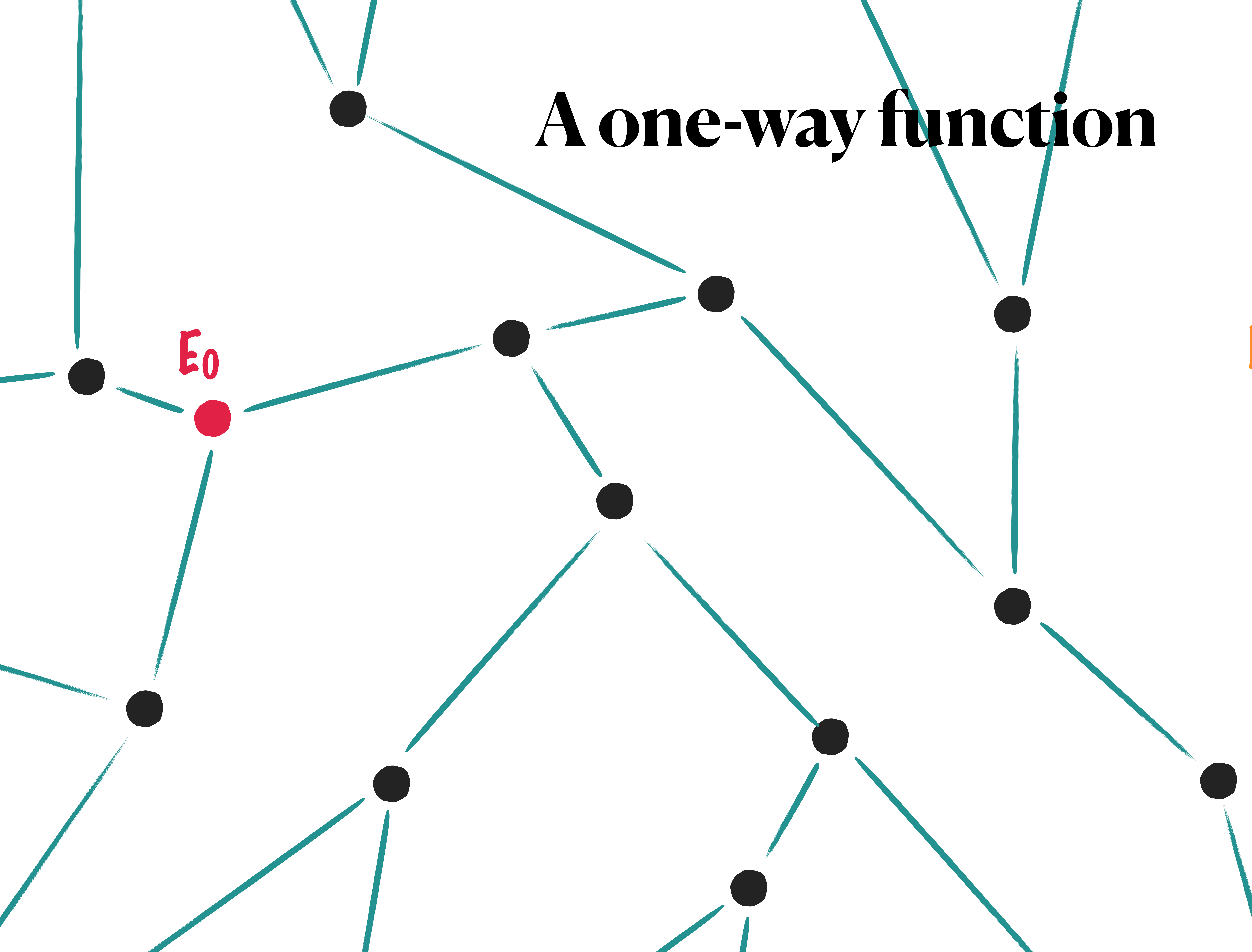
A one-way function



A supersingular 2-isogeny graph

- Very large
- 3-regular
- Connected

A one-way function

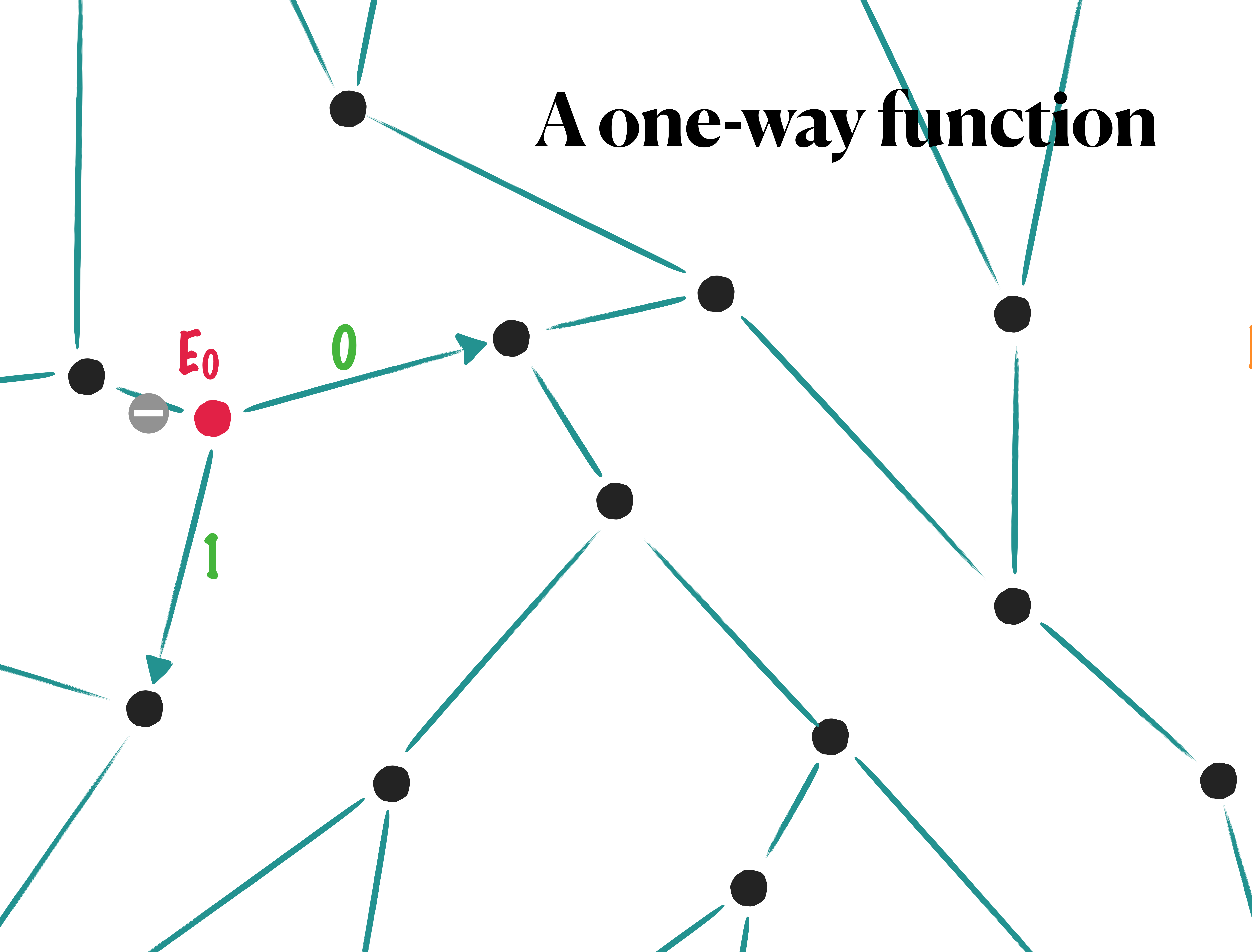


Fix a reference vertex E_0

Input: a message m in $\{0,1\}^*$

$m = 0001$

A one-way function

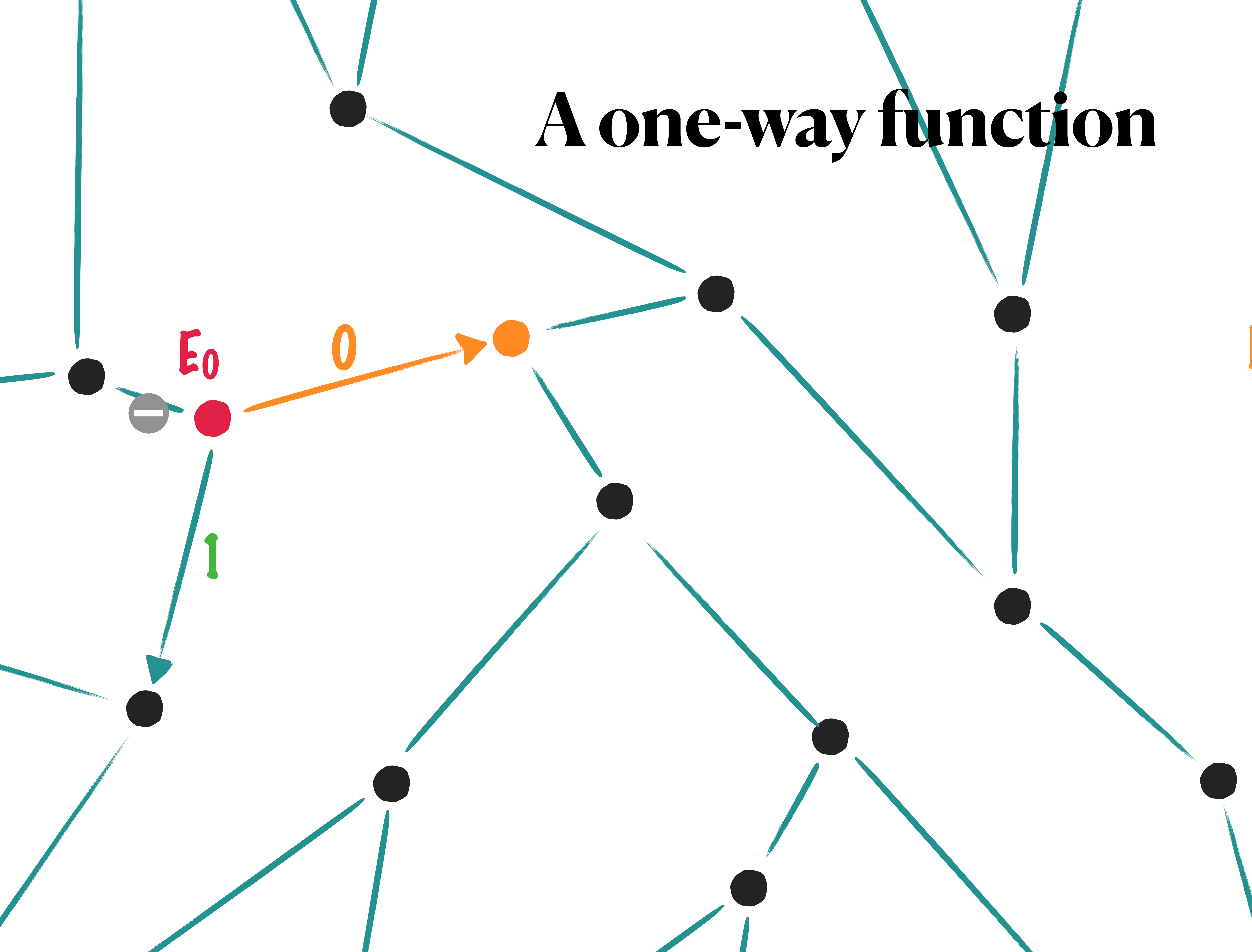


Fix a reference vertex E_0

Input: a message m in $\{0,1\}^*$

$m = 0001$

A one-way function

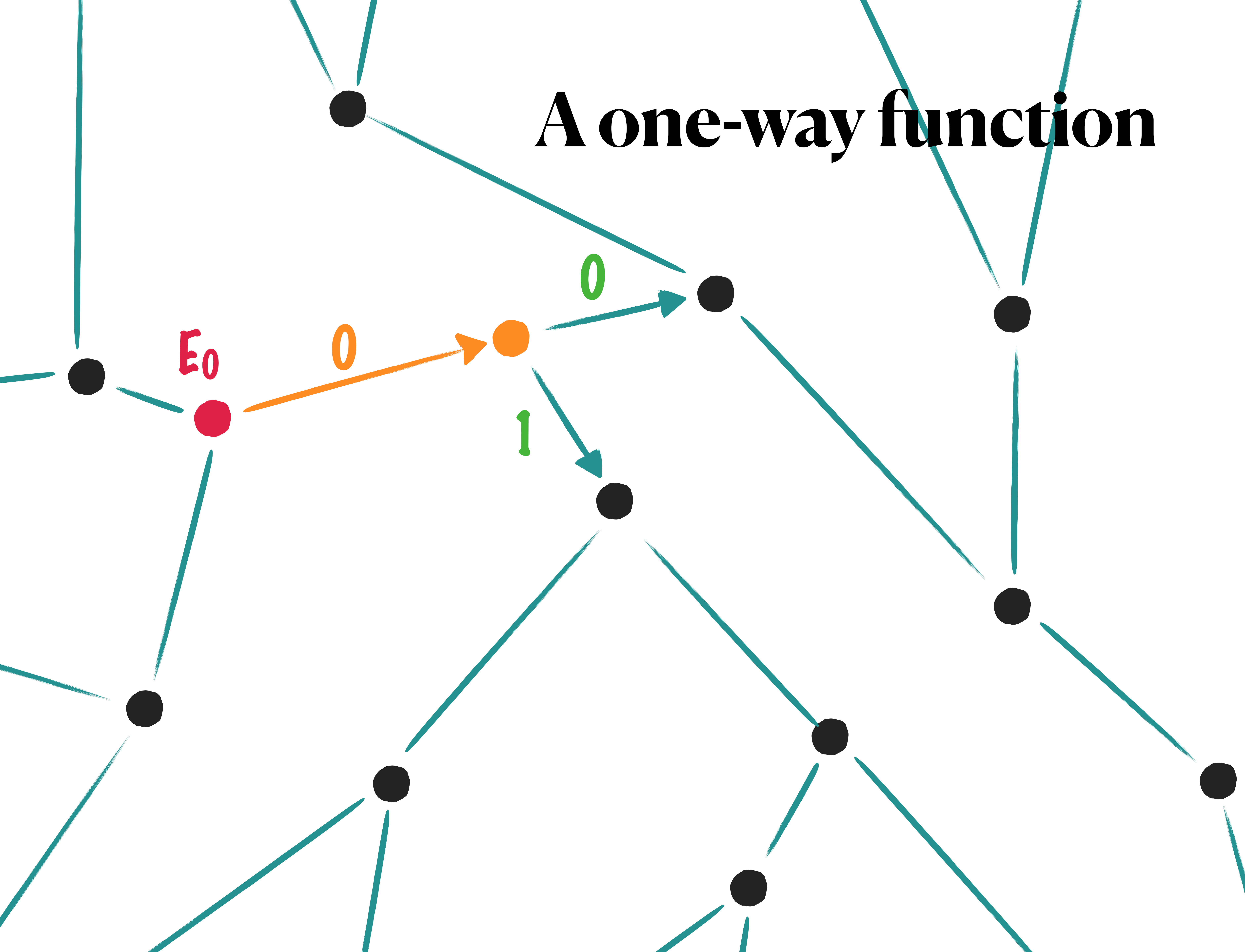


Fix a reference vertex E_0

Input: a message m in $\{0,1\}^*$

$m = 0001$

A one-way function

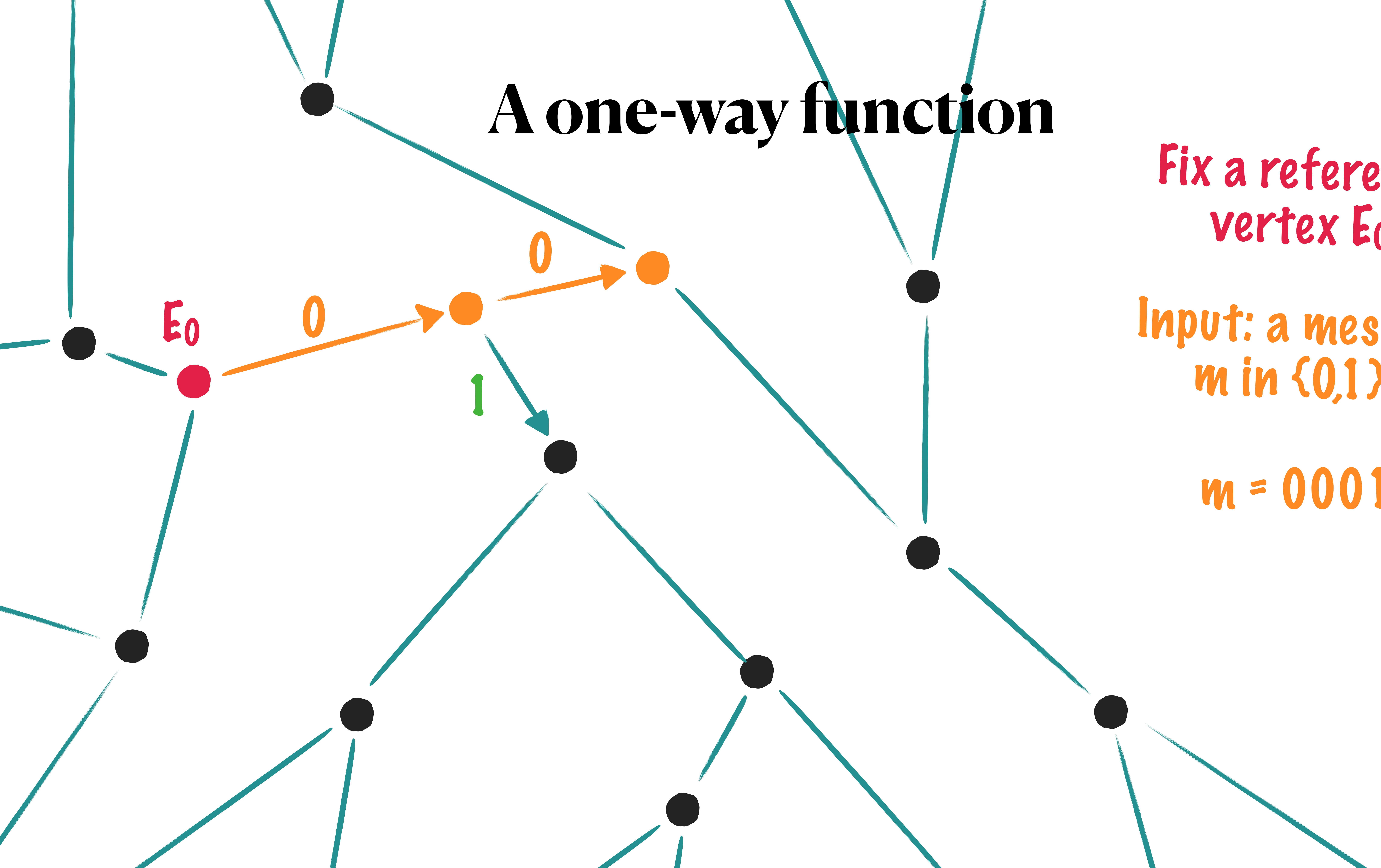


Fix a reference vertex E_0

Input: a message m in $\{0,1\}^*$

$m = 0001$

A one-way function

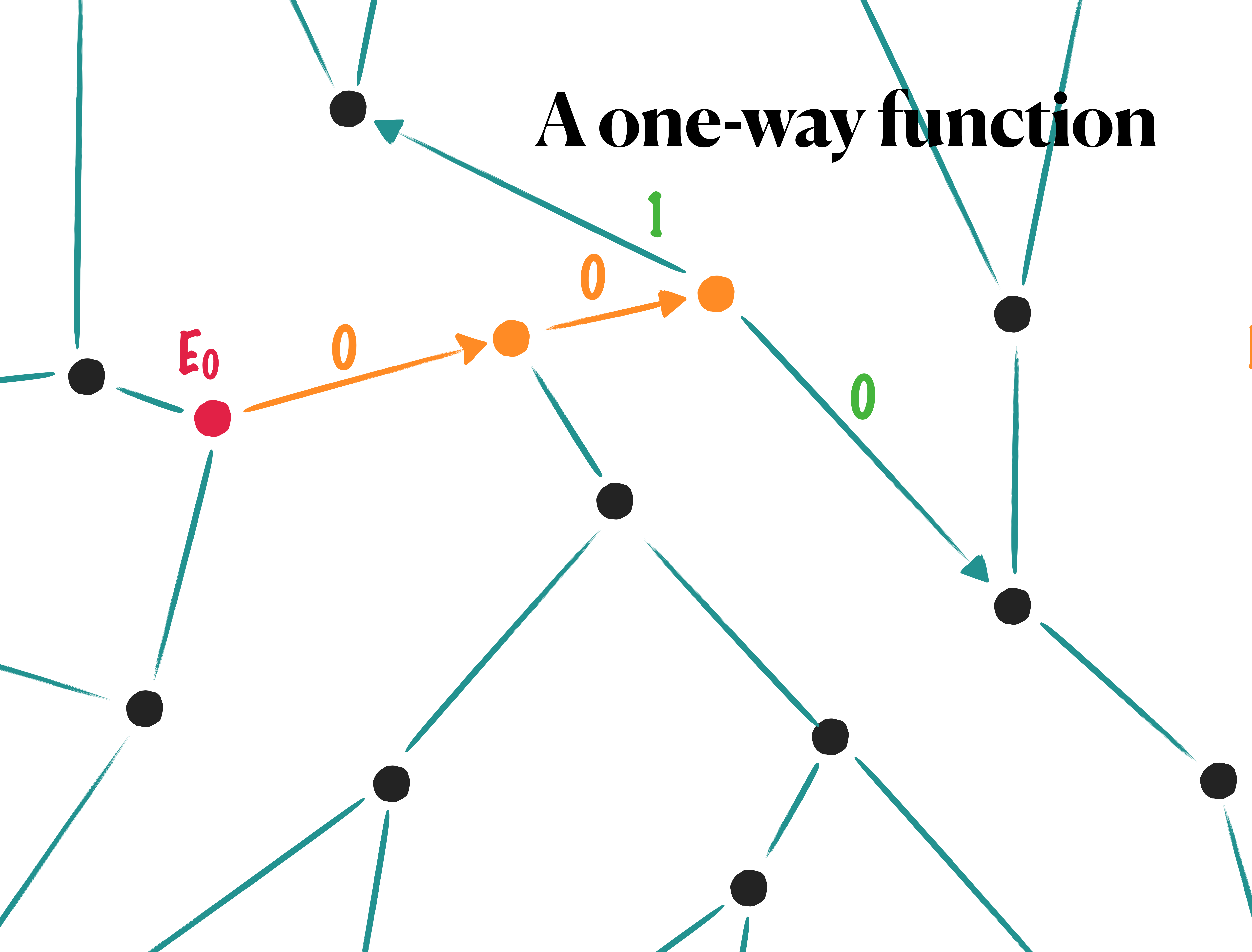


Fix a reference vertex E_0

Input: a message m in $\{0,1\}^*$

$m = 0001$

A one-way function

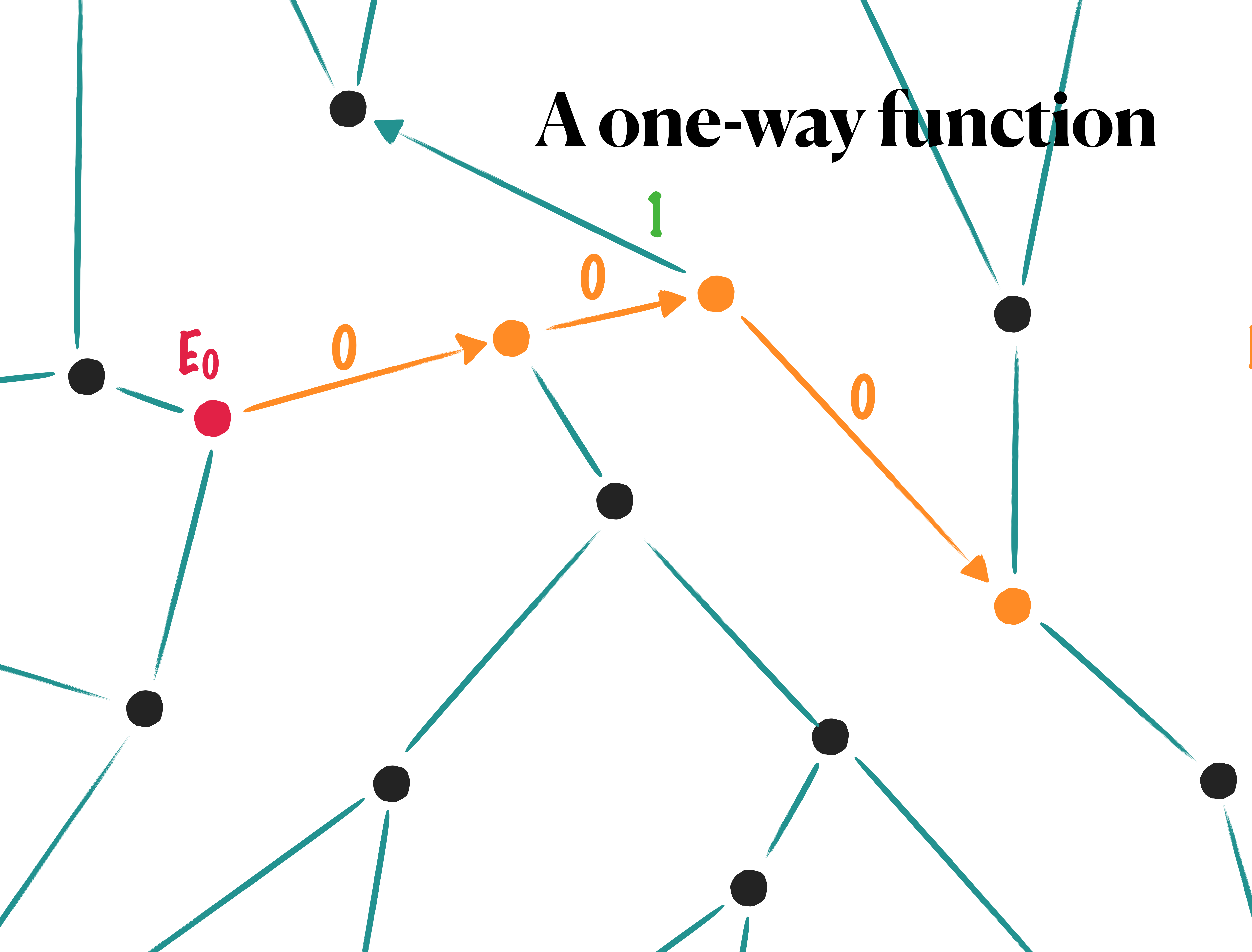


Fix a reference vertex E_0

Input: a message m in $\{0,1\}^*$

m = 0001

A one-way function

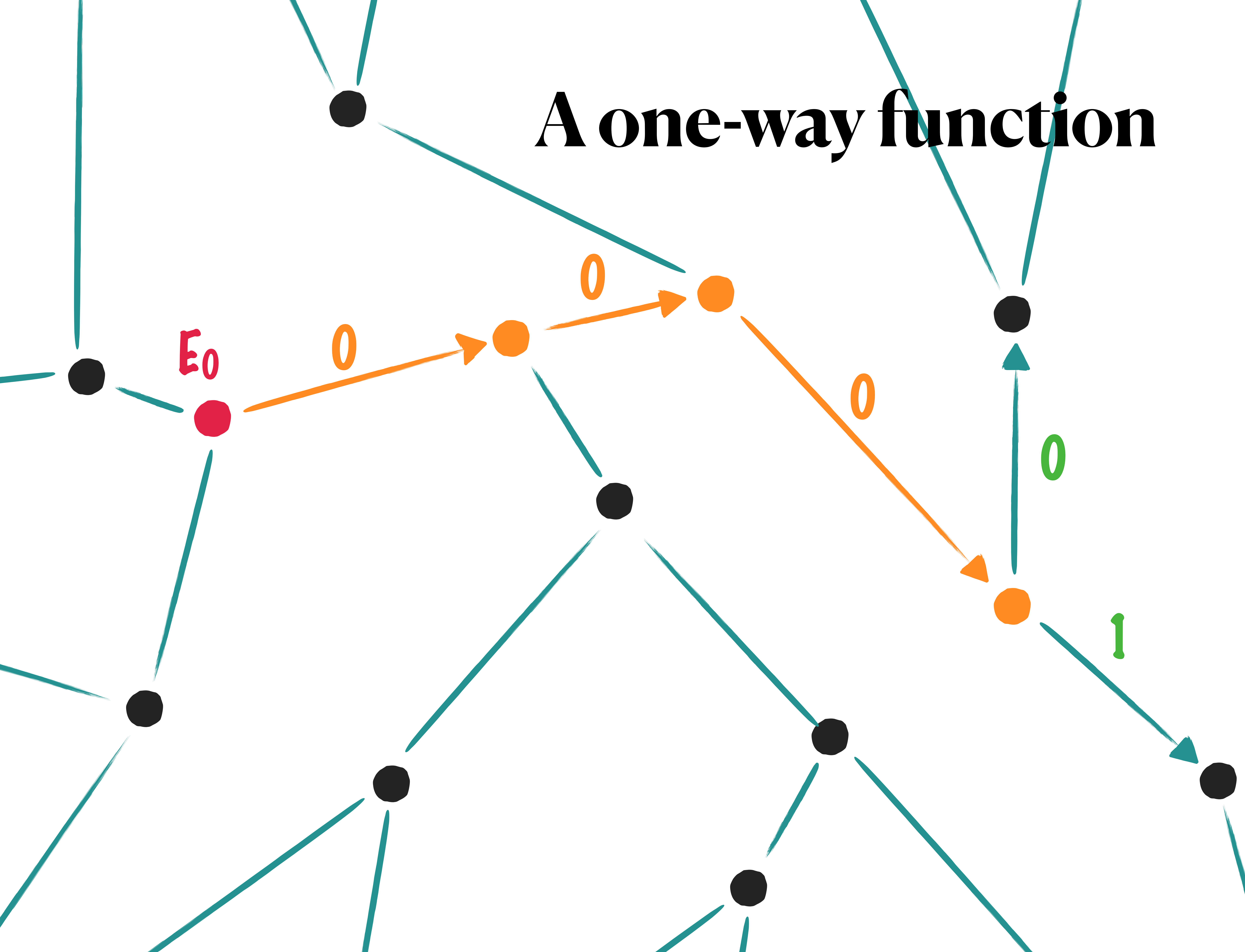


Fix a reference vertex E_0

Input: a message m in $\{0,1\}^*$

$m = 0001$

A one-way function

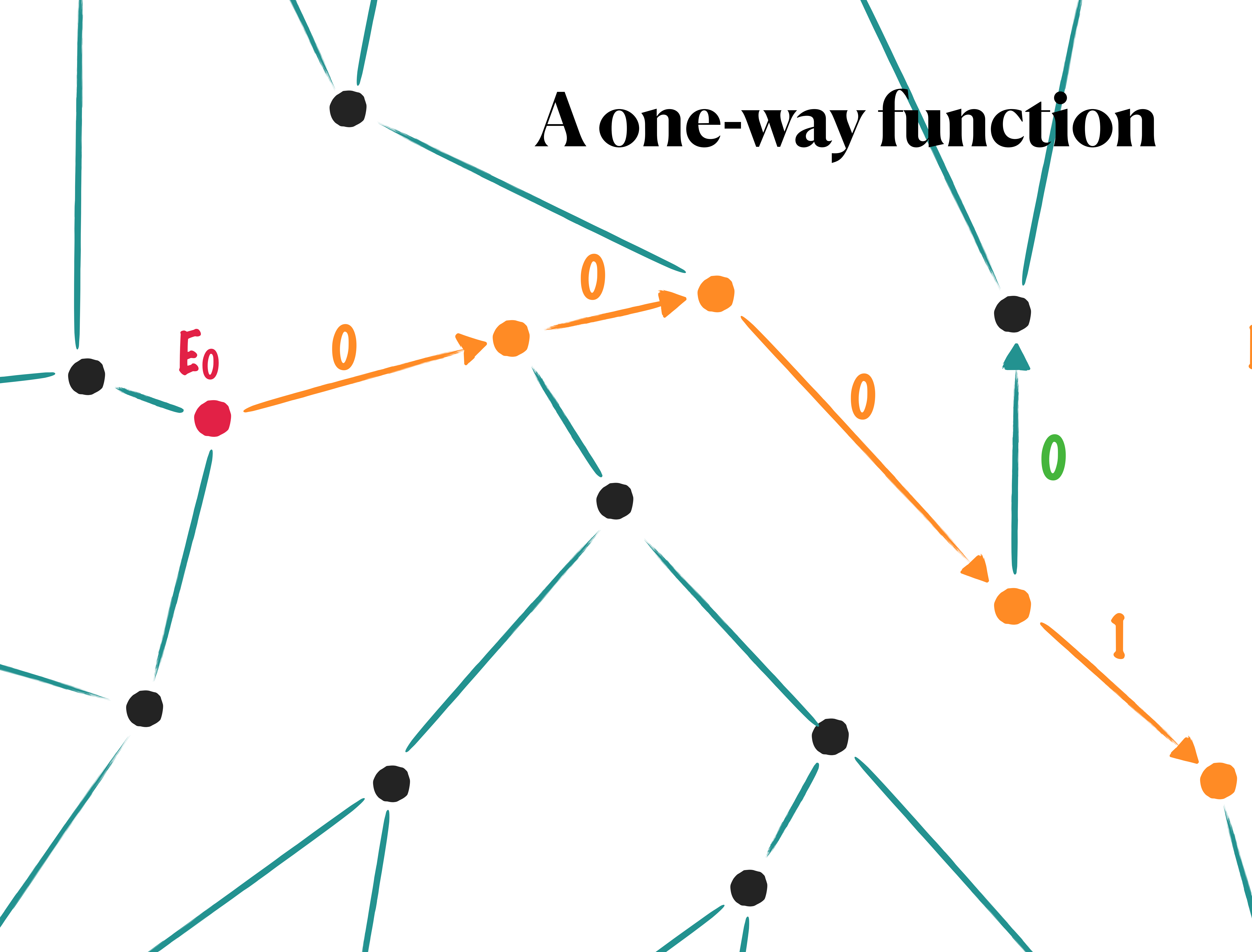


Fix a reference vertex E_0

Input: a message m in $\{0,1\}^*$

m = 0001

A one-way function

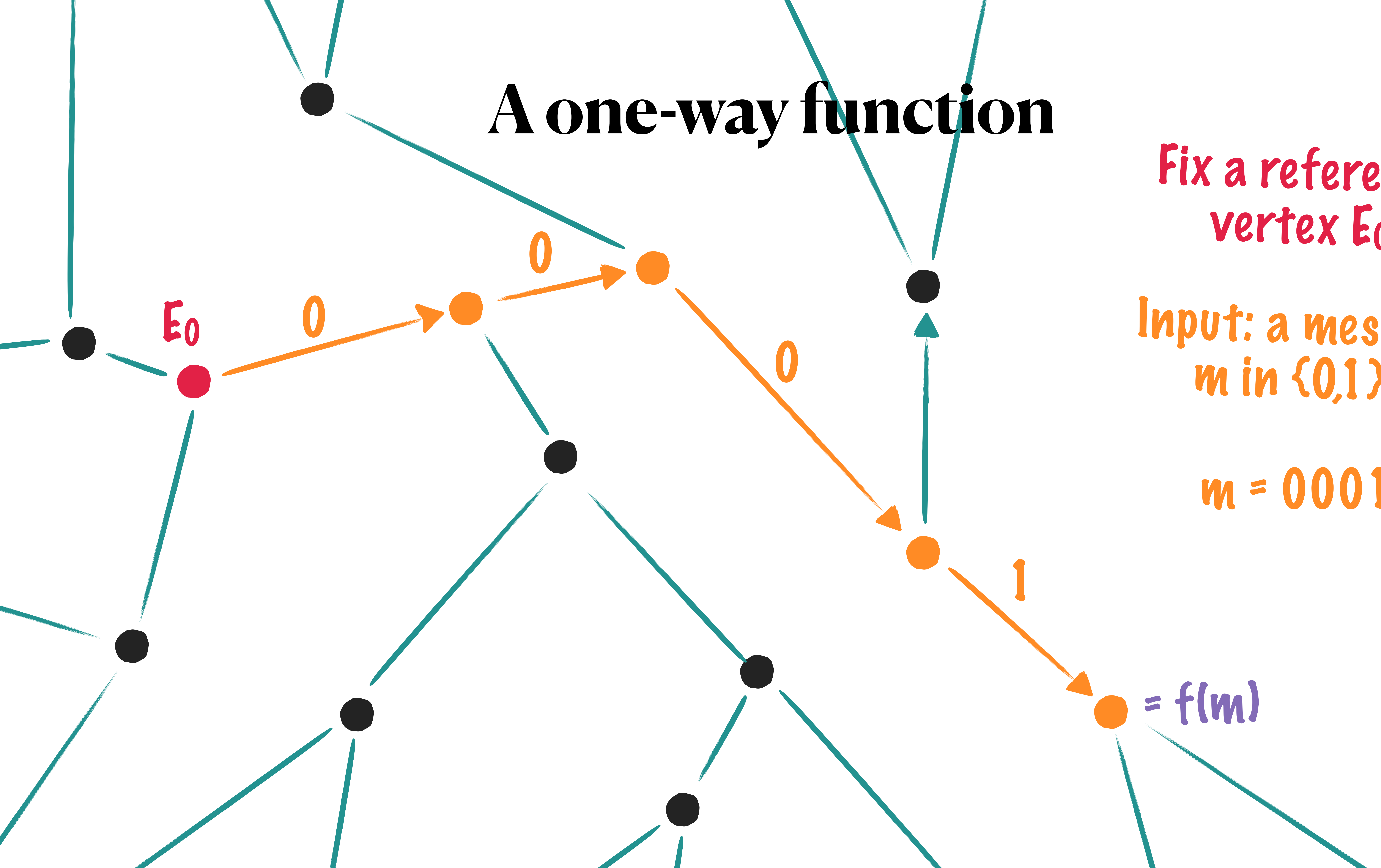


Fix a reference vertex E_0

Input: a message m in {0,1}*

m = 0001

A one-way function

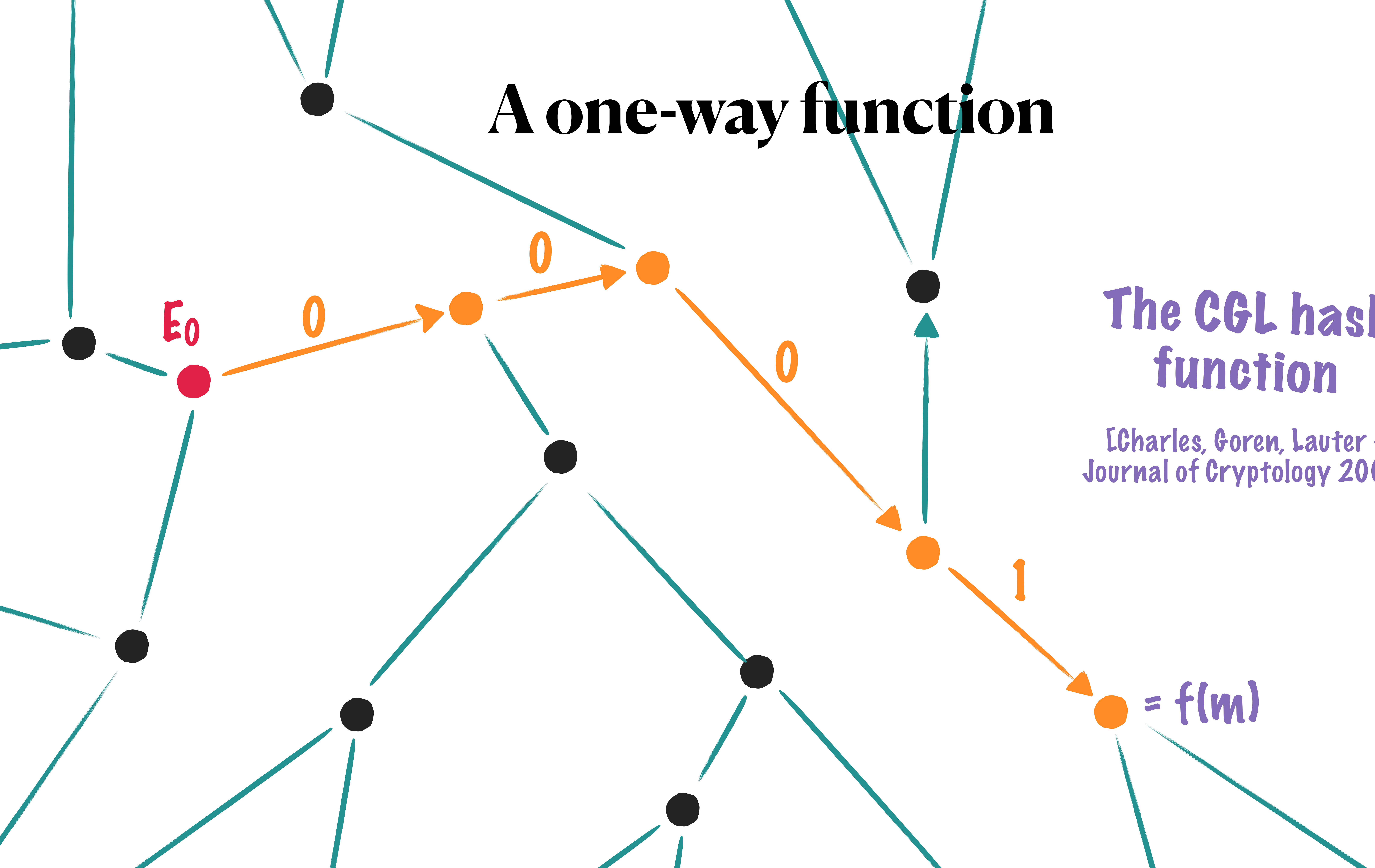


Fix a reference vertex E_0

Input: a message m in $\{0,1\}^*$

$m = 0001$

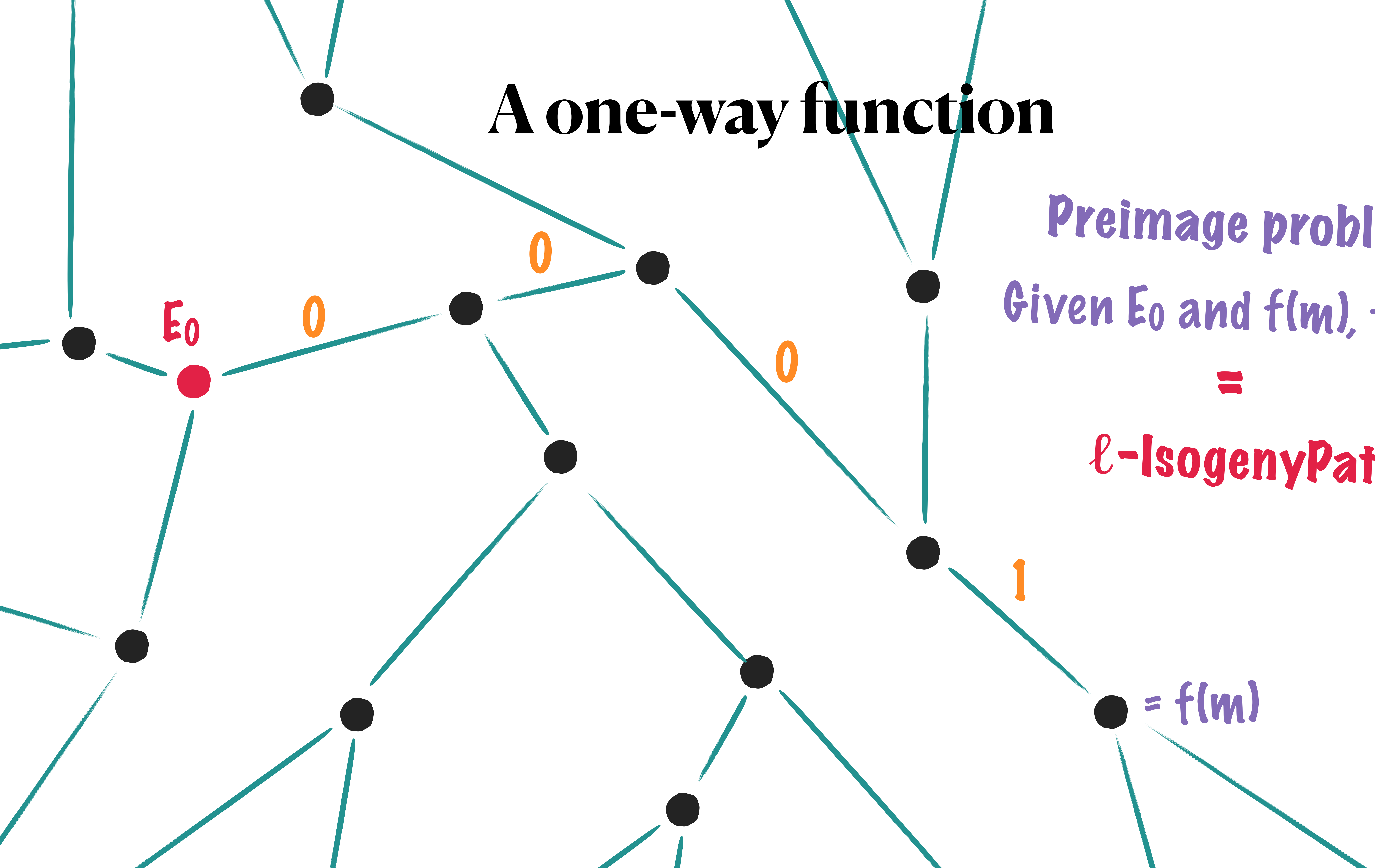
A one-way function



The CGL hash function

[Charles, Goren, Lauter - Journal of Cryptology 2009]

A one-way function



Preimage problem:

Given E_0 and $f(m)$, find m

=

ℓ-IsogenyPath

Isogeny-based cryptography

Hope: cryptosystems as secure as isogeny problem is hard

The isogeny problem = **Security of cryptosystems**

Hard even for quantum algorithms

Post-quantum cryptography

A one-way function 👍

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)
Vectorisation	$\leq$	CSIDH (key recovery)
~~**SSI-T**~~	~~$\leq$~~	~~SIDH (key recovery)~~

[Castryck, Decru]
[Maino, Martindale, Panny, Pope, W.]
[Robert]
Eurocrypt 2023

Endomorphisms

And the supersingular endomorphism ring problem

The CGL hash function

Is the CGL hash function collision-resistant?

Endomorphism ring

An **endomorphism** of E is an isogeny $\varphi : E \to E$ (or the zero map [0])

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

- $\varphi + \psi$ is pointwise addition: $(\varphi + \psi)(P) = \varphi(P) + \psi(P)$
- $\varphi\psi$ is the composition: $(\varphi\psi)(P) = \varphi(\psi(P))$

Multiplication by $m \in \mathbb{Z}$ is an endomorphism

$$[m] : E \to E : P \mapsto P + \ldots + P$$

It forms a subring $\mathbb{Z} \subset \mathrm{End}(E)$

Endomorphism ring

What is the structure of End(E)?

- It contains $\mathbb{Z} \subset \text{End}(E)$...
- (End(E), +) is a **lattice** of dimension 2 or 4

Endomorphism ring

What is the structure of End(E)?

- It contains $\mathbb{Z} \subset \text{End}(E)$...
- (End(E), +) is a **lattice** of dimension 2 or 4

A curve E is **supersingular** if (End(E), +) is a lattice of dimension 4

Then, there is a $\mathbb{Z}$-basis 1, α_2, α_3, α_4: as a lattice,

$$\text{End}(E) = \mathbb{Z} \oplus \mathbb{Z}\alpha_2 \oplus \mathbb{Z}\alpha_3 \oplus \mathbb{Z}\alpha_4$$

Endomorphism ring

What is the structure of End(E)?

- It contains $\mathbb{Z} \subset \text{End}(E)$...
- (End(E), +) is a **lattice** of dimension 2 or 4
- Has a **Euclidean norm**: $||\alpha||^2 = \deg(\alpha)$
- **Scalar product** $\langle \alpha, \beta \rangle = (\deg(\alpha + \beta) - \deg(\alpha - \beta))/4$, volume...

A curve E is **supersingular** if (End(E), +) is a lattice of dimension 4

Then, there is a $\mathbb{Z}$-basis $1, \alpha_2, \alpha_3, \alpha_4$: as a lattice,

$$\text{End}(E) = \mathbb{Z} \oplus \mathbb{Z}\alpha_2 \oplus \mathbb{Z}\alpha_3 \oplus \mathbb{Z}\alpha_4$$

The endomorphism ring problem

Given a supersingular E,
"compute End(E)"...

EndRing: Find four endomorphisms that form a basis of End(E)

Example

Example: $p \equiv 3 \pmod 4$, so $\mathbb{F}_{p^2} = \mathbb{F}_p(\alpha)$ where $\alpha^2 = -1$, and

Consider $E_0 : y^2 = x^3 + x$

Two non-trivial endomorphisms:

- $\pi : E_0 \to E_0 : (x, y) \longmapsto (x^p, y^p)$
- $\iota : E_0 \to E_0 : (x, y) \longmapsto (-x, \alpha y)$

$\pi^2 = [-p]$

$\iota^2 = [-1]$

and $\iota\pi = -\pi\iota$

$$\mathbf{End}(E_0) \overset{?}{=} \mathbb{Z} \oplus \mathbb{Z}\iota \oplus \mathbb{Z}\pi \oplus \mathbb{Z}\iota\pi$$

Example

Example: $p \equiv 3 \pmod 4$, so $\mathbb{F}_{p^2} = \mathbb{F}_p(\alpha)$ where $\alpha^2 = -1$, and

Consider $E_0 : y^2 = x^3 + x$

Two non-trivial endomorphisms:

- $\pi : E_0 \to E_0 : (x, y) \longmapsto (x^p, y^p)$
- $\iota : E_0 \to E_0 : (x, y) \longmapsto (-x, \alpha y)$

$\pi^2 = [-p]$

$\iota^2 = [-1]$

and $\iota\pi = -\pi\iota$

$$\mathbf{End}(\boldsymbol{E_0}) = \mathbb{Z} \oplus \mathbb{Z}\iota \oplus \mathbb{Z}\frac{\iota + \pi}{2} \oplus \mathbb{Z}\frac{1 + \iota\pi}{2}$$

EndRing

The endomorphism ring problem

Given a supersingular E,
"compute End(E)"...

EndRing: Find four endomorphisms that form a basis of $\mathrm{End}(E)$

MaxOrder: Compute the "abstract structure" of $\mathrm{End}(E)$

- $\mathrm{End}(E)$ is isomorphic to a ring of quaternions. Find which!

Quaternion algebra

The **quaternion algebra $\boldsymbol{B_{p,\infty}}$** is the ring (for $p \equiv 3 \pmod 4$)

$$B_{p,\infty} = \mathbb{Q} \oplus \mathbb{Q}\, i \oplus \mathbb{Q}\, j \oplus \mathbb{Q}\, k$$

where $i^2 = -1$, $j^2 = -p$, and $k = ij = -ji$

End(E) is (isomorphic to) a discrete subrings of $B_{p,\infty}$

- End(E) is a **maximal order** in $B_{p,\infty}$
- There are **many** maximal orders in $B_{p,\infty}$

The endomorphism ring problem

Given a supersingular E,
"compute End(E)"...

EndRing: Find four endomorphisms that form a basis of End(E)

MaxOrder: Compute the "abstract structure" of End(E)

- Find a subring of $\boldsymbol{B_{p,\infty}}$ isomorphic to End(E)

Example

Example: $p \equiv 3 \pmod 4$, so $\mathbb{F}_{p^2} = \mathbb{F}_p(\alpha)$ where $\alpha^2 = -1$, and

Consider $E_0 : y^2 = x^3 + x$

Two non-trivial endomorphisms:

- $\pi : E_0 \to E_0 : (x, y) \longmapsto (x^p, y^p)$
- $\iota : E_0 \to E_0 : (x, y) \longmapsto (-x, \alpha y)$

$\pi^2 = [-p]$

$\iota^2 = [-1]$

and $\iota\pi = -\pi\iota$

$$\mathbf{End}(E_0) = \mathbb{Z} \oplus \mathbb{Z}\iota \oplus \mathbb{Z}\frac{\iota + \pi}{2} \oplus \mathbb{Z}\frac{1 + \iota\pi}{2}$$

EndRing

$$\simeq \mathbb{Z} \oplus \mathbb{Z}i \oplus \mathbb{Z}\frac{i + j}{2} \oplus \mathbb{Z}\frac{1 + ij}{2} \subset B_{p,\infty}$$

MaxOrder

The CGL hash function

The *OneEnd* problem

Given E (supersingular) find ***one*** endomorphism $\alpha \in \mathrm{End}(E) \setminus \mathbb{Z}$

The endomorphism ring problem

Given a supersingular E,
"compute End(E)"...

EndRing: Find four endomorphisms that form a basis of End(E)

MaxOrder: Compute the "abstract structure" of End(E)

- Find a subring of $\boldsymbol{B_{p,\infty}}$ isomorphic to End(E)

OneEnd: Find a single non-scalar endomorphism in $\alpha \in \mathrm{End}(E) \setminus \mathbb{Z}$

Foundations

Relations between problems

Which is hardest? Easiest?

EndRing

ℓ-IsogenyPath

OneEnd

MaxOrder

Isogeny

Relating OneEnd to EndRing

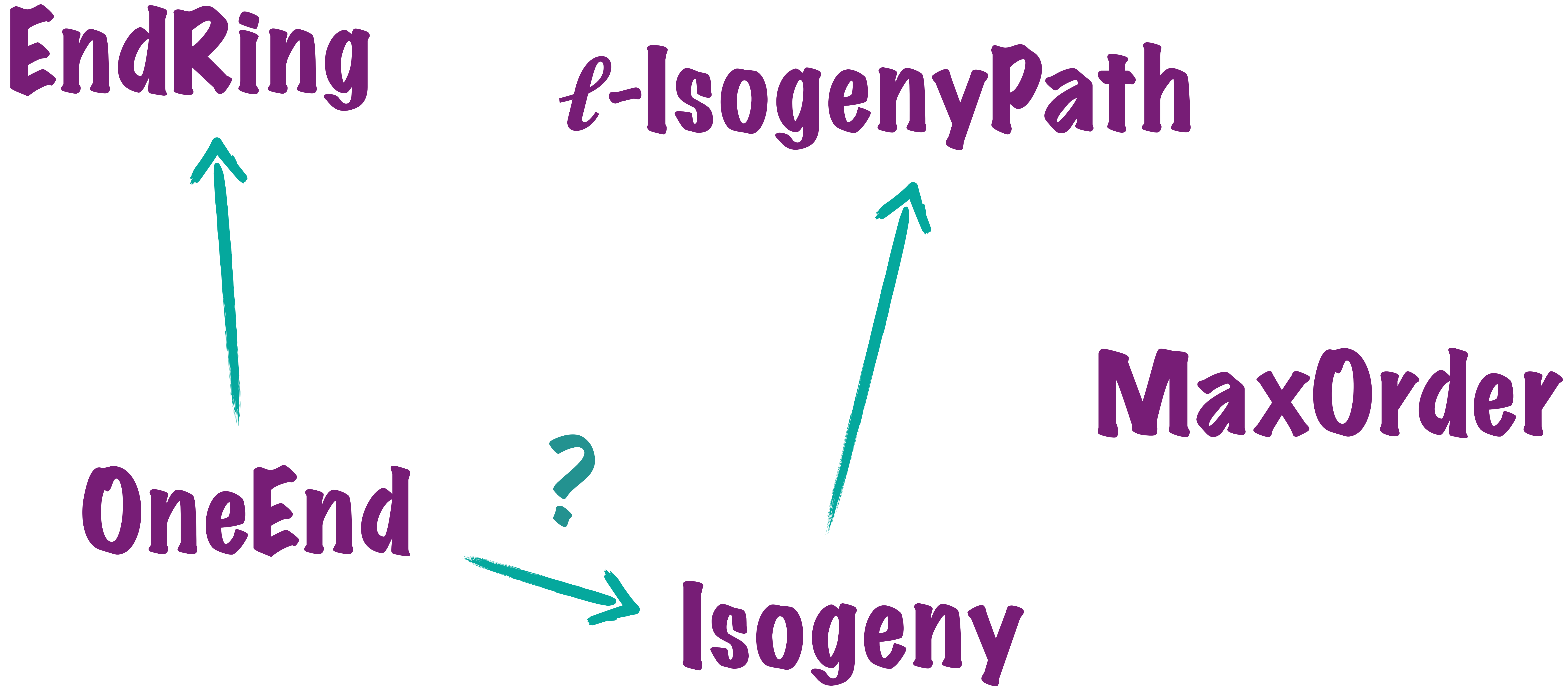
Suppose we can solve **EndRing**. Can we solve **OneEnd**?

Given E, we solve **OneEnd** for E as follows:

1. Solve **EndRing** for E, finding a basis $1, \alpha_2, \alpha_3, \alpha_4$ of $\mathrm{End}(E)$
2. Return α_2

We have that $\alpha_2 \in \mathrm{End}(E) \setminus \mathbb{Z}$ because α_2 is not in $\mathrm{span}(1) = \mathbb{Z}$

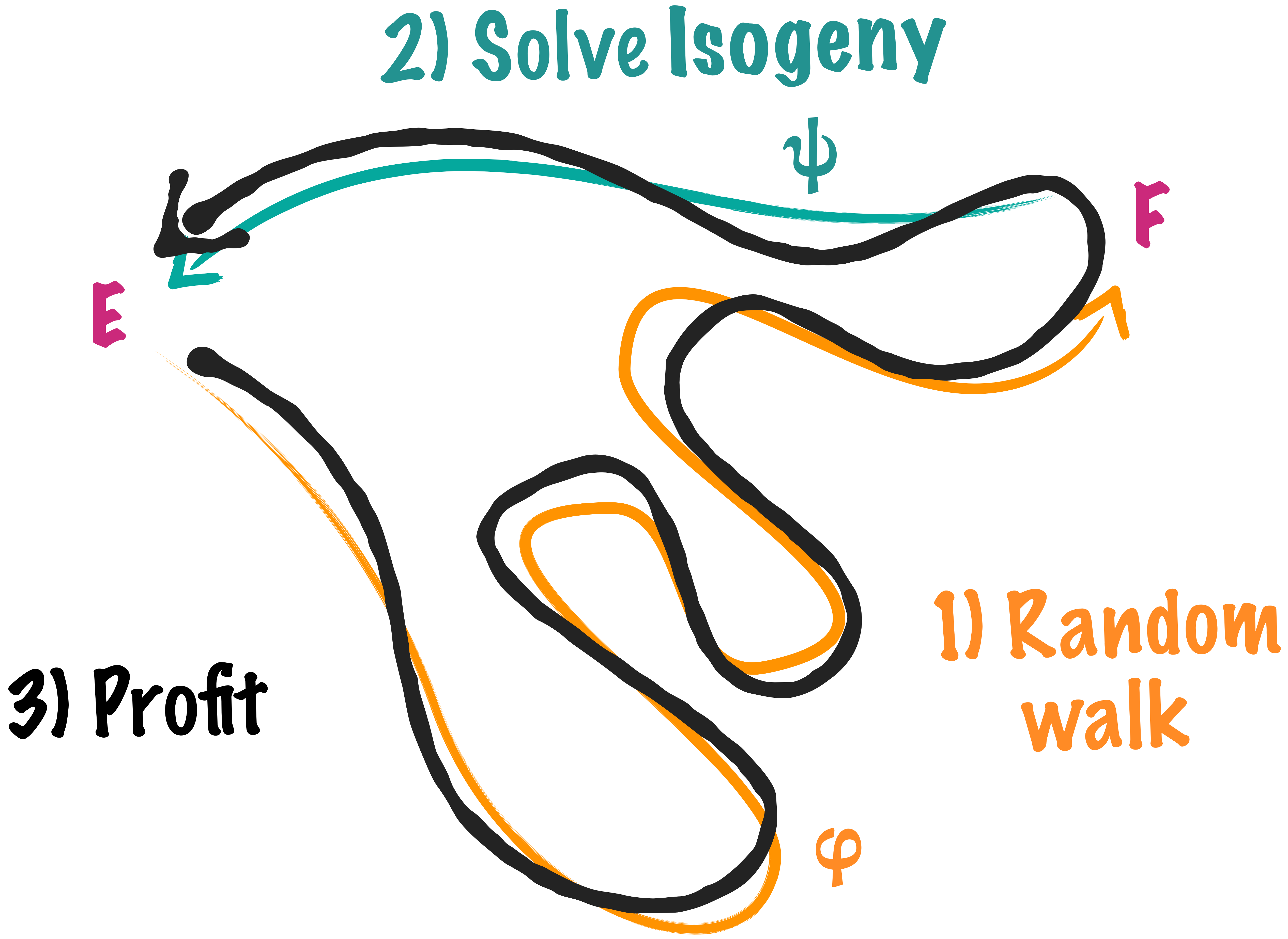
Which is hardest? Easiest?



Relating OneEnd to Isogeny

Suppose we can solve **Isogeny**.
Can we solve **OneEnd**?

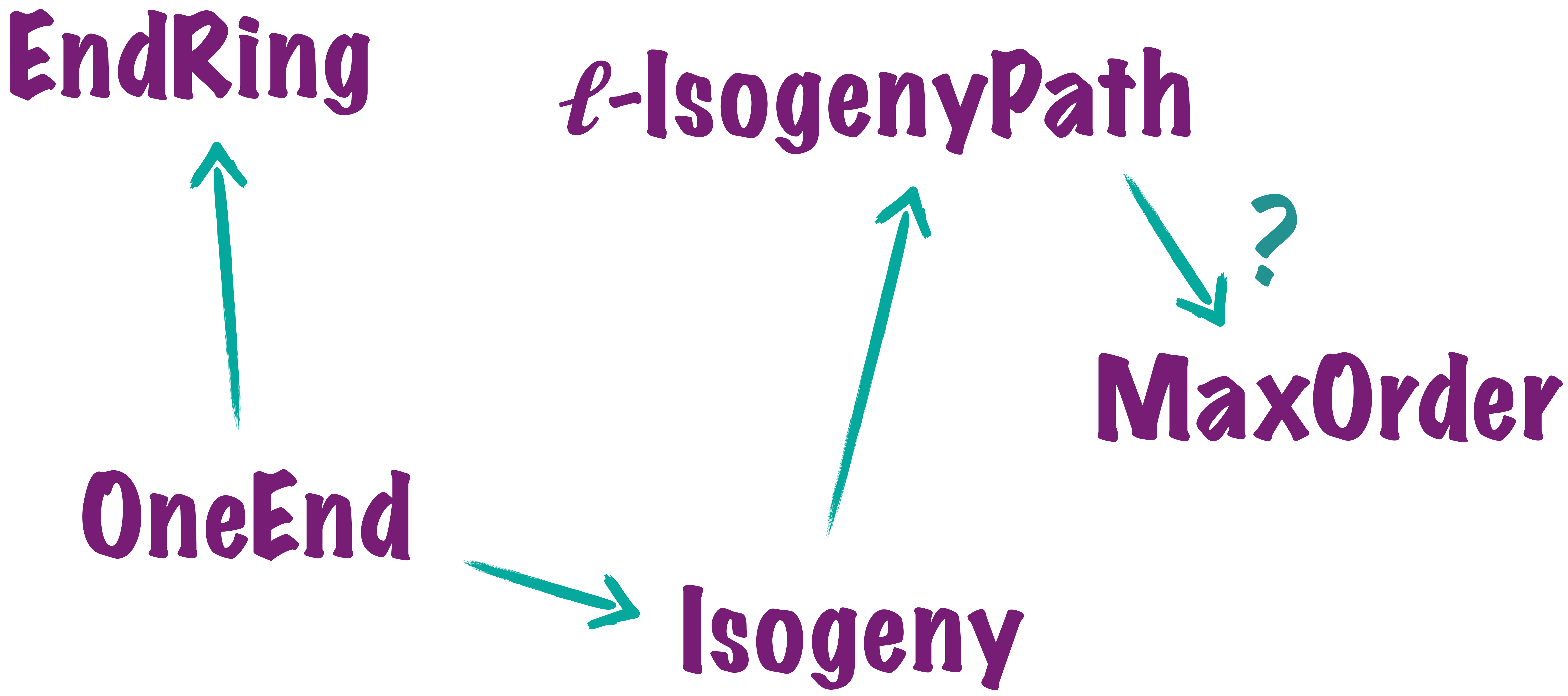
How to find endomorphisms of E:



Does $\psi \circ \varphi \in \mathbb{Z}$?

- Not if φ is long enough, and has cyclic kernel

Which is hardest? Easiest?



Isogeny World | Quaternion World

Deuring correspondence

Isogeny World	Quaternion World
Supersingular curves $\boldsymbol{E}$ over $\mathbb{F}_{p^2}$ (up to isomorphism)	Maximal orders $\mathcal{O}$ in $B_{p,\infty}$ $\mathcal{O} \simeq \mathbf{End}(\boldsymbol{E})$ (up to isomorphism)
Isogenies $\varphi : \boldsymbol{E} \to \boldsymbol{E'}$	$(\mathcal{O},\mathcal{O}')$-ideals $\boldsymbol{I}$, $\mathcal{O} \simeq \mathbf{End}(\boldsymbol{E})$ and $\mathcal{O}' \simeq \mathbf{End}(\boldsymbol{E'})$

HARD | **HARD? EASY?**

MaxOrder

ℓ-Isogeny Path:

Given E and E',
find φ : E $\longrightarrow$ E' of degree ℓ^n

ℓ-Quaternion Path:

Given $\mathcal{O}$ and $\mathcal{O}'$, find an
$(\mathcal{O},\mathcal{O}')$-ideal I of norm ℓ^n

Solving the Quaternion Path Problem

Theorem: There exists an algorithm that **solves the ℓ-quaternion path problem** in expected polynomial time (assuming GRH).

Full proof under GRH: **[W. – FOCS 2021]** *The supersingular isogeny path and endomorphism ring problems are equivalent*.

Much faster, but heuristic algorithm: **[Kohel, Lauter, Petit, Tignol – ANTS 2014]** *On the quaternion ℓ -isogeny path problem*.

The "KLPT" algorithm

Isogeny World

Quaternion World

Deuring correspondence

Isogeny World	Quaternion World
Supersingular curves $\boldsymbol{E}$ over $\mathbb{F}_{p^2}$ (up to isomorphism)	Maximal orders $\mathcal{O}$ in $B_{p,\infty}$; $\mathcal{O} \simeq \mathbf{End}(\boldsymbol{E})$ (up to isomorphism)
Isogenies $\varphi : \boldsymbol{E} \to \boldsymbol{E'}$	$(\mathcal{O},\mathcal{O}')$-ideals $\boldsymbol{I}$, $\mathcal{O} \simeq \mathbf{End}(\boldsymbol{E})$ and $\mathcal{O}' \simeq \mathbf{End}(\boldsymbol{E'})$

HARD → **MaxOrder** → **EASY**

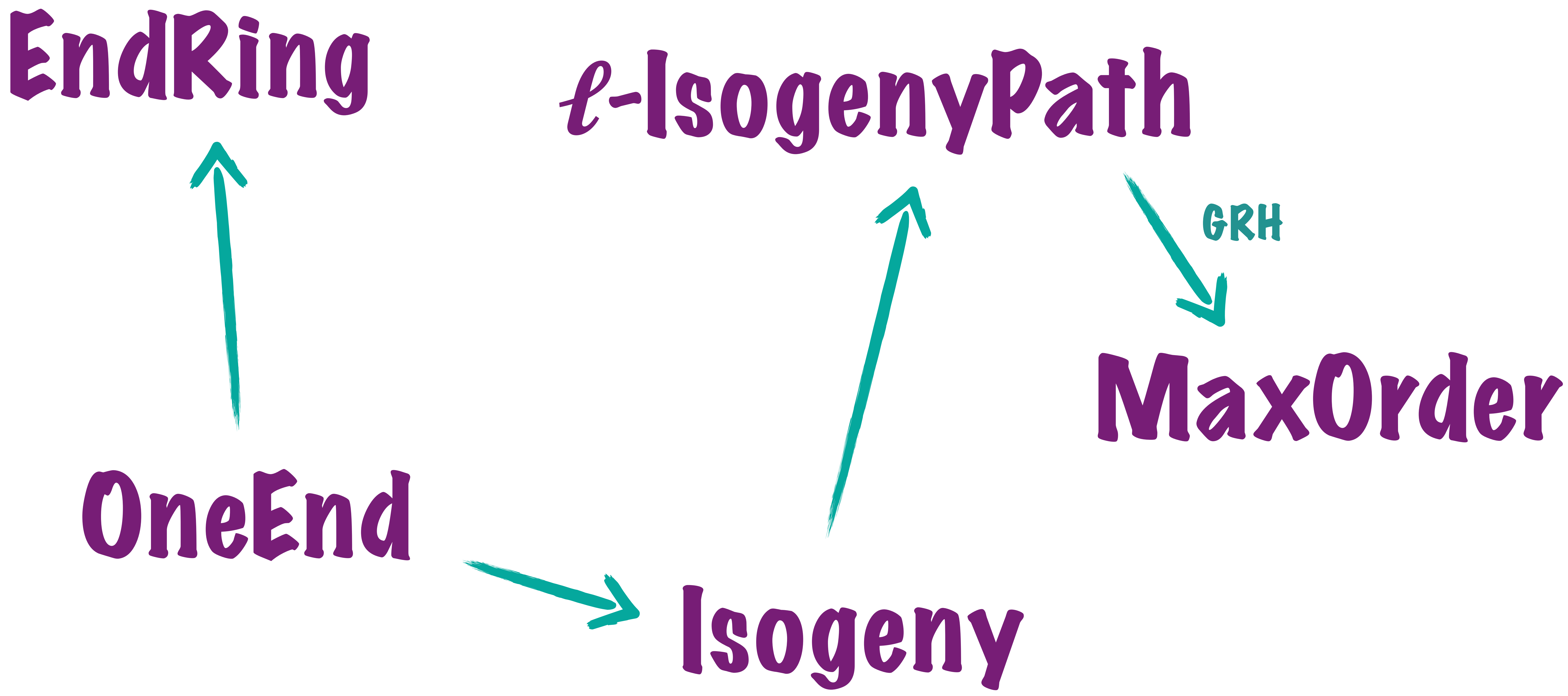
ℓ-Isogeny Path:

Given E and E',
find φ : E ⟶ E' of degree ℓ^n

ℓ-Quaternion Path:

Given $\mathcal{O}$ and $\mathcal{O}'$, find an
$(\mathcal{O},\mathcal{O}')$-ideal I of norm ℓ^n

Which is hardest? Easiest?



Which is hardest? Easiest?

$$\text{EndRing} \overset{\text{GRH}}{\longleftrightarrow} \text{MaxOrder} \overset{\text{GRH}}{\longleftrightarrow} \ell\text{-IsogenyPath}$$

Proof assuming GRH:

[W. – FOCS 2021] *The supersingular isogeny path and endomorphism ring problems are equivalent.*

Earlier heuristic reductions:

[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.*

Which is hardest? Easiest?

$$\begin{tikzcd} \text{EndRing} \arrow[r, leftrightarrow, "\text{GRH}"] & \text{MaxOrder} \arrow[r, leftrightarrow, "\text{GRH}"] & \ell\text{-IsogenyPath} \\ \text{OneEnd} \arrow[u] \arrow[r] & \begin{array}{l}\text{CGL collision-resistance}\\ \text{SQIsign soundness}\end{array} \arrow[ul] \end{tikzcd}$$

Which is hardest? Easiest?

EndRing $\overset{\text{GRH}}{\longleftrightarrow}$ MaxOrder $\overset{\text{GRH}}{\longleftrightarrow}$ ℓ-IsogenyPath

$\uparrow\downarrow$

[Page, W. – Eurocrypt 2024] *The supersingular Endomorphism Ring and One Endomorphism problems are equivalent.*

OneEnd $\longleftrightarrow$ CGL collision-resistance
SQIsign soundness

Which is hardest? Easiest?

$$
\begin{tikzcd}
\textsf{EndRing} \arrow[r, leftrightarrow, "\text{GRH}"] \arrow[d, leftrightarrow] \arrow[dr, phantom, "\text{Unconditional!}"] & \textsf{MaxOrder} \arrow[r, leftrightarrow, "\text{GRH}"] & \ell\text{-}\textsf{IsogenyPath} \\
\textsf{OneEnd} \arrow[r] & \textsf{Isogeny} \arrow[ur] &
\end{tikzcd}
$$

Which is hardest? Easiest?

$$\begin{tikzcd}
\text{EndRing} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \text{MaxOrder} \arrow[d, leftrightarrow] & \ell\text{-IsogenyPath} \arrow[l, "\text{GRH}"'] \\
\text{OneEnd} \arrow[r, leftrightarrow] & \text{Isogeny} \arrow[ur]
\end{tikzcd}$$

[Herlédan Le Merdy, W. — to appear] *Unconditional foundations for supersingular isogeny-based cryptography*

Average hardness

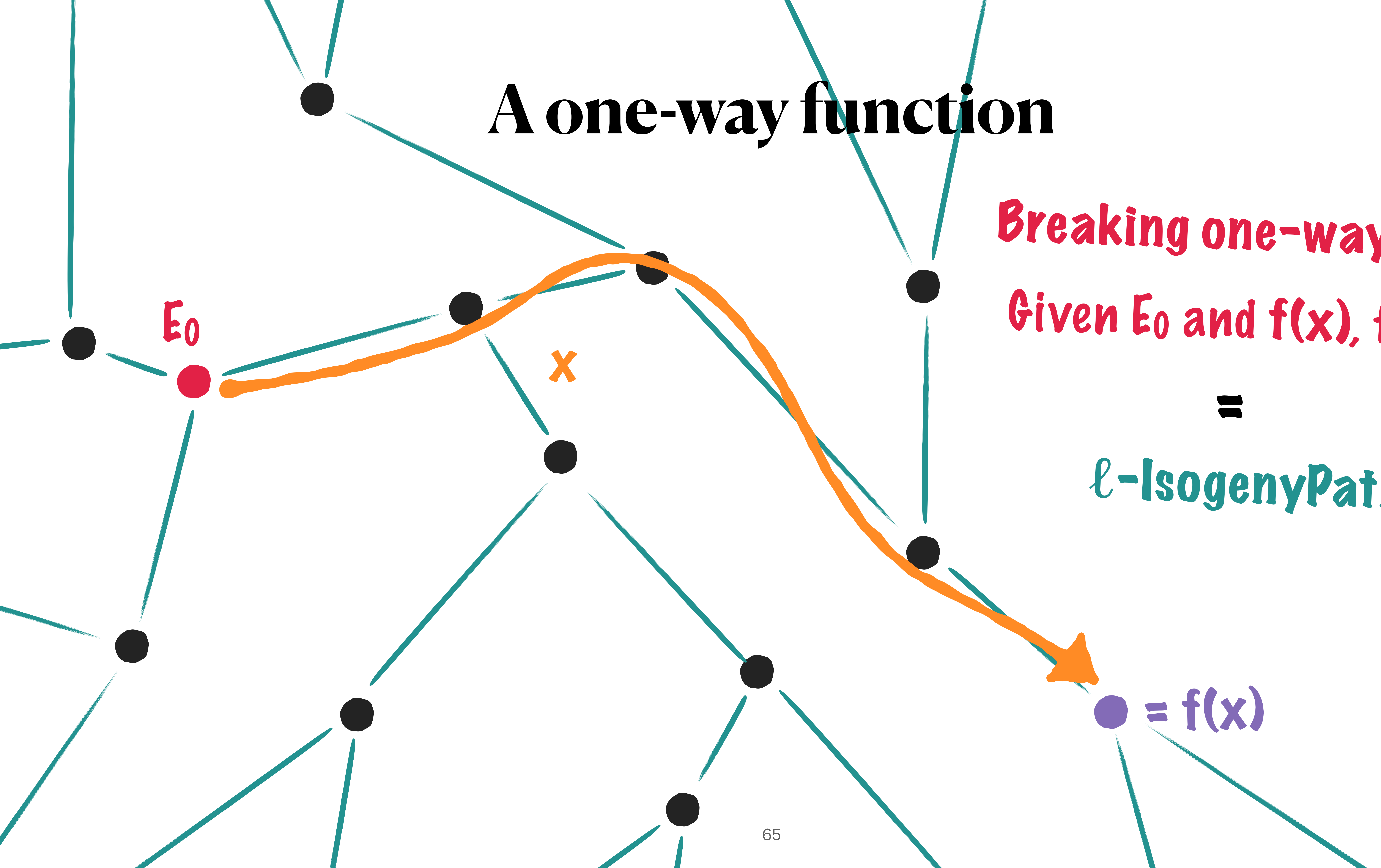
and worst-case to
average-case reductions

A one-way function

The CGL hash function

[Charles, Goren, Lauter – Journal of Cryptology 2009]

A one-way function



Breaking one-wayness:

Given E_0 and f(x), find x

=

ℓ-IsogenyPath

A one-way function

- **One-way function:** a function $f : X \to Y$ which is
  - ➡ **Easy to evaluate:** given $x \in X$, it is easy to compute $f(x)$
  - ➡ ~~**Hard to invert:** given $y \in Y$, it is hard to find some $x \in X$ such that $f(x) = y$~~
  - ➡ **Hard to invert:** let $x \in X$ uniformly random, and $y = f(x)$. There is no *efficient* algorithm A such that $A(y)$ outputs a preimage of y with *good probability*

For security, we care about average hardness

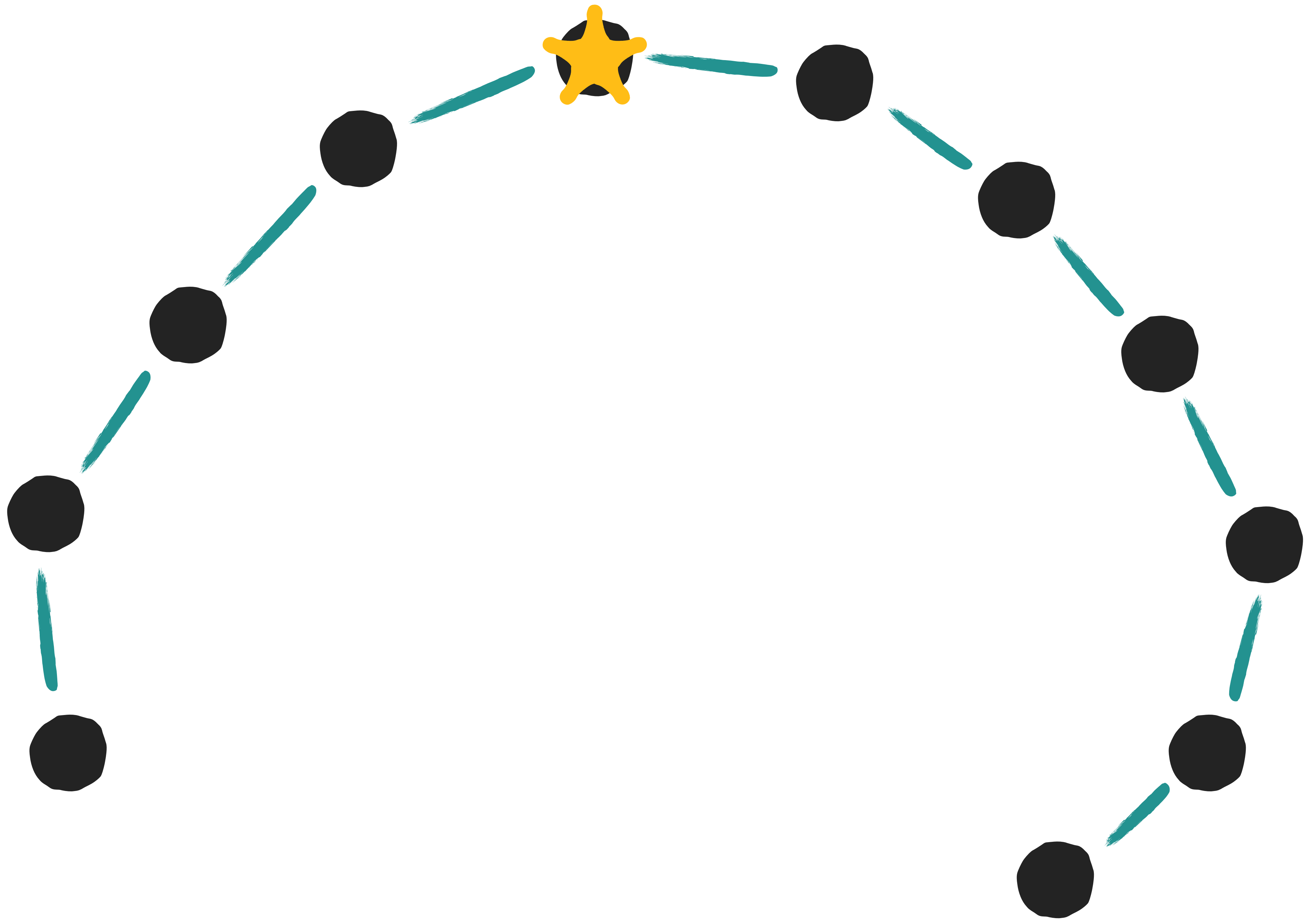
A problem should be hard on average for random inputs

A one-way function

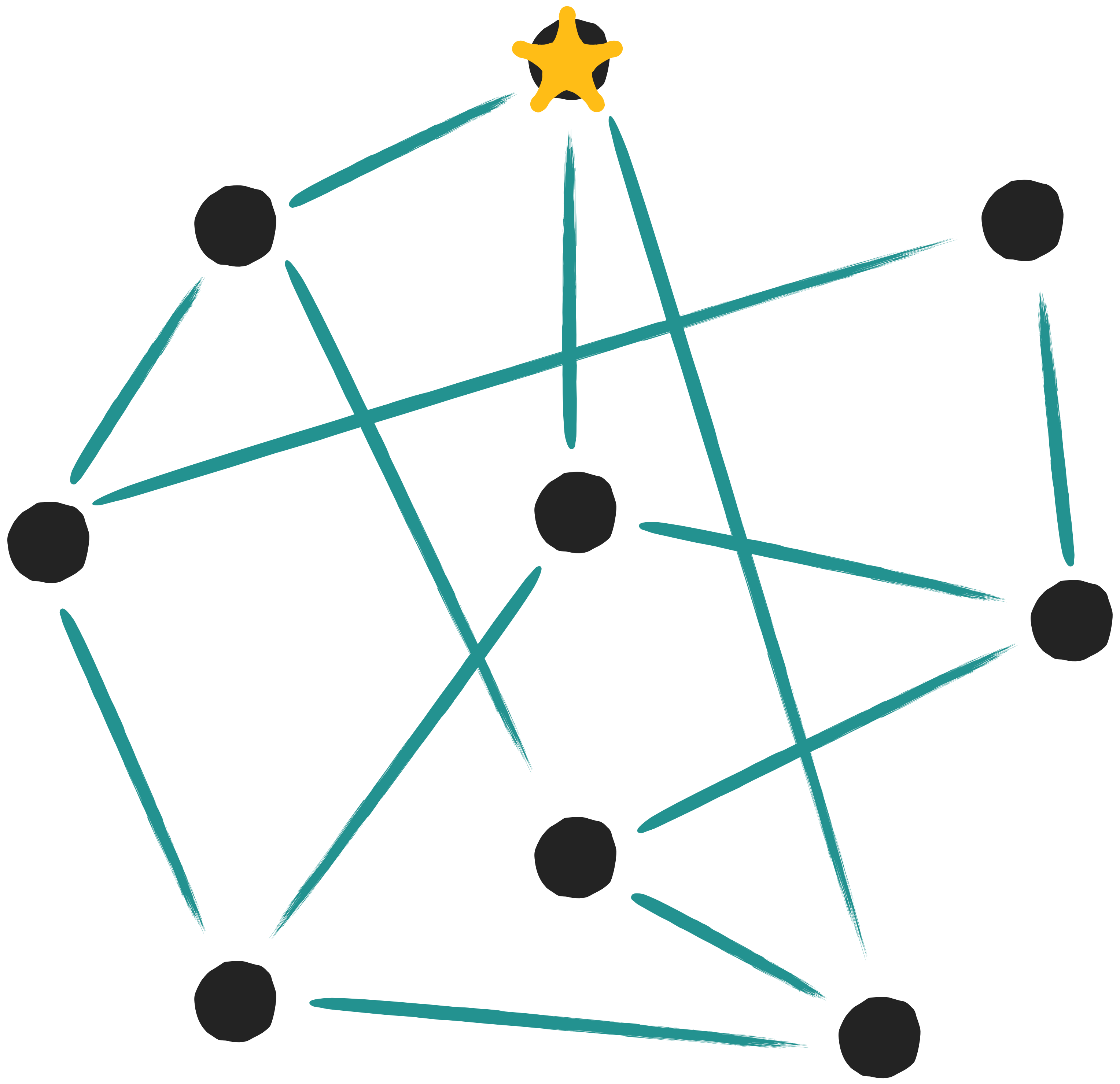
Rapid mixing

- Some graphs have better "mixing" properties than others...

Stays close to starting point for a long time...



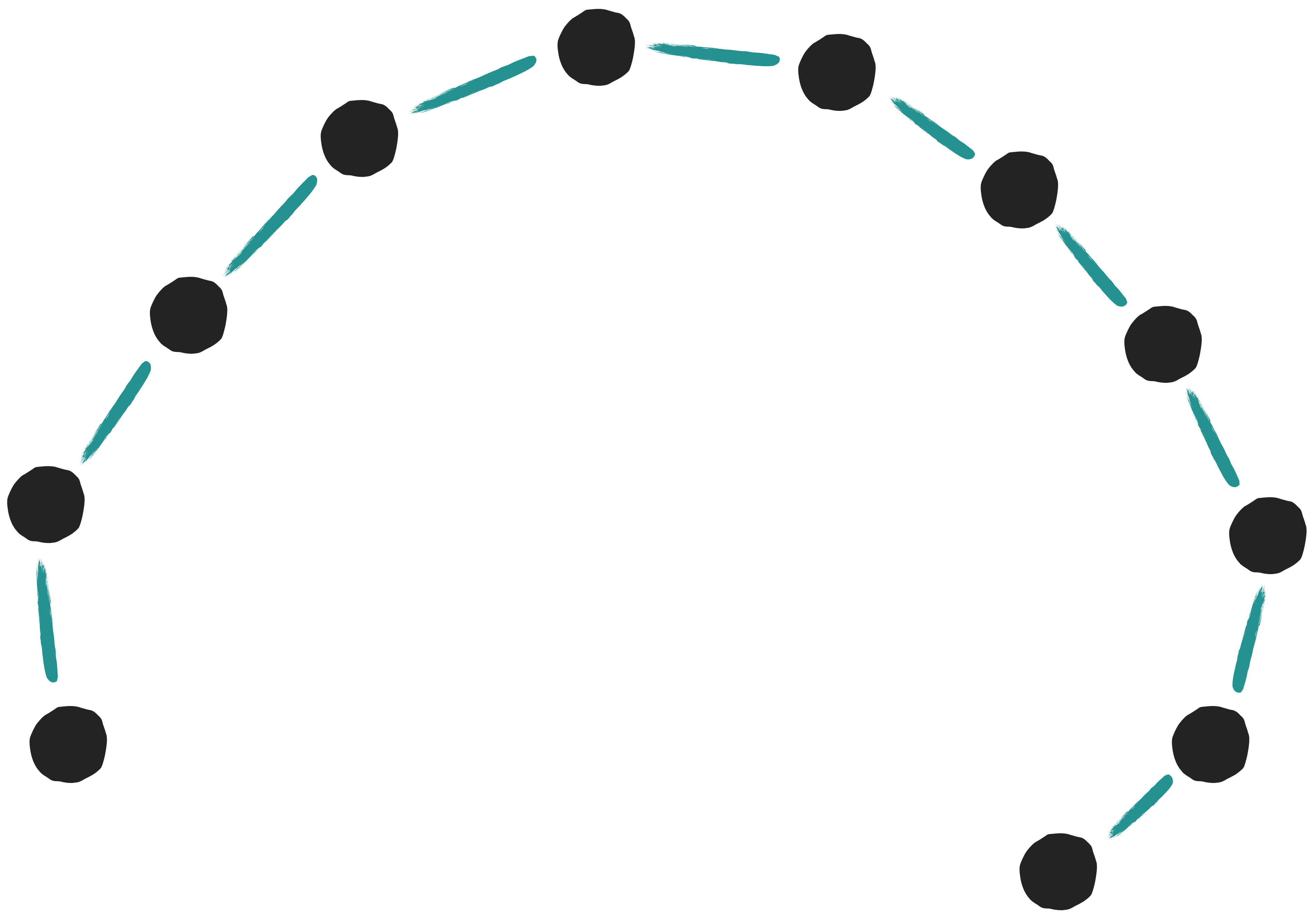
Rapidly goes anywhere



Rapid mixing

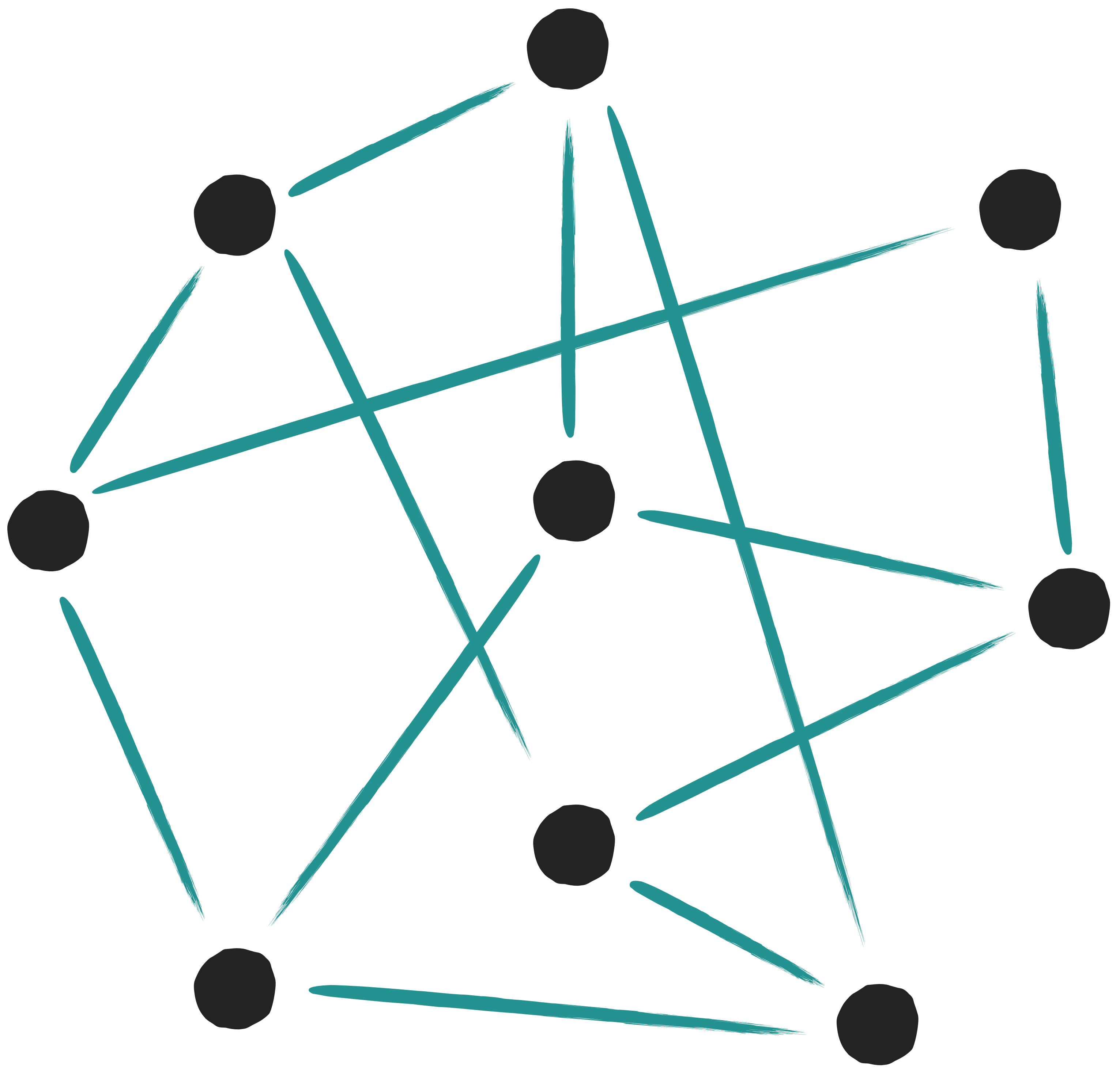
"slow mixing"

Stays close to starting point for a long time...



"rapid mixing"

Rapidly goes anywhere



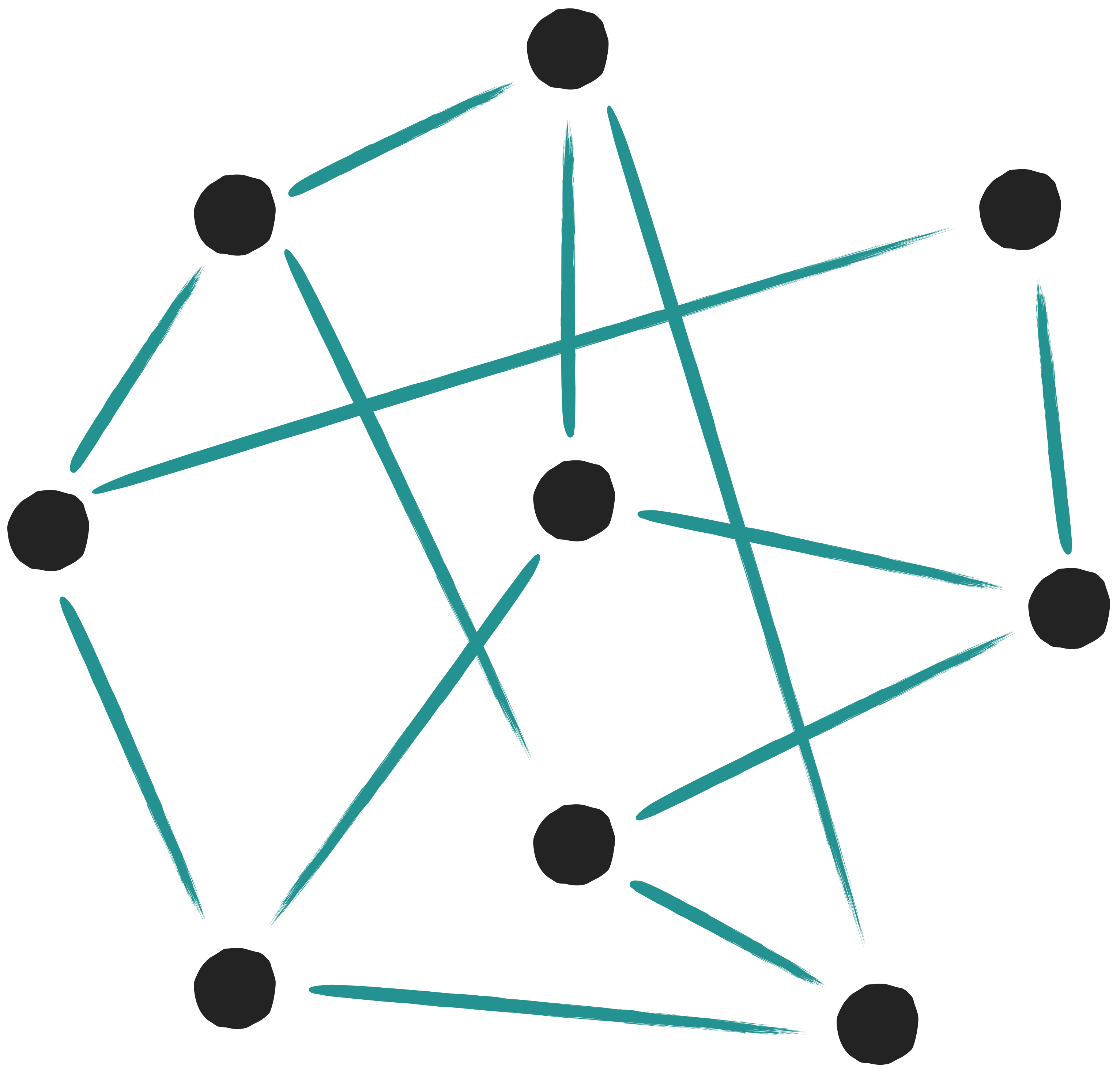
Rapid mixing

The best mixers are
Ramanujan graphs

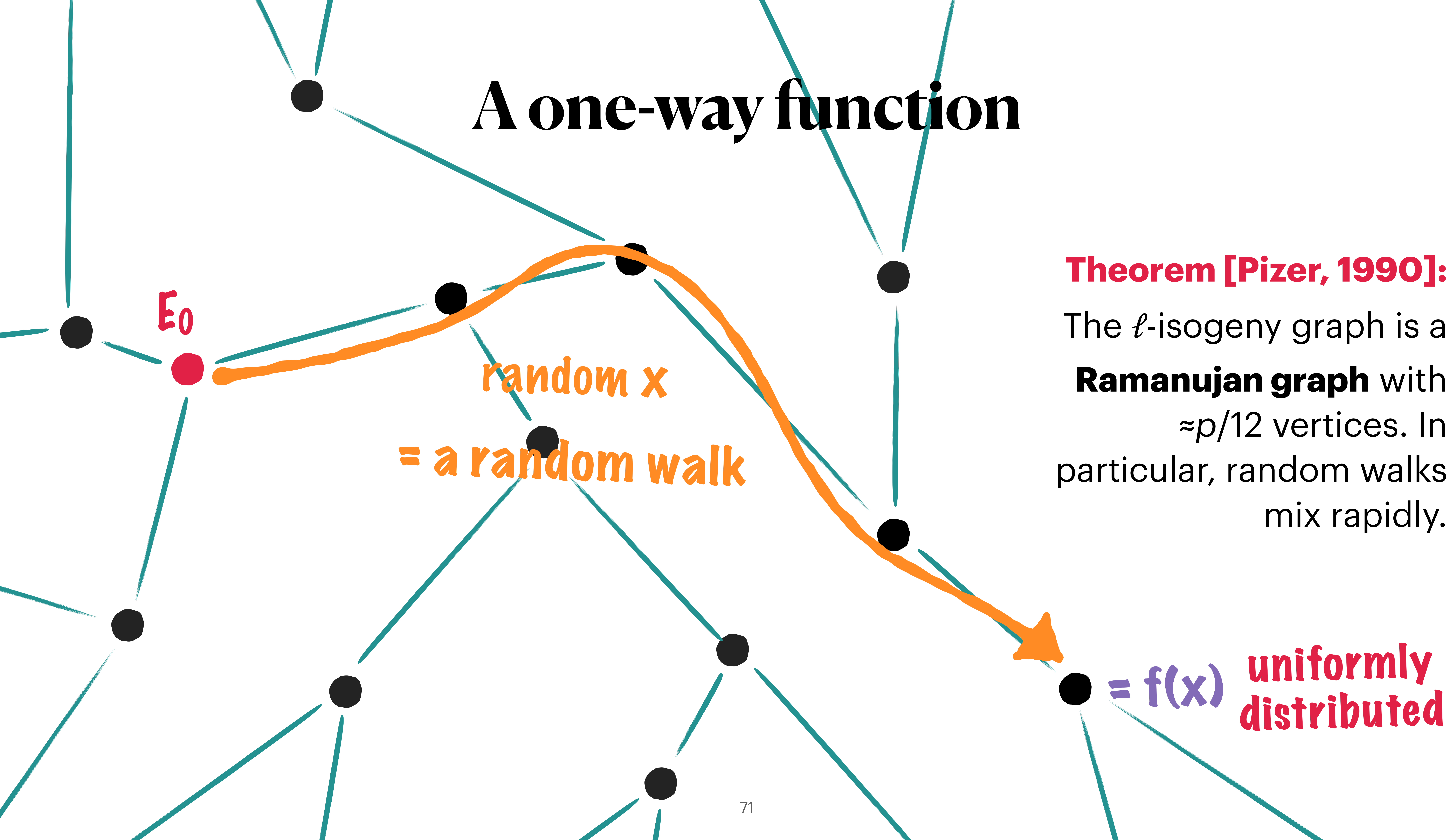
Theorem: In a **Ramanujan graph** with n vertices, a random walk of length ≈log(n) reaches a distribution **indistinguishable from uniform.**

"rapid mixing"

Rapidly goes anywhere



A one-way function

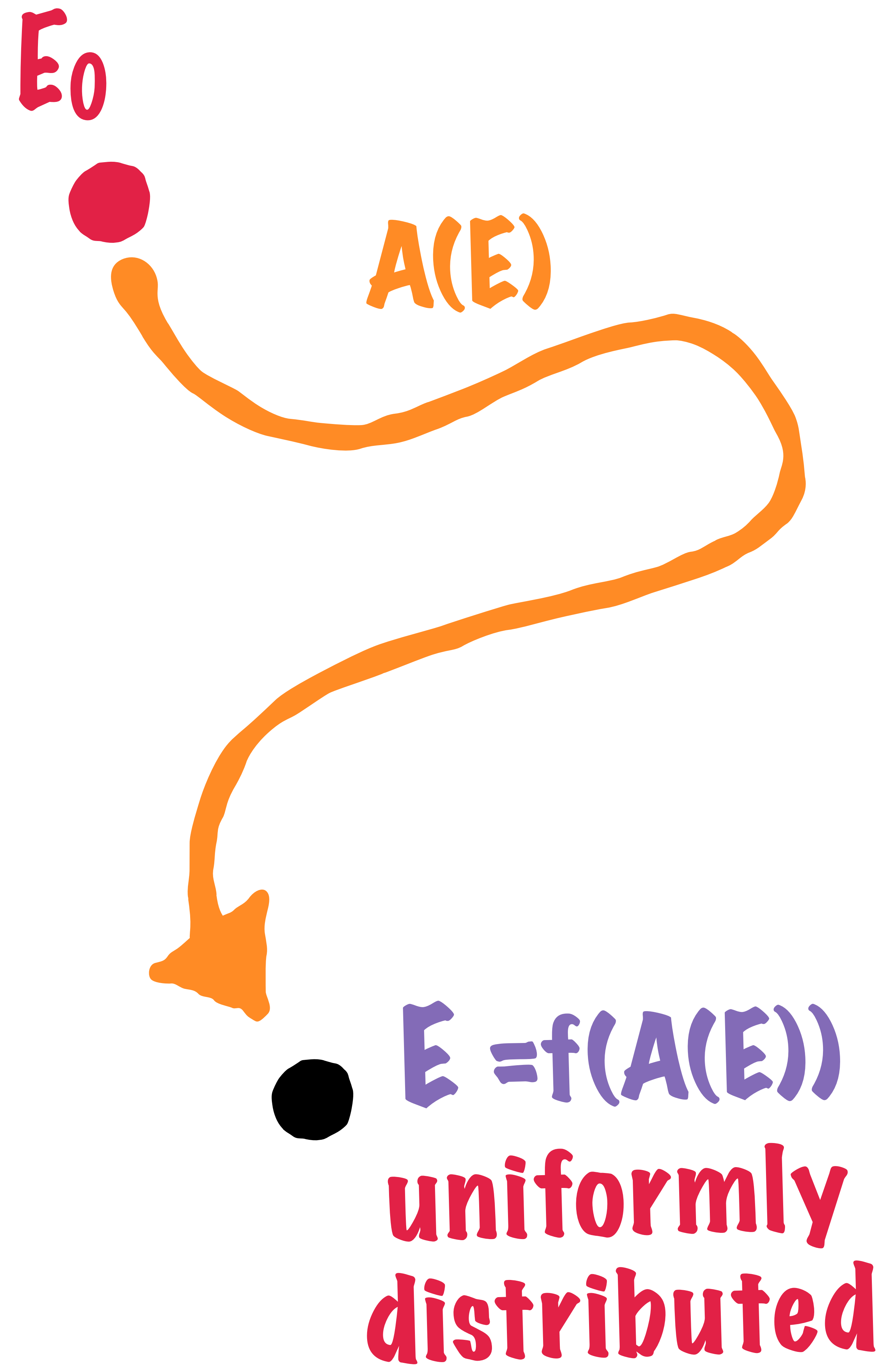


Theorem [Pizer, 1990]:

The ℓ-isogeny graph is a **Ramanujan graph** with $\approx p/12$ vertices. In particular, random walks mix rapidly.

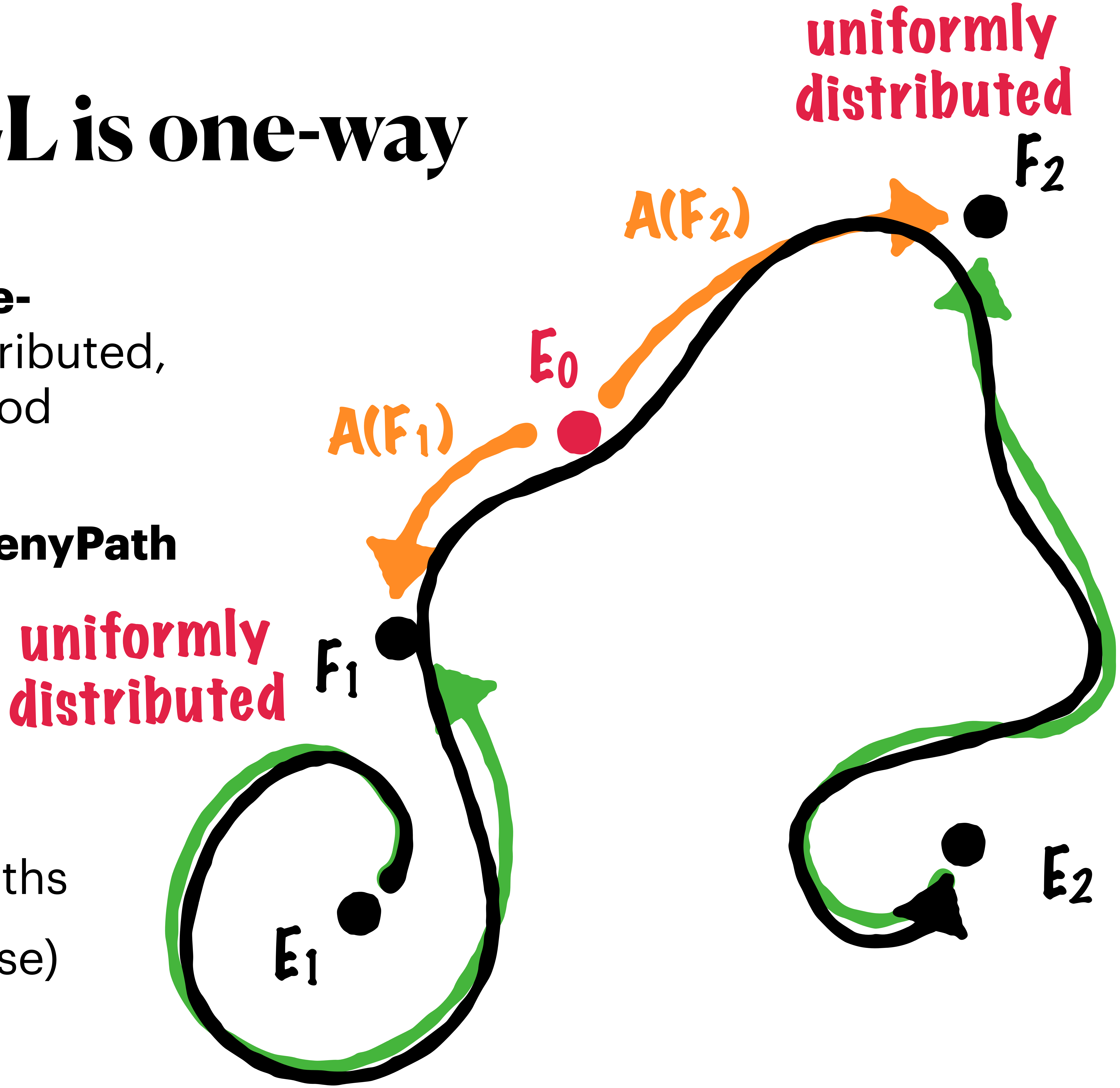
CGL is one-way

- Let **A** an algorithm **breaking one-wayness**: given E uniformly distributed, **A**(E) finds a path $E_0 \to E$ with good probability



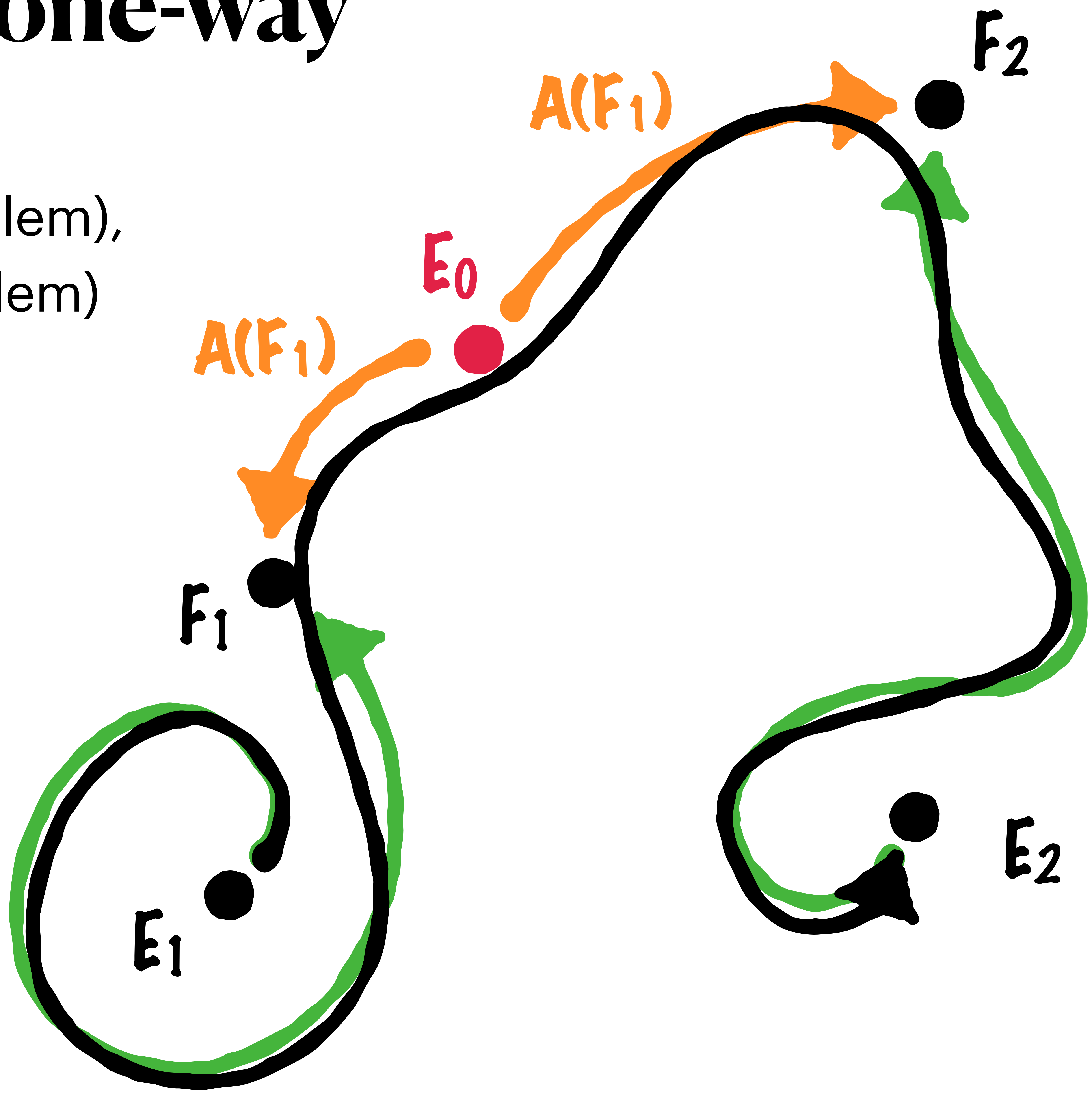
CGL is one-way

- Let **A** an algorithm **breaking one-wayness**: given E uniformly distributed, $\mathbf{A}(E)$ finds a path $E_0 \to E$ with good probability
- Let (E_1, E_2) an instance of **ℓ-IsogenyPath**
  1. Random walk $E_1 \to F_1$
  2. Call $\mathbf{A}(F_1)$
  3. Same for E_2...
  4. Return concatenation of paths
- Solves **ℓ-IsogenyPath** (worst case)



CGL is one-way

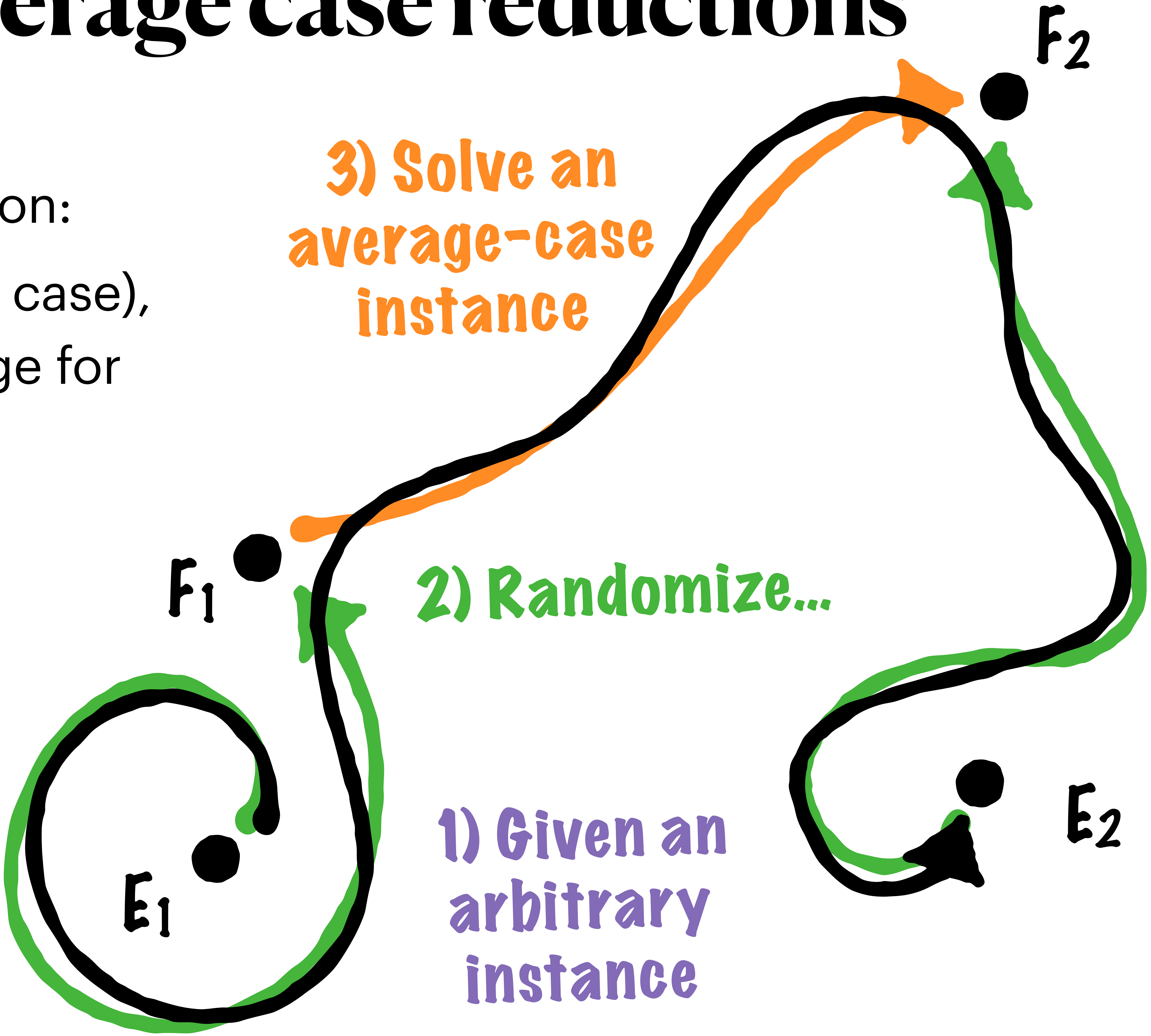
If ℓ-**IsogenyPath** is hard (worst case problem), then **CGL is one-way** (average-case problem)



Worst-case to average case reductions

A worst-case to average-case reduction:

If **ℓ-IsogenyPath** is hard (in the worst case), then **ℓ-IsogenyPath** is hard on average for uniformly random input



Which is hardest? Easiest?

EndRing

MaxOrder

ℓ-IsogenyPath

OneEnd

Isogeny

Assuming GRH, **if any of these is hard in the worst case, then all are hard on average!**

Without GRH, almost always true.

[Herlédan Le Merdy, W. — to appear] *Unconditional foundations for supersingular isogeny-based cryptography*

Solving ℓ-IsogenyPath

and Isogeny, EndRing, OneEnd, MaxOrder...

How hard are they?

EndRing

MaxOrder

ℓ-IsogenyPath

OneEnd

Isogeny

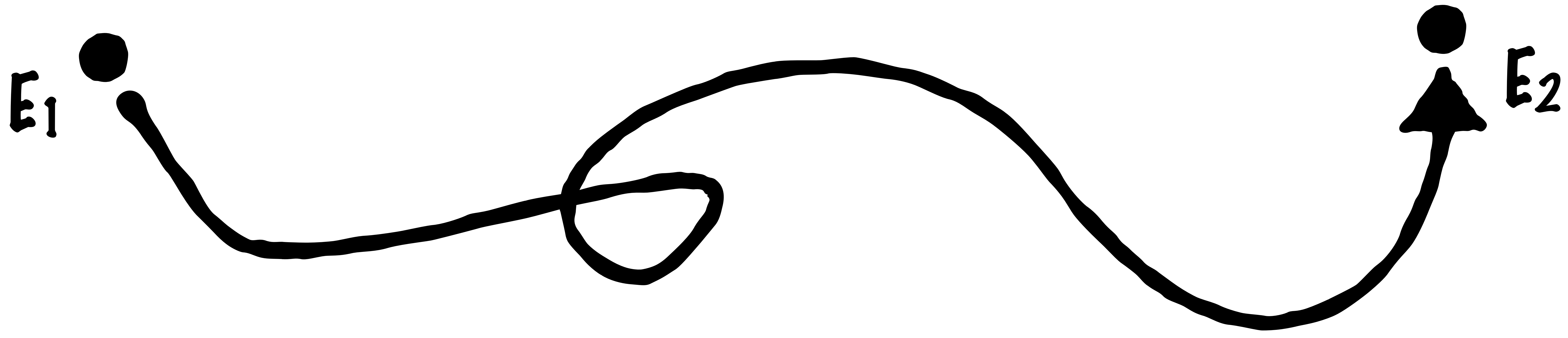
They are all as hard as each other...

But **how hard** is that?

Solving ℓ-IsogenyPath

The *ℓ-IsogenyPath* problem

Given E_1 and E_2 (supersingular) find an ℓ-isogeny path from E_1 to E_2

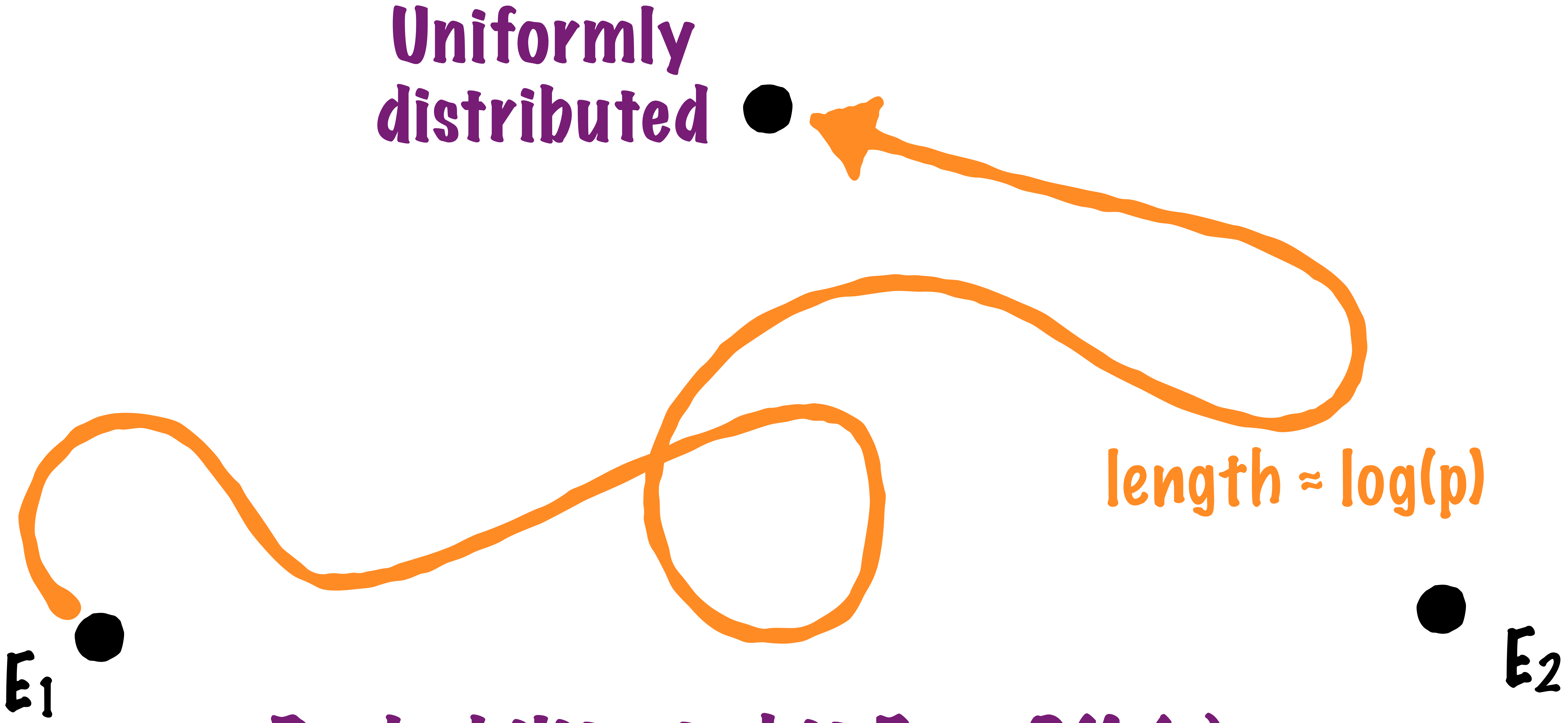


Solving ℓ-IsogenyPath

The supersingular ℓ-isogeny graph

- Approximately $p/12$ vertices
- Ramanujan

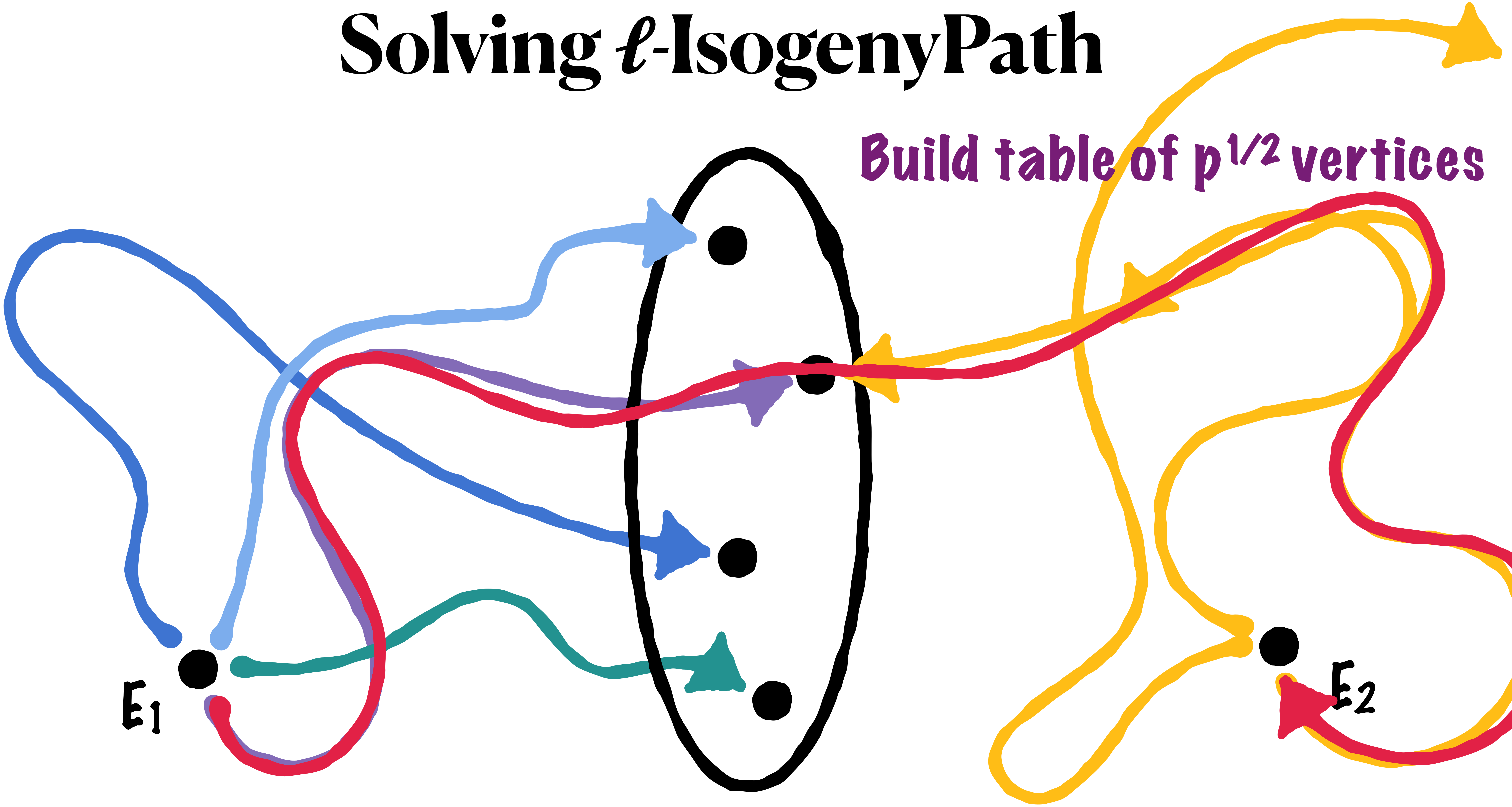
Solving ℓ-IsogenyPath



Probability to hit $E_2 = O(1/p)$

Success after $O(p)$ attempts...

Solving ℓ-IsogenyPath



Success after $O(p^{1/2})$ attempts!

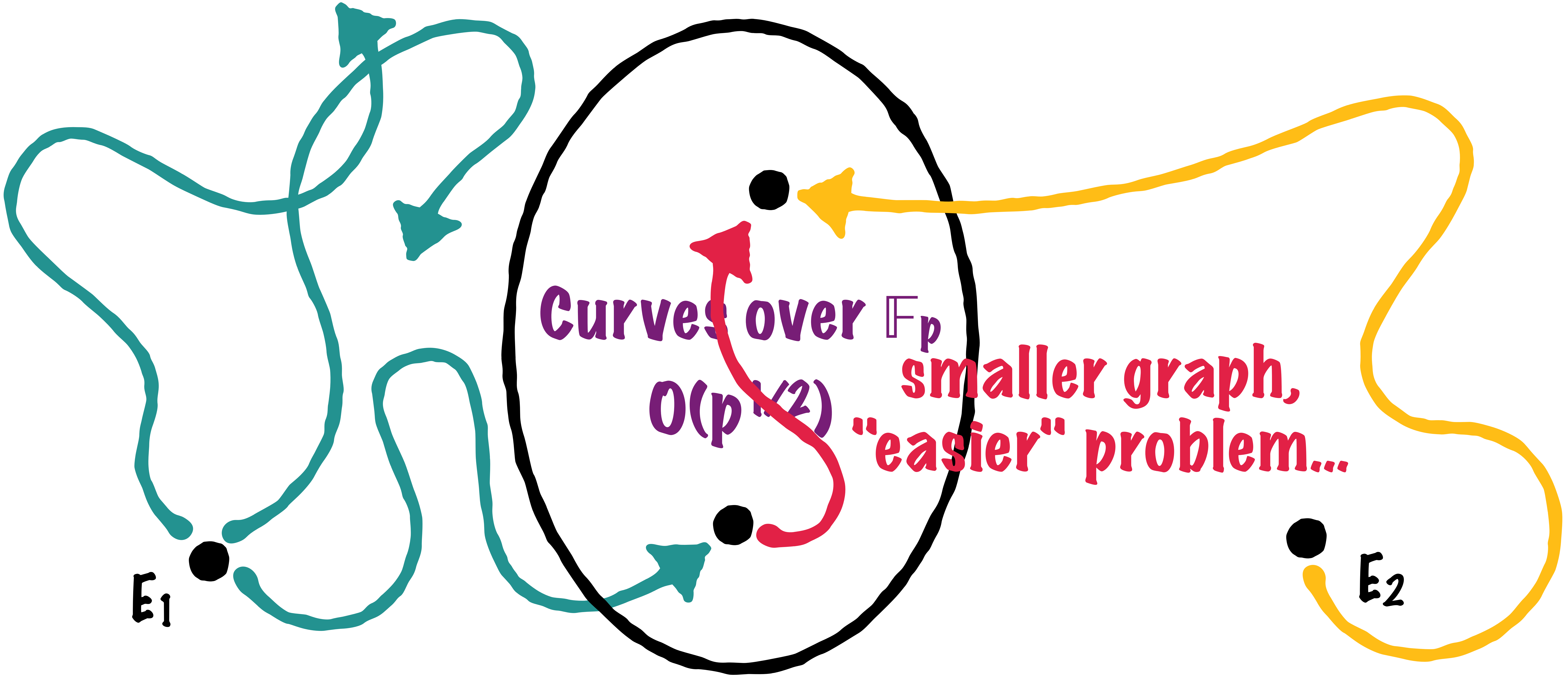
Solving ℓ-IsogenyPath

Theorem: There is an algorithm for **ℓ-IsogenyPath** in time $\tilde{O}(p^{1/2})$

Corollary: One can solve **Isogeny**, **EndRing**, **MaxOrder** and **OneEnd** in time $\tilde{O}(p^{1/2})$

Theorem [Delfs, Galbraith — DCC 2016]**:** There is an algorithm for **Isogeny** in time $\tilde{O}(p^{1/2})$ and space $\log(p)^{O(1)}$

Solving Isogeny



Success after $O(p^{1/2})$ attempts!

OneEnd = EndRing

OneEnd to find them all

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…*

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

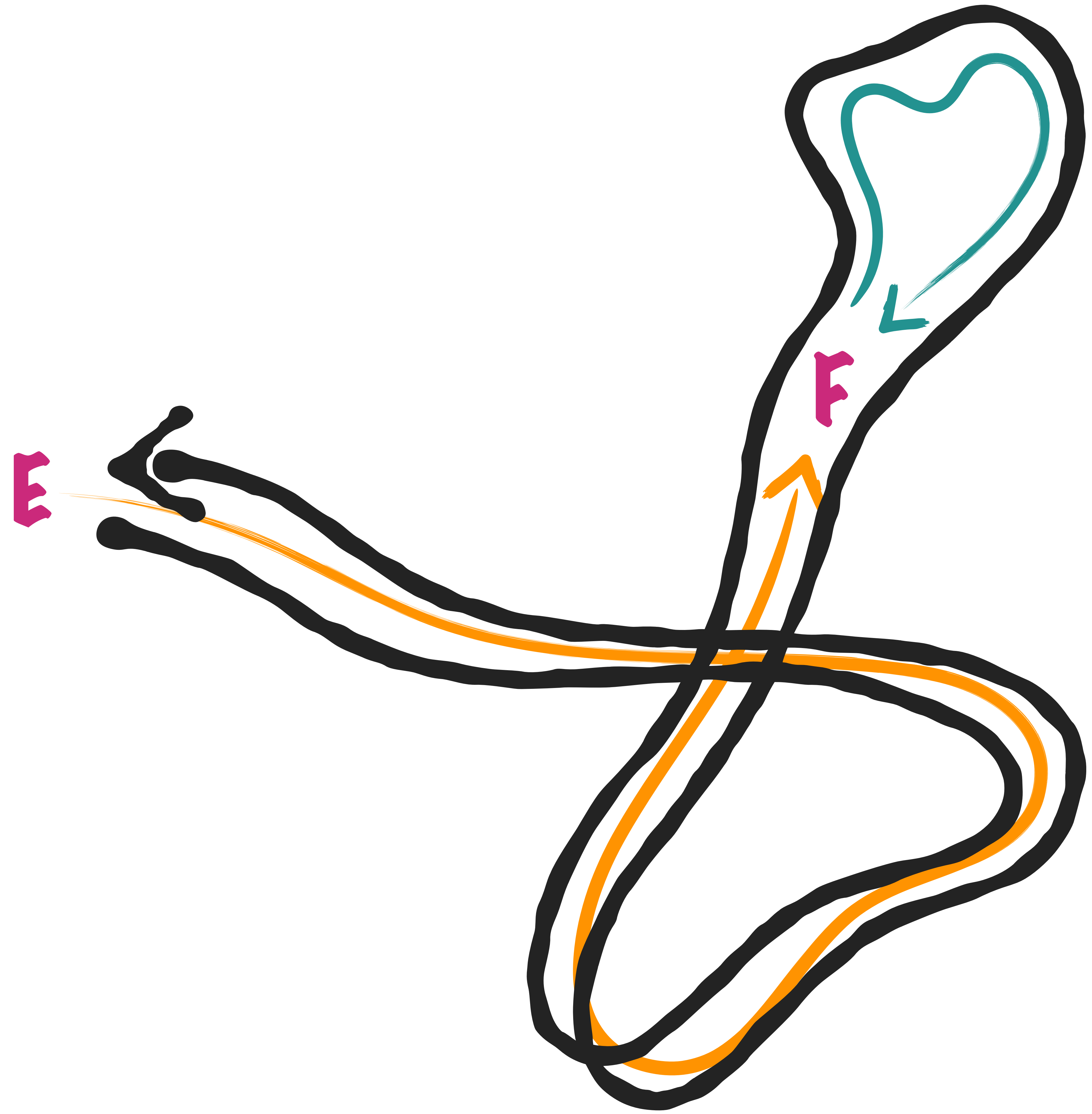
1. For i = 1, 2,… call $\mathcal{O}(E)$, which returns some $\alpha_i \in \mathrm{End}(E) \setminus \mathbb{Z}$
2. As soon as $(\alpha_i)_i$ generates End(E), extract a basis and return it

Efficient linear algebra!

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 $\mathtt{Rich}^{\mathcal{O}}$

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 $\mathtt{\textbf{Rich}}^{\mathcal{O}}(E)$, 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

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 $\beta^{-1}\alpha\beta$

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

Stabilization

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

Select E, call $\alpha \leftarrow \mathcal{O}(E)$, return (E, α) Random variable with distribution D_0

A "random walk operator" T on the space of probability distributions of (E, α)

Select E, a random isogeny $\varphi : E \to F$, call $\alpha \leftarrow \mathcal{O}(F)$, return $(E, \hat{\varphi} \circ \alpha \circ \varphi)$ $D_1 = T(D_0)$

- Long walk (i.e., large degree φ) $\Rightarrow$ $T(D_0)$ converges to a stationary distribution
- Stationary distribution $\Rightarrow$ stable under conjugation
- **Spectral analysis** of the operator T gives **convergence speed**

Stabilization

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

Deuring correspondence

Elliptic curves → Quaternions

The random walk operator is a Hecke operator on quaternionic automorphic forms

+

Jacquet-Langlands correspondence

Quaternions → Modular forms

Eigenvalues of the Hecke operator can be read off the coefficients of a classical modular form

+

Deligne's bound on coefficients of modular forms

Modular forms → … Modular forms

Bounds on coefficients imply bounds on eigenvalues of the random walk operator

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

Terminates! 👍

In exponential time... 👎

Faster stabilization

Idea 3: Stabilization can be made much faster by "reducing" each oracle output α_i.

Next problem: "Reducing" requires factoring large integers...

Idea 4: "Lazy reduction": do a partial factorization, and if something fails, it reveals a new factor

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

Polynomial time!

$$
\begin{tikzcd}
\text{EndRing} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \text{MaxOrder} \arrow[d, leftrightarrow] & \ell\text{-IsogenyPath} \arrow[l, "\text{GRH}"'] \\
\text{OneEnd} \arrow[r, leftrightarrow] & \text{Isogeny} \arrow[ur] &
\end{tikzcd}
$$