SQIsignHD

Sqiing in higher dimensions

Benjamin Wesolowski, CNRS and ENS de Lyon

November 2024, *Emerging topics in design and cryptanalysis of post-quantum schemes*, Paris, France

Based on joint works with **Andrea Basso**, **Pierrick Dartois**, **Luca De Feo**, **Antonin Leroux**, **Luciano Maino**, **Giacomo Pope**, and **Damien Robert**

SQIsign & friends

Isogeny-based signature schemes

Picture by Beppe Rijs

SQIsign

[De Feo, Kohel, Leroux, Petit, W. — Asiacrypt 2020] SQISign: compact post-quantum signatures from quaternions and isogenies

- **Isogeny-based** post-quantum signature scheme
- **Very compact**: PK + Signature combined **5× smaller** than Falcon

Secret key (bytes)	Public key (bytes)	Signature (bytes)	Security
16	64	204	NIST-I

SQIsign

[De Feo, Kohel, Leroux, Petit, W. — Asiacrypt 2020] SQISign: compact post-quantum signatures from quaternions and isogenies

- **Isogeny-based** post-quantum signature scheme
- **Very compact**: PK + Signature combined **5× smaller** than Falcon

Secret key (bytes)	Public key (bytes)	Signature (bytes)	Security
16	64	204	NIST-I

	Key gen. (MCycles)	Signing (MCycles)	Verif. (MCycles)
Original SQIsign	2800	4600	93
Optimized SQIsign	400	1880	29

SQIsign

[De Feo, Kohel, Leroux, Petit, W. — Asiacrypt 2020] SQISign: compact post-quantum signatures from quaternions and isogenies

- **Isogeny-based** post-quantum signature scheme
- **Very compact**: PK + Signature combined **5× smaller** than Falcon

Secret key (bytes)	Public key (bytes)	Signature (bytes)	Security
16	64	204	NIST-I

	Key gen. (MCycles)	Signing (MCycles)	Verif. (MCycles)
Original SQIsign	2800	4600	93
Optimized SQIsign	400	1880 620ms	29 10ms

Drawbacks of SQIsign

- Signing in 600ms is **too slow**
- Security proof: the ZK property is based on an ***ad hoc* assumption**
- **Bad scaling** to higher security levels (signing at NIST-V takes 40s)

Drawbacks of SQIsign

- Signing in 600ms is **too slow**
- Security proof: the ZK property is based on an ***ad hoc* assumption**
- **Bad scaling** to higher security levels (signing at NIST-V takes 40s)

SQIsignHD solves all of these

[Dartois, Leroux, Robert, W. — Eurocrypt 2024]

Drawbacks of SQIsign

- Signing in 600ms is **too slow**
- Security proof: the ZK property is based on an ***ad hoc* assumption**
- **Bad scaling** to higher security levels (signing at NIST-V takes 40s)

SQIsignHD solves all of these

[Dartois, Leroux, Robert, W. — Eurocrypt 2024]

Verification gets slower…

Drawbacks of SQIsign

- Signing in 600ms is **too slow**
- Security proof: the ZK property is based on an ***ad hoc* assumption**
- **Bad scaling** to higher security levels (signing at NIST-V takes 40s)

SQIsignHD solves all of these

[Dartois, Leroux, Robert, W. — Eurocrypt 2024]

Verification gets slower… Problem solved with

SQIsign2D

SQIsign2D-West (this talk) [Basso, Dartois, De Feo, Leroux, Maino, Pope, Robert, W. — Asiacrypt 2024]
SQIsign2D-East [Nakagawa, Onuki, Castryck, Chen, Invernizzi, Lorenzon, Vercauteren — Asiacrypt 2024]

The Isogeny problem

and how to represent an isogeny

Picture by Beppe Rijs

Elliptic curves

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

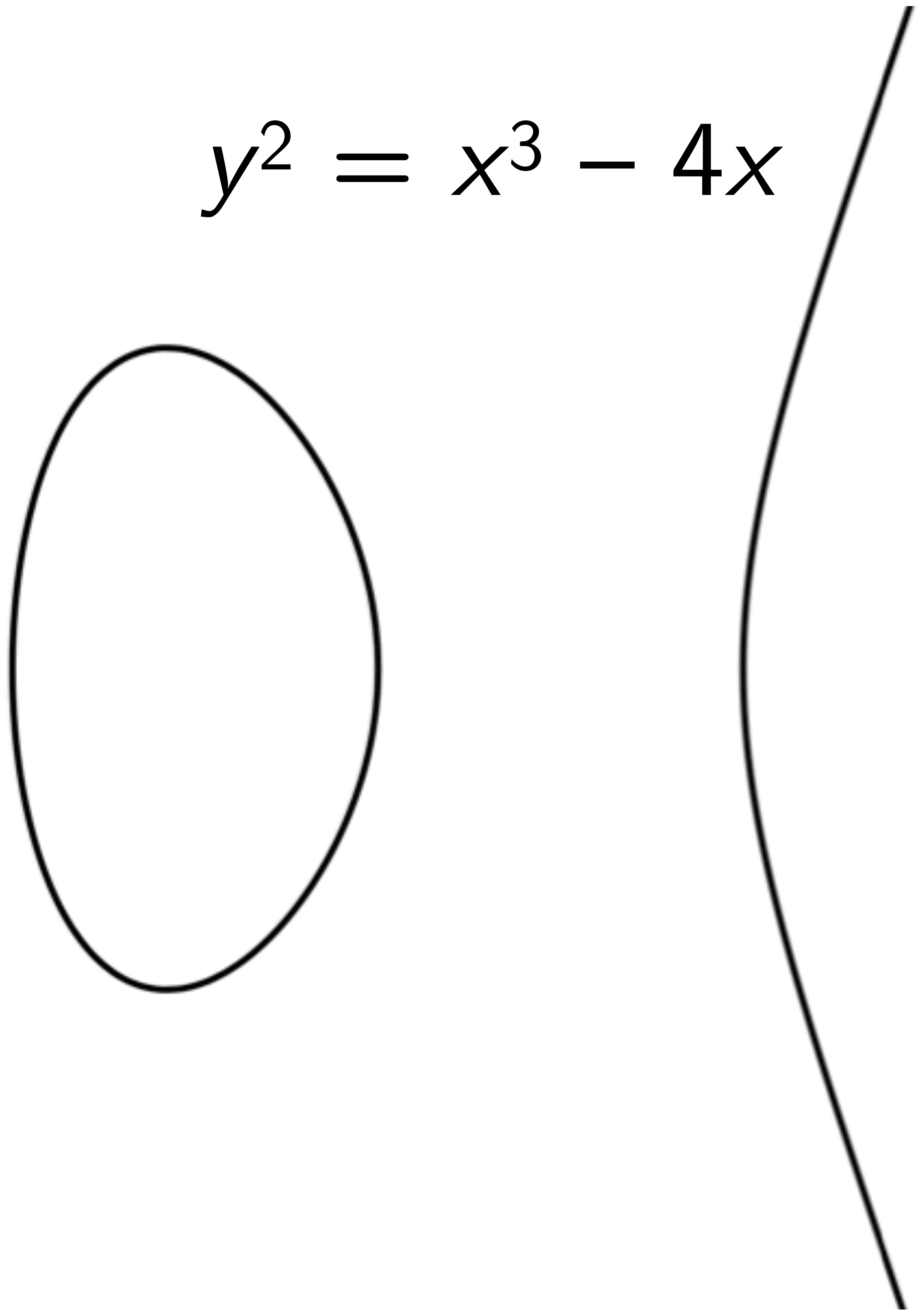
equations of the form

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

$y^2 = x^3 + x$

E_1

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

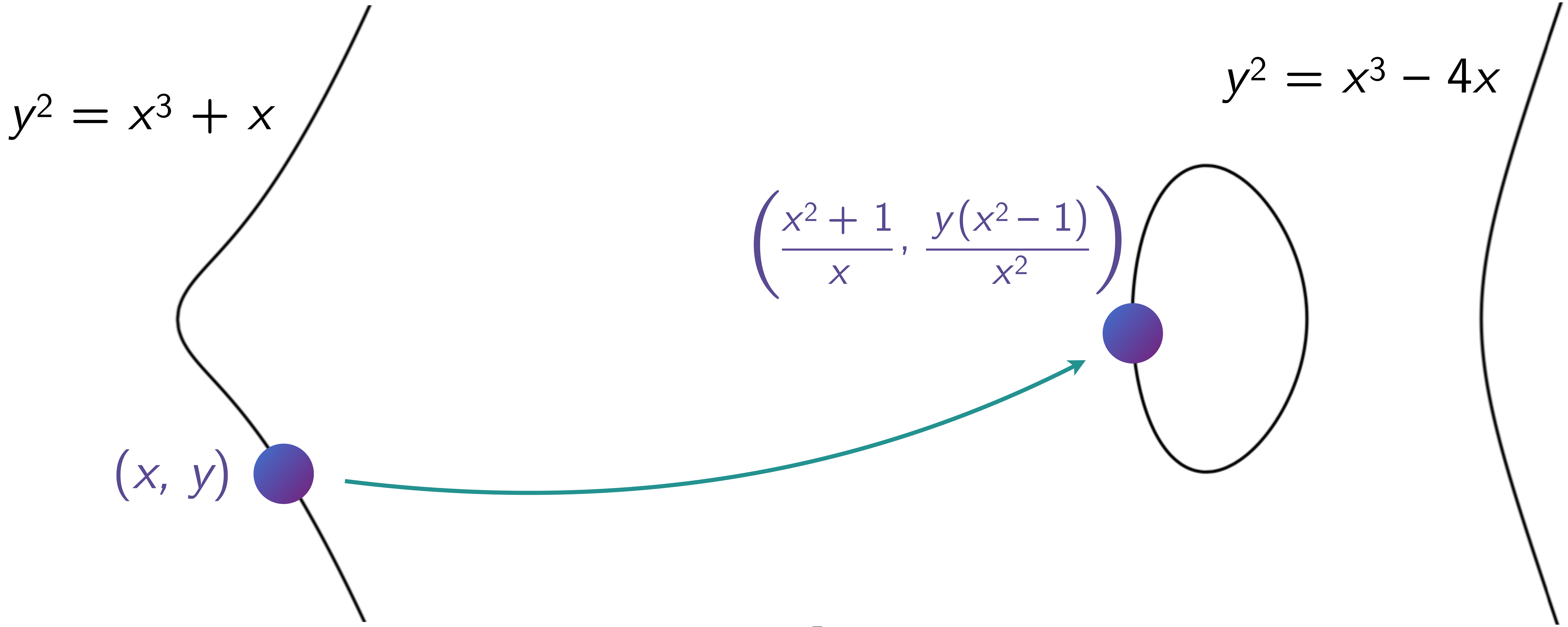


E_2

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

Elliptic curves

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

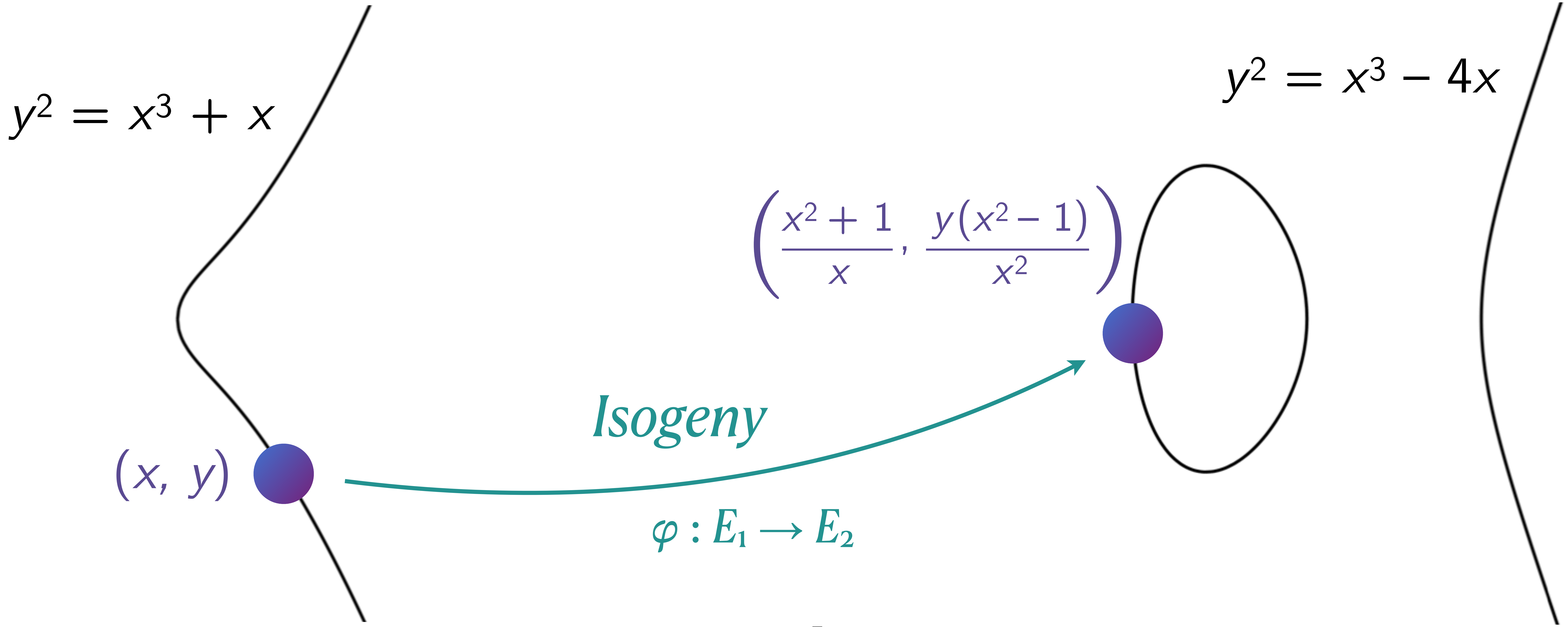


Elliptic curves

- 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

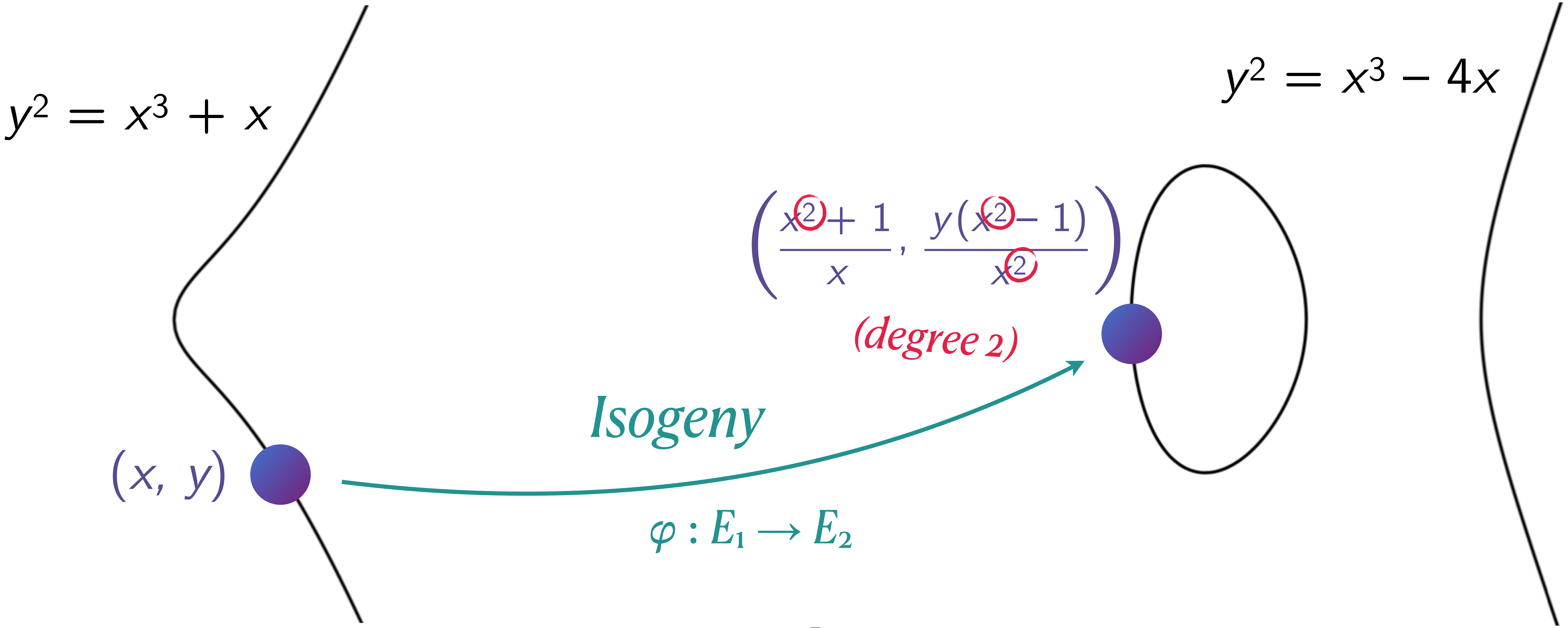


Elliptic curves

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

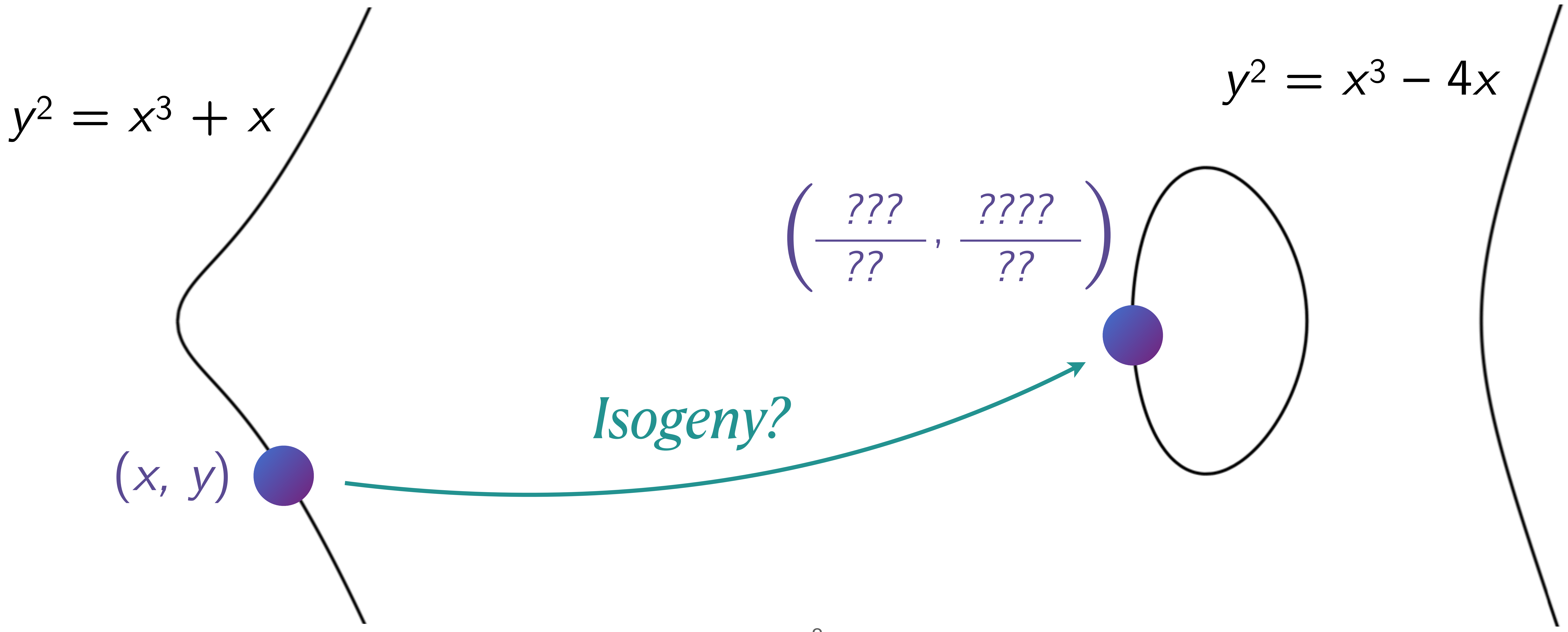
- Isogenies are **group homomorphisms**
- **Degree** = size of kernel

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 $\varphi : E_1 \to E_2$

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

The *Isogeny* problem

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

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

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

The *Isogeny* problem

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

- 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...

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…

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 combinations* 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?

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

(degree 2)

fine for small degree...

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

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

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 combinations* of "small" ones

$$\begin{tikzcd} E_1 \arrow[r, "2"] \arrow[rrrr, bend right, "\text{degree } 2^{256}"'] & E_2 \arrow[r, "2"] & E_3 \arrow[r, "2"] & \ldots \arrow[r, "2"] & E_{257} \end{tikzcd}$$

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?

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

(degree 2)

fine for small degree...

- Build "big" isogenies as *formal combinations* of "small" ones
  - $\varphi \circ \psi$ represented by ($'\mathtt{comp}'$, φ, ψ) where φ and ψ are in efficient repr.
  - $\varphi + \psi$ represented by ($'\mathtt{add}'$, φ, ψ) where φ and ψ are both $E_1 \to E_2$

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?
  - any **efficient representation**: an encoding which allows one to evaluate $\varphi(P)$ in polynomial time for any P

Isogeny graph

$$E_1 \longrightarrow E_2$$

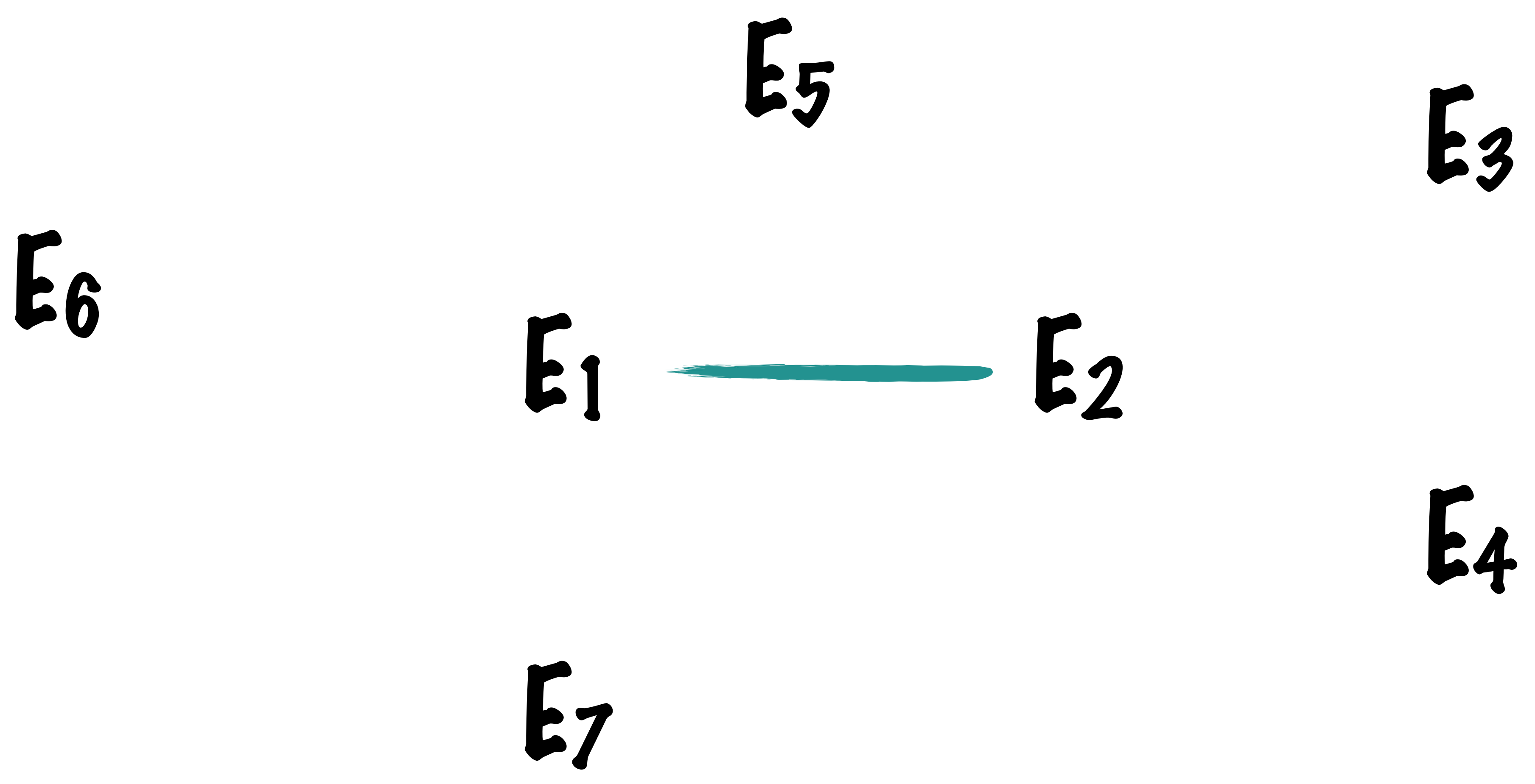
an isogeny of degree 2 = an edge in a graph

Isogeny graph

- **The 2-isogeny graph** (supersingular…)

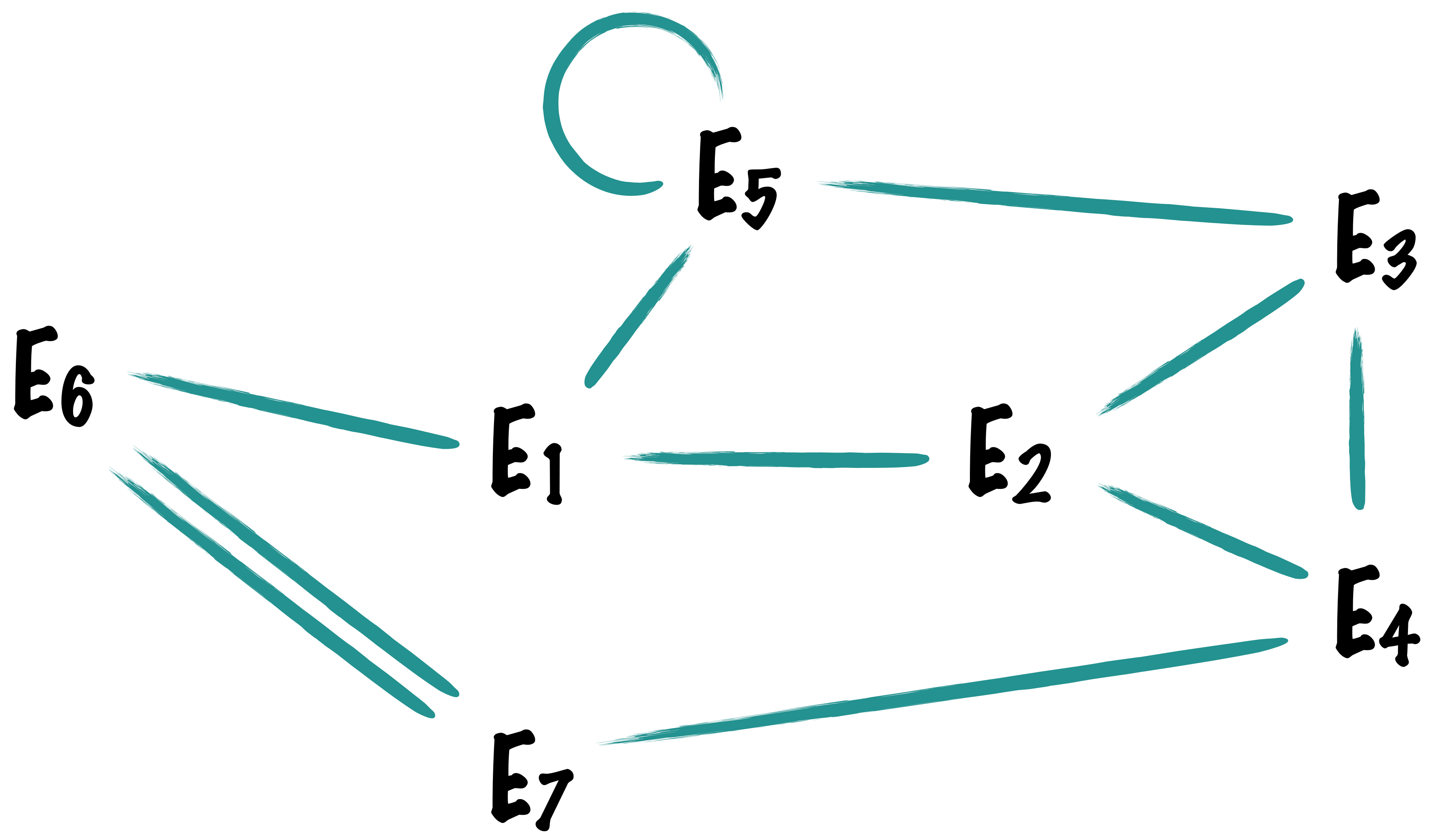
Isogeny graph

- **The 2-isogeny graph** (supersingular…)



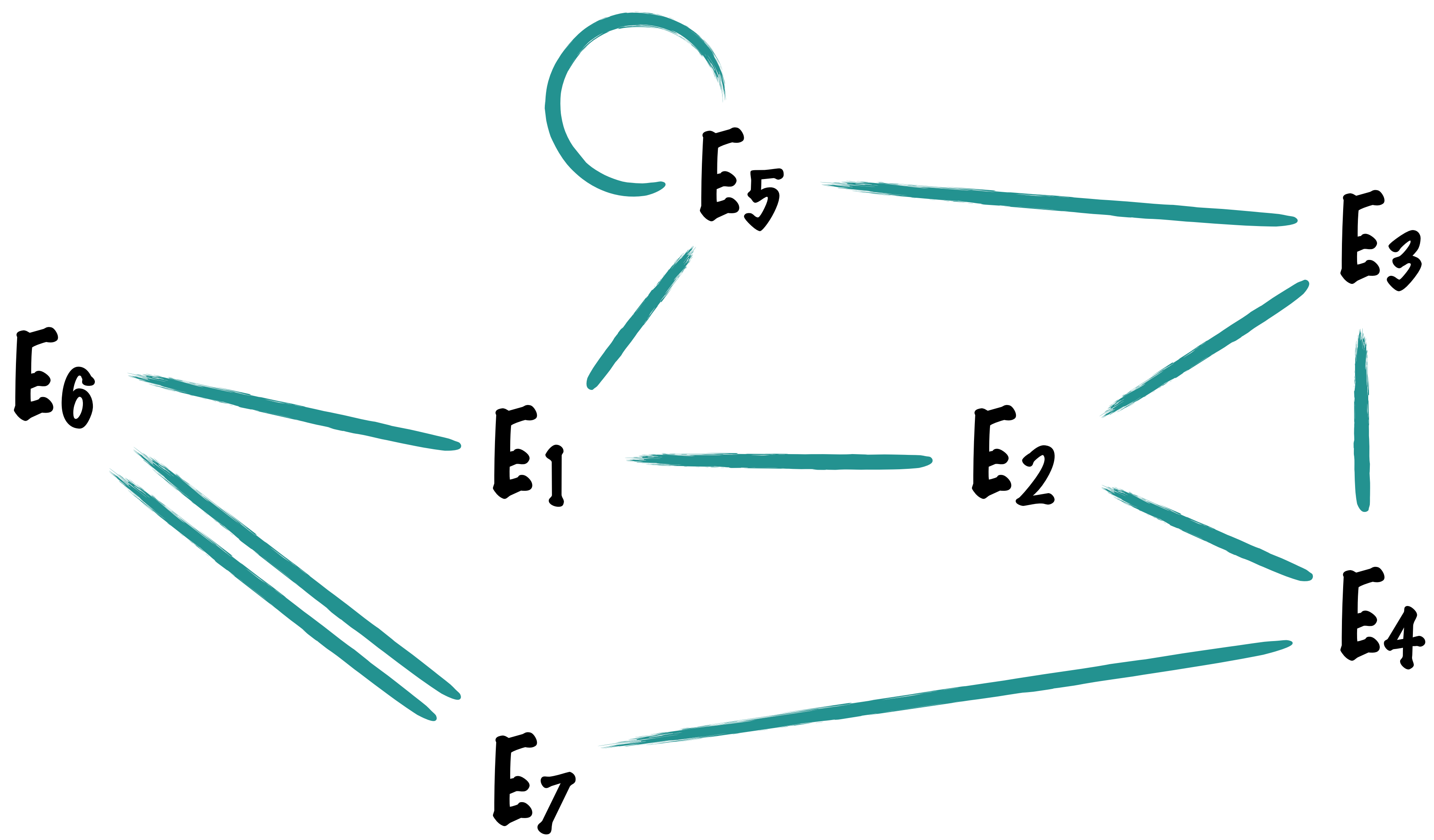
Isogeny graph

- **The 2-isogeny graph** (supersingular...)



Isogeny graph

- **The 2-isogeny graph** (supersingular...)



- 3-regular, **connected** (for supersingular curves)


205
206
166
165
164
167
163
162
168
202
161
169
101
100
99
201
103
102
160
104
98
170
97
210
105
96
106
171
95
159
94
172
54
53
55
52
200
107
56
158
173
51
93
57
157
50
108
92
58
28
27
174
49
29
26
91
109
59
156
90
110
30
13
48
155
60
14
175
111
25
199
89
154
112
12
61
198
47
88
31
6
15
113
153
24
87
114
62
152
5
46
176
115
2
11
151
32
7
63
86
116
23
45
150
16
85
1
3
177
117
64
33
44
84
196
149
22
4
118
195
10
178


205
206
166
165
164
167
163
162
202
168
161
169
101
100
102
103
99
98
201
160
104
97
210
170
105
96
159
106
171
95
94
172
55
54
53
52
200
158
51
107
56
173
93
57
157
50
108
28
27
92
58
174
29
26
49
91
109
59
90
156
110
30
13
48
155
60
14
25
199
175
111
89
154
12
112
61
198
88
47
31
15
6
113
153
24
87
114
62
152
5
46
176
115
2
11
151
32
7
86
63
116
23
45
150
16
85
1
3
33
117
177
64
196
44
84
149
4
22
10
118
195
178


205 206 166 165 164 167 163 162 202
168 169 161 201 101 102 103 100 99 98 97 104 160
170 105 96 210 171 106 95 159 94
172 55 54 53 52 200 56 107 51 158 173 93 57 157
108 50 28 27 58 92 174 29 26 49 91 109 59 156
110 90 13 48 30 175 60 14 111 25 155 199 89
112 12 154 61 31 47 198 6 15 88 113 153
62 114 24 5 87 152 176 46 115 32 2 11 151
7 86 63 116 23 45 16 85 150 1 3
177 117 64 33 44 84 196 149 118 4 22 10 195 178

205
206
166
165
164
167
163
162
168
202
169
161
101
100
102
103
99
98
160
201
104
97
170
210
105
96
171
106
95
159
94
172
55
54
53
52
200
107
56
51
158
173
93
57
50
157
108
58
28
27
92
174
29
49
91
109
59
26
156
110
90
30
13
48
155
60
14
25
199
175
111
89
112
12
154
61
31
47
198
113
15
6
88
153
114
62
24
87
152
176
5
46
115
32
2
11
151
63
7
86
116
16
23
45
150
85
177
117
3
1
33
64
44
84
196
149
118
4
22
10
195


205
206
166
165
164
167
163
162
202
168
169
103
102
101
100
99
98
161
201
170
104
97
160
210
105
96
171
106
95
159
94
172
55
54
53
52
200
107
56
51
158
173
93
57
50
157
108
58
28
27
92
174
109
59
29
26
49
91
156
110
13
90
30
48
155
175
111
60
14
25
89
199
112
12
154
61
31
15
6
47
88
198
113
153
114
62
24
87
152
176
115
5
46
32
2
7
11
151
63
116
16
86
23
45
150
85
177
117
3
1
33
64
44
84
149
196
118
4
22
195
10

205
206
166
165
164
167
163
162
202
168
169
103
102
101
100
99
98
161
201
210
160
170
104
105
97
96
171
106
95
159
172
55
54
53
94
200
52
158
173
107
56
51
93
57
157
108
50
58
28
27
92
174
109
29
49
91
26
156
59
90
110
13
30
48
155
60
199
175
111
14
25
89
12
154
112
61
15
6
47
88
198
113
153
114
62
24
87
152
176
5
46
115
2
11
151
32
7
86
63
116
23
45
150
16
85
3
1
177
117
64
33
44
84
149
196
4
22
118
10
195


205
206
166
165
164
167
163
162
168
202
169
161
201
103
102
101
100
99
98
97
210
104
160
170
105
96
95
159
171
106
94
172
55
54
53
52
51
200
158
107
56
93
173
57
157
50
108
58
28
27
92
174
109
49
91
59
29
26
110
90
156
60
30
13
48
155
199
111
14
25
89
175
112
12
154
61
88
198
47
41
6
113
15
153
24
62
114
5
87
152
46
176
115
2
11
32
7
151
63
86
116
16
23
45
150
85
177
117
3
1
64
33
44
196
84
149
22
118
4
10
195


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


Uniformly distributed

Endomorphisms

and computational problems

Picture by Beppe Rijs

Endomorphism ring

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

Endomorphism ring

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

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

Endomorphism ring

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

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

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

Endomorphism ring

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

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

- $\varphi + \psi$ 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 $\mathbb{Z} \subset \mathrm{End}(E)$...

Endomorphism ring

What is the structure of End(E)?

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

Endomorphism ring

What is the structure of End(E)?

- It contains $\mathbb{Z} \subset \mathrm{End}(E)$...
- $(\mathrm{End}(E), +)$ 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

Endomorphism ring

What is the structure of $\mathrm{End}(E)$?

- It contains $\mathbb{Z} \subset \mathrm{End}(E)$...
- $(\mathrm{End}(E), +)$ 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,

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

The endomorphism ring problem

For E **supersingular** $\mathrm{End}(E) = \{\varphi : E \to E\}$ is a lattice of dimension 4

The *EndRing* problem

Given E (supersingular) find 4 generators of the endomorphism ring $\mathrm{End}(E)$

The endomorphism ring problem

For E **supersingular** $\mathrm{End}(E) = \{\varphi : E \to E\}$ is a lattice of dimension 4

The *EndRing* problem

Given E (supersingular) find 4 generators of the endomorphism ring $\mathrm{End}(E)$

Solution of the form $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$...

- How are α_i represented? Any *efficient representation*

Example

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

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

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-scalar endomorphisms:

Example

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

Consider $\boldsymbol{E_0 : y^2 = x^3 + x}$

Two non-scalar endomorphisms:

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

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-scalar endomorphisms:

- $\iota : E_0 \longrightarrow E_0 : (x, y) \longmapsto (-x, \alpha y)$ $\iota^2 = [-1]$

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-scalar endomorphisms:

- $\iota : E_0 \longrightarrow E_0 : (x, y) \longmapsto (-x, \alpha y)$ $\qquad \iota^2 = [-1]$
- $\pi : E_0 \longrightarrow E_0 : (x, y) \longmapsto (x^p, y^p)$

Example

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

$$\textbf{Consider } E_0 : y^2 = x^3 + x$$

Two non-scalar endomorphisms:

- $\iota : E_0 \to E_0 : (x, y) \longmapsto (-x, \alpha y)$ $\qquad \iota^2 = [-1]$
- $\pi : E_0 \to E_0 : (x, y) \longmapsto (x^p, y^p)$ $\qquad \iota\pi = -\pi\iota$

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-scalar endomorphisms:

- $\iota : E_0 \to E_0 : (x, y) \longmapsto (-x, \alpha y)$ $\quad \iota^2 = [-1]$
- $\pi : E_0 \to E_0 : (x, y) \longmapsto (x^p, y^p)$ $\quad \iota\pi = -\pi\iota$

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

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-scalar endomorphisms:

- $\iota : E_0 \to E_0 : (x, y) \longmapsto (-x, \alpha y)$ $\quad\iota^2 = [-1]$
- $\pi : E_0 \to E_0 : (x, y) \longmapsto (x^p, y^p)$ $\quad\iota\pi = -\pi\iota$

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

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-scalar endomorphisms:

- $\iota : E_0 \to E_0 : (x, y) \longmapsto (-x, \alpha y)$ $\iota^2 = [-1]$
- $\pi : E_0 \to E_0 : (x, y) \longmapsto (x^p, y^p)$ $\iota\pi = -\pi\iota$

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

EndRing

EndRing problem ⇔ **Isogeny** problem

Computing End(–) $\Leftrightarrow$ Finding isogenies

[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*.

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

Computing End(–) ⇔ Finding isogenies

End(–) is a GPS that allows you to find your way between supersingular curves:

- given End(E_1) and End(E_2), one can find an isogeny $E_1 \to E_2$ in poly. time

[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*.

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

Computing End(–) ⇔ Finding isogenies

End(–) is a GPS that allows you to find your way between supersingular curves:

- given End(E_1) and End(E_2), one can find an isogeny $E_1 \to E_2$ in poly. time

You can update the GPS coordinates as you travel through isogenies:

- given End(E_1), and a (smooth) isogeny $E_1 \to E_2$, one can find End(E_2) in poly. time

[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*.

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

Computing End(–) $\Leftrightarrow$ Finding isogenies

End(–) is a GPS that allows you to find your way between supersingular curves:

- given End(E_1) and End(E_2), one can find an isogeny $E_1 \to E_2$ in poly. time

You can update the GPS coordinates as you travel through isogenies:

- given End(E_1), and a (smooth) isogeny $E_1 \to E_2$, one can find End(E_2) in poly. time

For E_1, E_2 supersingular, Hom(E_1, E_2) is a lattice of rank 4

Computing End(–) ⇔ Finding isogenies

End(–) is a GPS that allows you to find your way between supersingular curves:

- given End(E_1) and End(E_2), one can find ~~an isogeny $E_1 \to E_2$ in poly. time~~
  a basis of Hom(E_1, E_2)

You can update the GPS coordinates as you travel through isogenies:

- given End(E_1), and a (smooth) isogeny $E_1 \to E_2$, one can find End(E_2) in poly. time

For E_1, E_2 supersingular, Hom(E_1, E_2) is a lattice of rank 4

Computing End(–) ⇔ Finding isogenies

End(–) is a GPS that allows you to find your way between supersingular curves:

- given End(E_1) and End(E_2), one can find ~~an isogeny $E_1 \to E_2$ in poly. time~~
  a basis of Hom(E_1, E_2)

You can update the GPS coordinates as you travel through isogenies:

- given End(E_1), and a (smooth) isogeny $E_1 \to E_2$, one can find End(E_2) in poly. time

For E_1, E_2 supersingular, Hom(E_1, E_2) is a lattice of rank 4

Computing End(–) ⇔ Computing Hom(–, –)

Key generation

Generating a curve with its endomorphism ring

Picture by Beppe Rijs

Idea of SQIsign

Basic idea of SQIsign:

- **Public key**: a supersingular curve E_{pk}
- **Secret key**: a basis of $\mathrm{End}(E_{\mathrm{pk}})$

Idea of SQIsign

Basic idea of SQIsign:

- **Public key**: a supersingular curve E_{pk}
- **Secret key**: a basis of $\mathrm{End}(E_{\mathrm{pk}})$

Key recovery = EndRing

Idea of SQIsign

Basic idea of SQIsign:

- **Public key**: a supersingular curve E_{pk}
- **Secret key**: a basis of $\mathrm{End}(E_{\mathrm{pk}})$
- **SQIsign proof of knowledge**: a sigma protocol to prove knowledge of $\mathrm{End}(E_{\mathrm{pk}})$

Key recovery = EndRing

Idea of SQIsign

Basic idea of SQIsign:

- **Public key**: a supersingular curve E_{pk}
- **Secret key**: a basis of $\mathrm{End}(E_{\mathrm{pk}})$
- **SQIsign proof of knowledge**: a sigma protocol to prove knowledge of $\mathrm{End}(E_{\mathrm{pk}})$
- **SQIsign**: Fiat-Shamir transform of the proof of knowledge

Key recovery = EndRing

Idea of SQIsign

Basic idea of SQIsign:

- **Public key**: a supersingular curve E_{pk}
- **Secret key**: a basis of $\mathrm{End}(E_{\mathrm{pk}})$
- **SQIsign proof of knowledge**: a sigma protocol to prove knowledge of $\mathrm{End}(E_{\mathrm{pk}})$
- **SQIsign**: Fiat-Shamir transform of the proof of knowledge

Key recovery = EndRing

How to generate a random E_{pk} together with $\mathrm{End}(E_{\mathrm{pk}})$?

Idea of SQIsign

Basic idea of SQIsign:

- **Public key**: a supersingular curve E_{pk}
- **Secret key**: a basis of $\text{End}(E_{\text{pk}})$
- **SQIsign proof of knowledge**: a sigma protocol to prove knowledge of $\text{End}(E_{\text{pk}})$
- **SQIsign**: Fiat-Shamir transform of the proof of knowledge

Key recovery = EndRing

How to generate a random E_{pk} together with $\text{End}(E_{\text{pk}})$?

How to generate even a single supersingular curve?

A special supersingular curve

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

Consider $\boldsymbol{E_0 : y^2 = x^3 + x}$

Two non-trivial endomorphisms:

- $\iota : E_0 \to E_0 : (x, y) \mapsto (-x, \alpha y)$ $\qquad \iota^2 = [-1]$
- $\pi : E_0 \to E_0 : (x, y) \mapsto (x^p, y^p)$ $\qquad \iota\pi = -\pi\iota$

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

A special supersingular curve

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

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

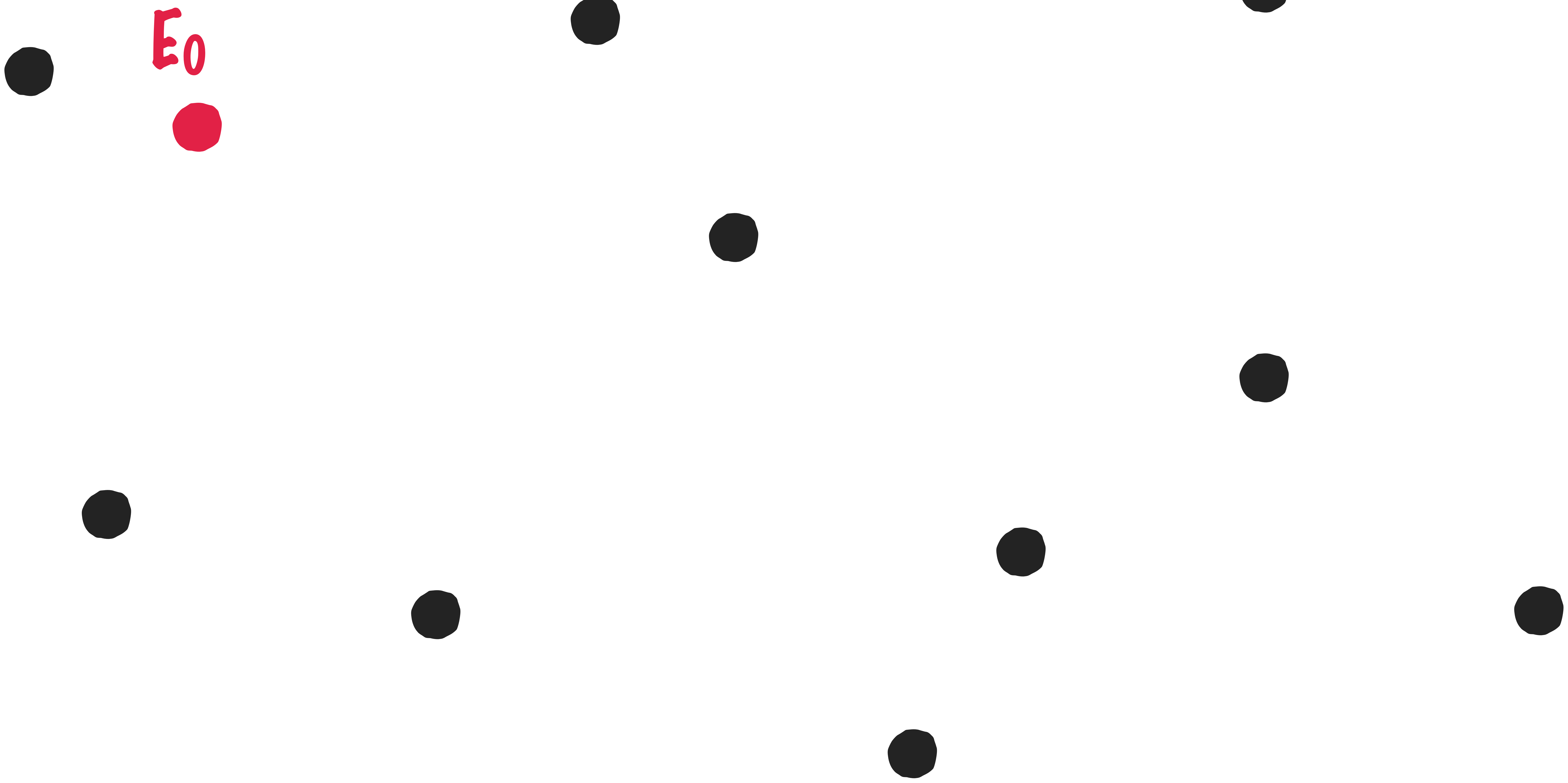
Two non-trivial endomorphisms:

- $\iota : E_0 \to E_0 : (x, y) \longmapsto (-x, \alpha y)$ $\qquad \iota^2 = [-1]$
- $\pi : E_0 \to E_0 : (x, y) \longmapsto (x^p, y^p)$ $\qquad \iota\pi = -\pi\iota$

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

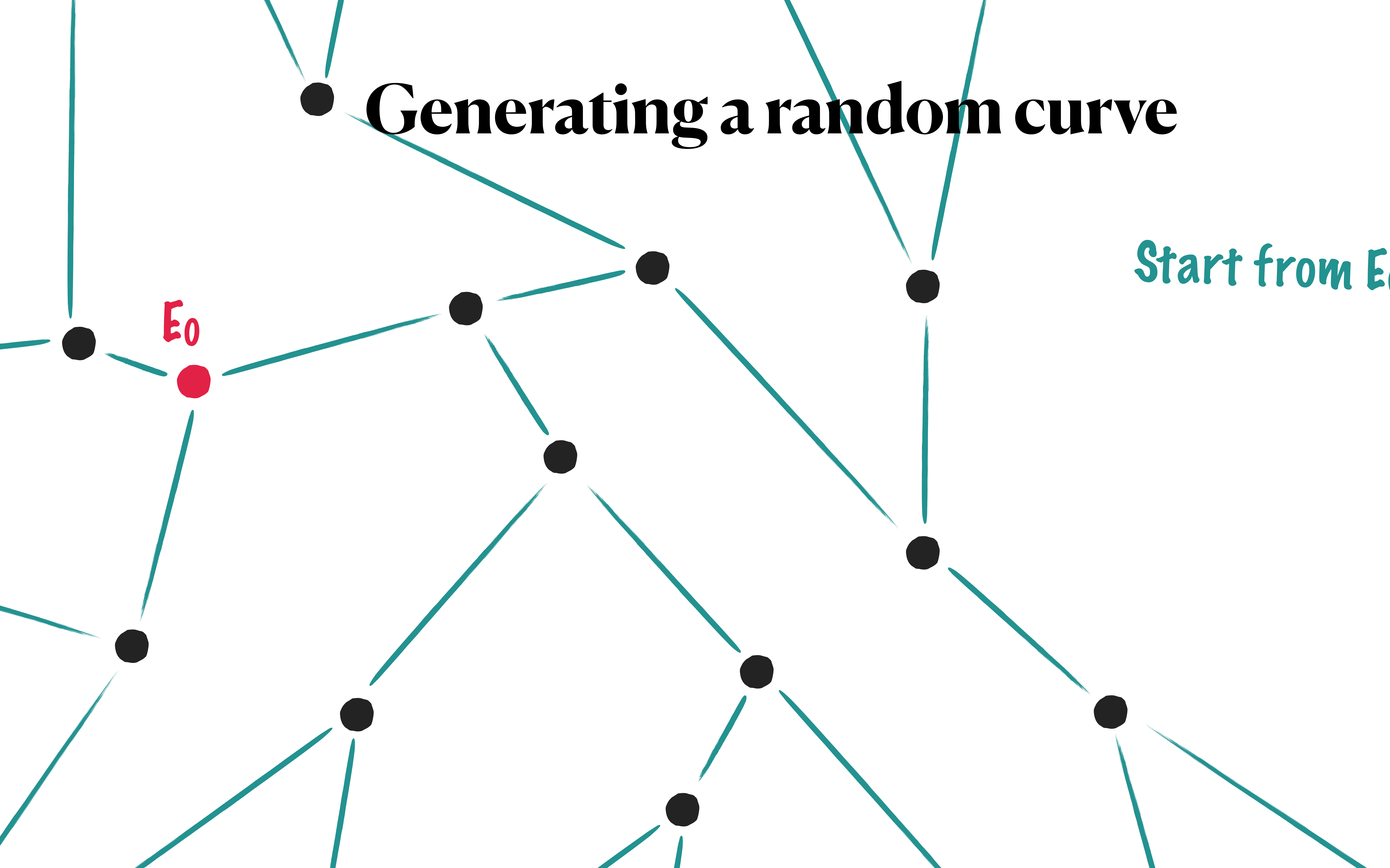
E_0 and $\mathrm{End}(E_0)$ is our reference

- # Generating a random curve



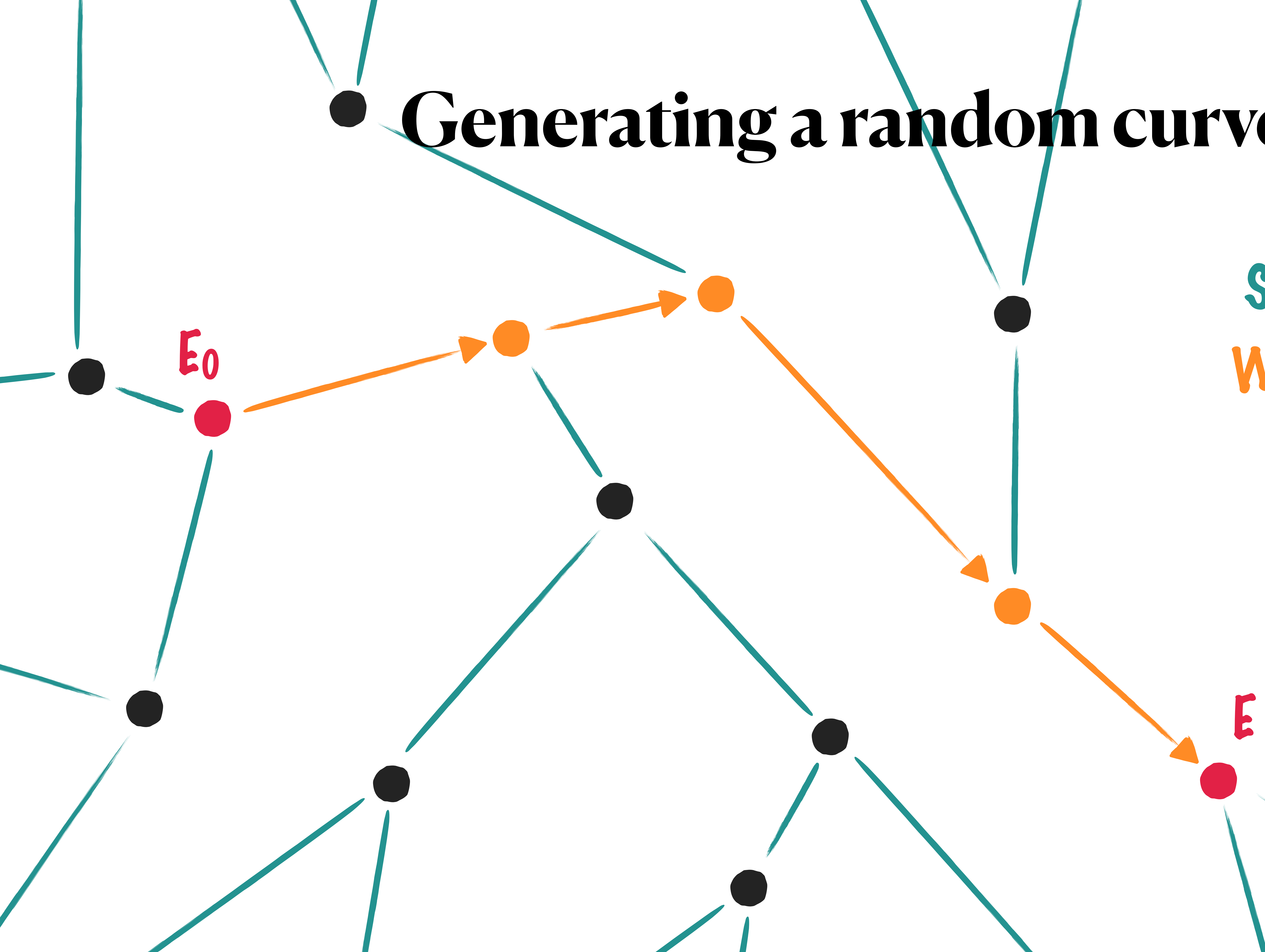
Start from E_0

Generating a random curve



Start from E_0

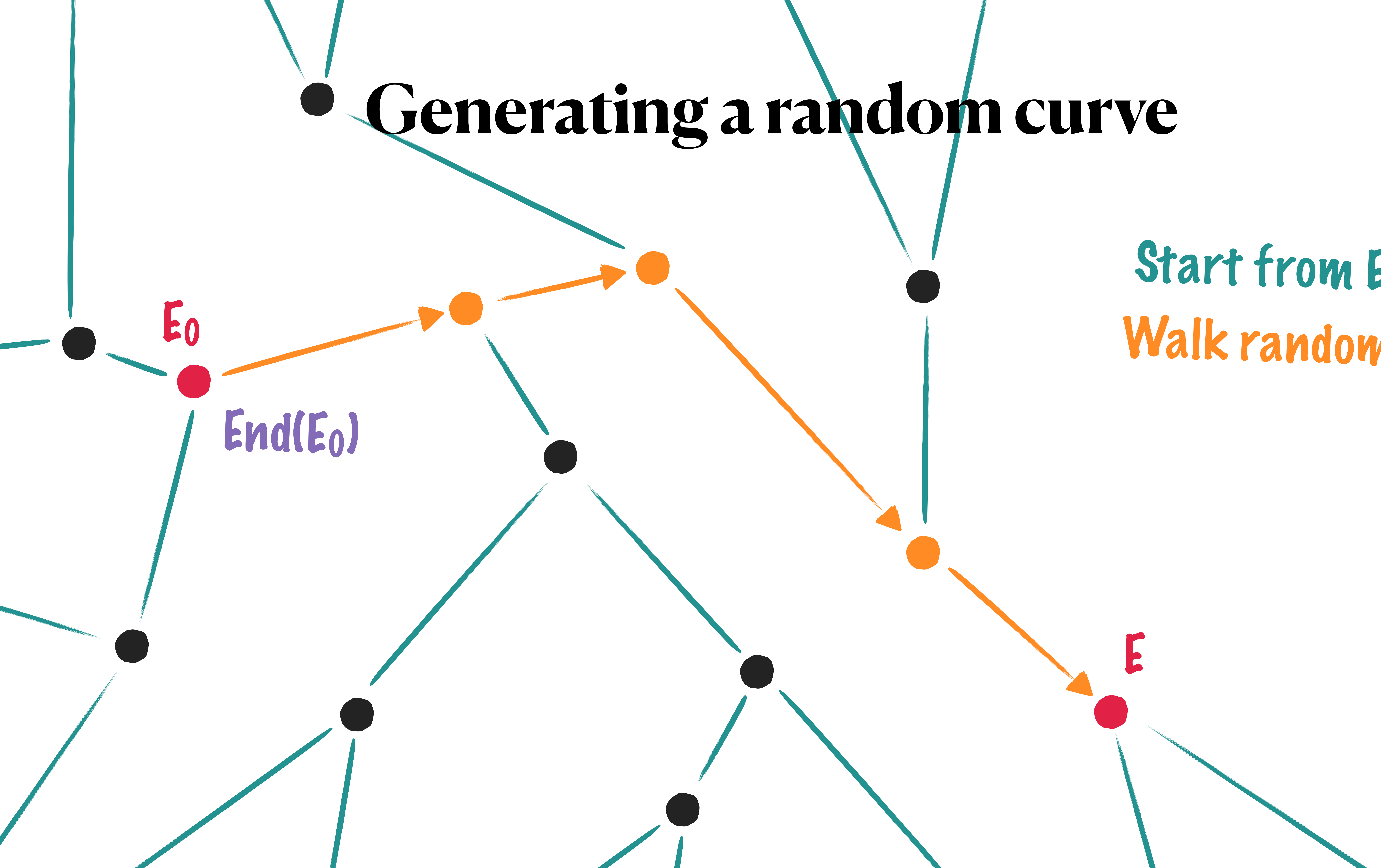
Generating a random curve



Start from E0

Walk randomly

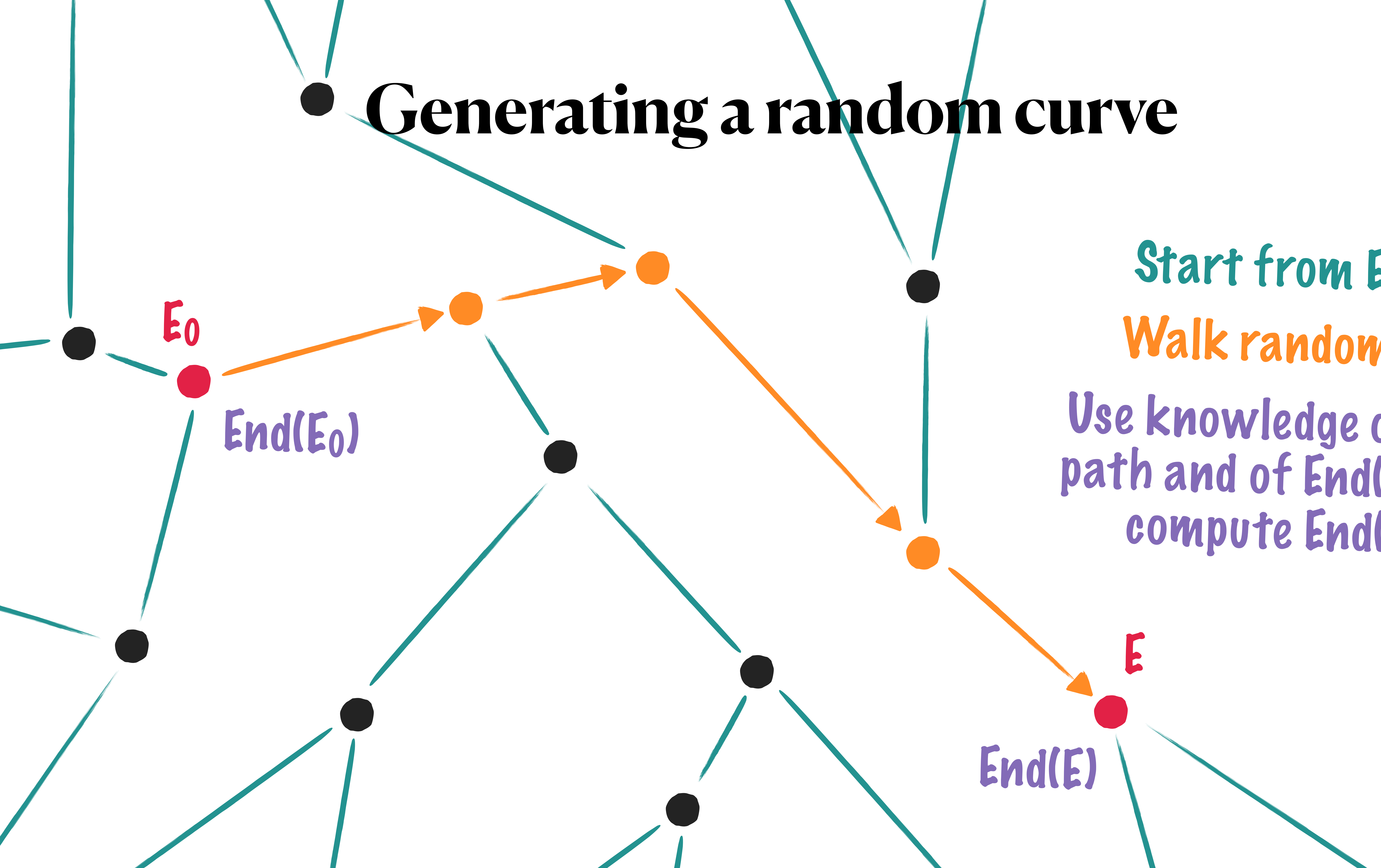
Generating a random curve



Start from E_0

Walk randomly

Generating a random curve



Start from E_0

Walk randomly

Use knowledge of the path and of $\mathrm{End}(E_0)$ to compute $\mathrm{End}(E)$

One can generate $(E, \mathrm{End}(E))$ with E uniform

One can generate $(E, \mathrm{End}(E))$ with E uniform

(Trapdoor generation of uniform **EndRing** instances)

SQIsign

Proving knowledge of an endomorphism

Picture by Beppe Rijs

A sigma protocol following [DKLPW20]

- Generate a random pair (E_{pk}, $\mathrm{End}(E_{\mathrm{pk}})$)
- **public key** = E_{pk}, **secret key** = $\mathrm{End}(E_{\mathrm{pk}})$

A sigma protocol following [DKLPW20]

- Generate a random pair $(E_{\mathrm{pk}}, \mathrm{End}(E_{\mathrm{pk}}))$
- **public key** = E_{pk}, **secret key** = $\mathrm{End}(E_{\mathrm{pk}})$

Alice (prover)

Bob (verifier)

A sigma protocol following [DKLPW20]

- Generate a random pair (E_{pk}, End(E_{pk}))
- **public key** = E_{pk}, **secret key** = End(E_{pk})

Alice (prover) **Bob** (verifier)

End(E_{pk})

E_{pk}

A sigma protocol following [DKLPW20]

- Generate a random pair (E_{pk}, $\text{End}(E_{\text{pk}})$)
- **public key** = E_{pk}, **secret key** = $\text{End}(E_{\text{pk}})$

$\text{End}(E_{\text{pk}})$

E_{pk}

Alice (prover) | **Bob** (verifier)

Generate random (E_{com}, $\text{End}(E_{\text{com}})$) $\xrightarrow{E_{\text{com}}}$

A sigma protocol following [DKLPW20]

- Generate a random pair (E_{pk}, End(E_{pk}))
- **public key** = E_{pk}, **secret key** = End(E_{pk})

End(E_{pk})

E_{pk}

End(E_{com})

E_{com}

Alice (prover)		**Bob** (verifier)
Generate random (E_{com}, End(E_{com}))	$\xrightarrow{E_{\text{com}}}$

A sigma protocol following [DKLPW20]

- Generate a random pair $(E_{\mathrm{pk}}, \mathrm{End}(E_{\mathrm{pk}}))$
- **public key** = E_{pk}, **secret key** = $\mathrm{End}(E_{\mathrm{pk}})$

$\mathrm{End}(E_{\mathrm{pk}})$

E_{pk}

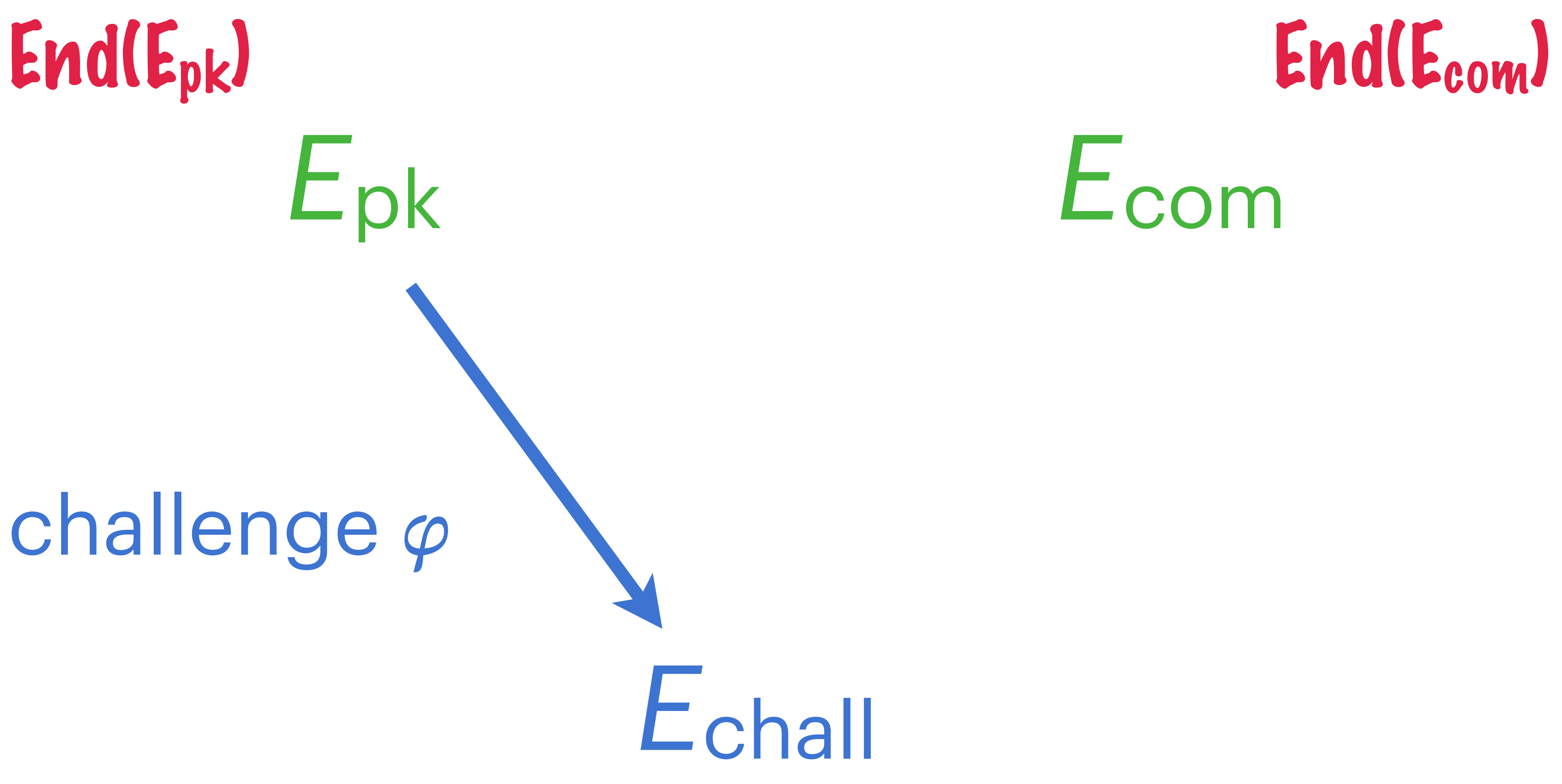
$\mathrm{End}(E_{\mathrm{com}})$

E_{com}

Alice (prover)		**Bob** (verifier)
Generate random $(E_{\mathrm{com}}, \mathrm{End}(E_{\mathrm{com}}))$	$\xrightarrow{E_{\mathrm{com}}}$	
	$\xleftarrow{\varphi}$	Generate random $\varphi : E_{\mathrm{pk}} \to E_{\mathrm{chall}}$

A sigma protocol following [DKLPW20]

- Generate a random pair $(E_{\mathrm{pk}}, \mathrm{End}(E_{\mathrm{pk}}))$
- **public key** = E_{pk}, **secret key** = $\mathrm{End}(E_{\mathrm{pk}})$



Alice (prover)

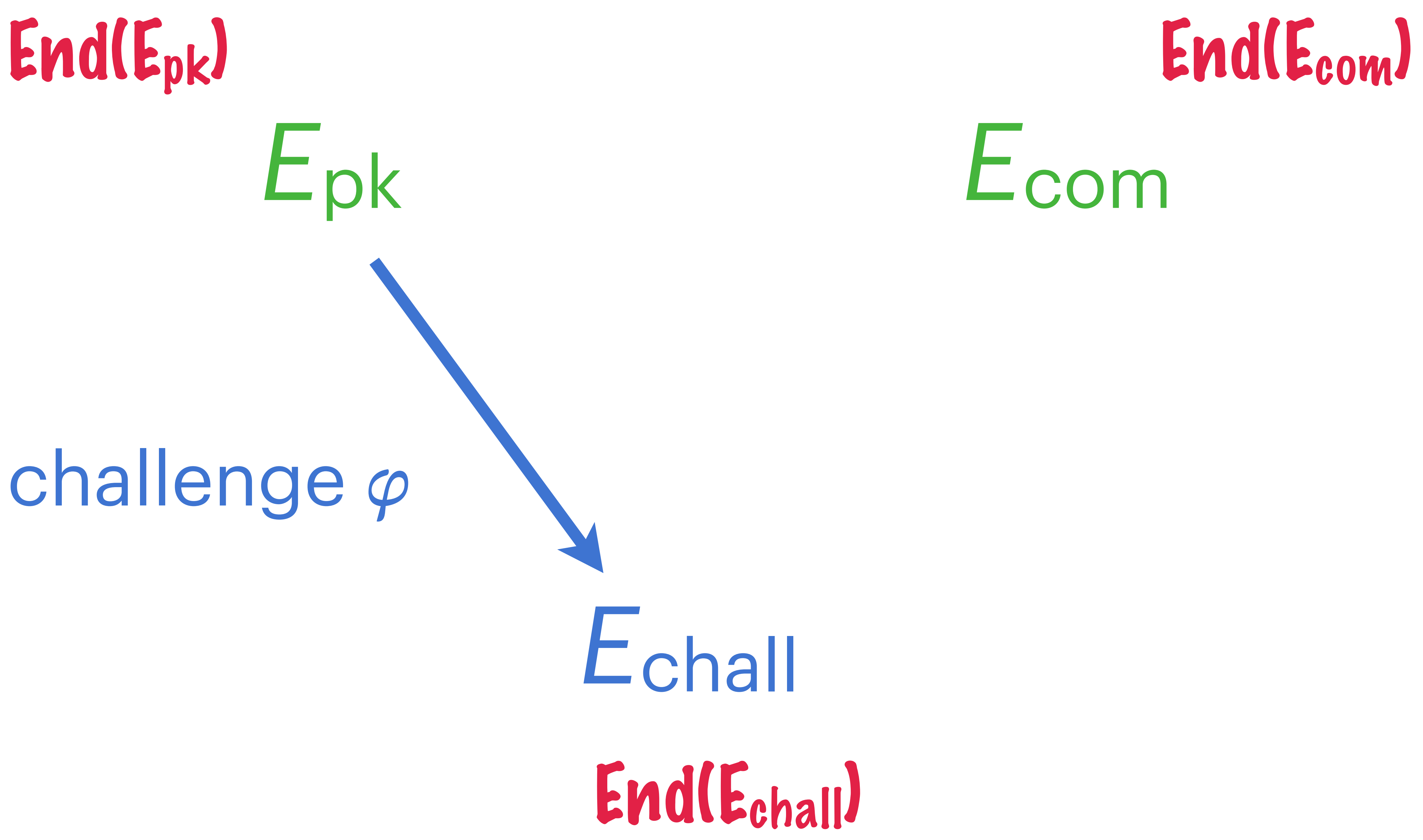
Bob (verifier)

Generate random $(E_{\mathrm{com}}, \mathrm{End}(E_{\mathrm{com}}))$ $\xrightarrow{E_{\mathrm{com}}}$

$\xleftarrow{\varphi}$ Generate random $\varphi : E_{\mathrm{pk}} \to E_{\mathrm{chall}}$

A sigma protocol following [DKLPW20]

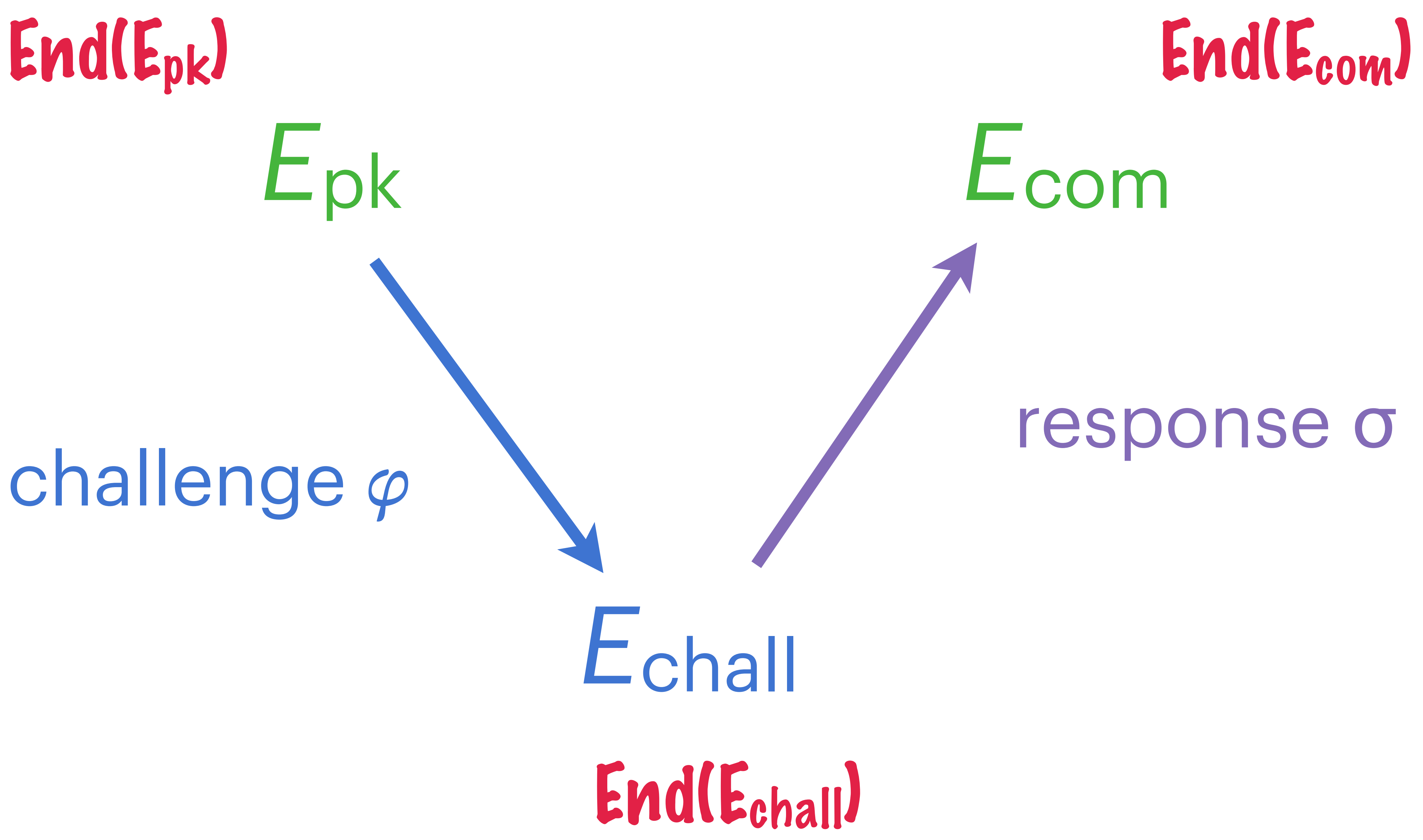
- Generate a random pair (E_{pk}, $\text{End}(E_{\text{pk}})$)
- **public key** = E_{pk}, **secret key** = $\text{End}(E_{\text{pk}})$



Alice (prover)		**Bob** (verifier)
Generate random (E_{com}, $\text{End}(E_{\text{com}})$)	$\xrightarrow{E_{\text{com}}}$	
	$\xleftarrow{\varphi}$	Generate random $\varphi : E_{\text{pk}} \to E_{\text{chall}}$

A sigma protocol following [DKLPW20]

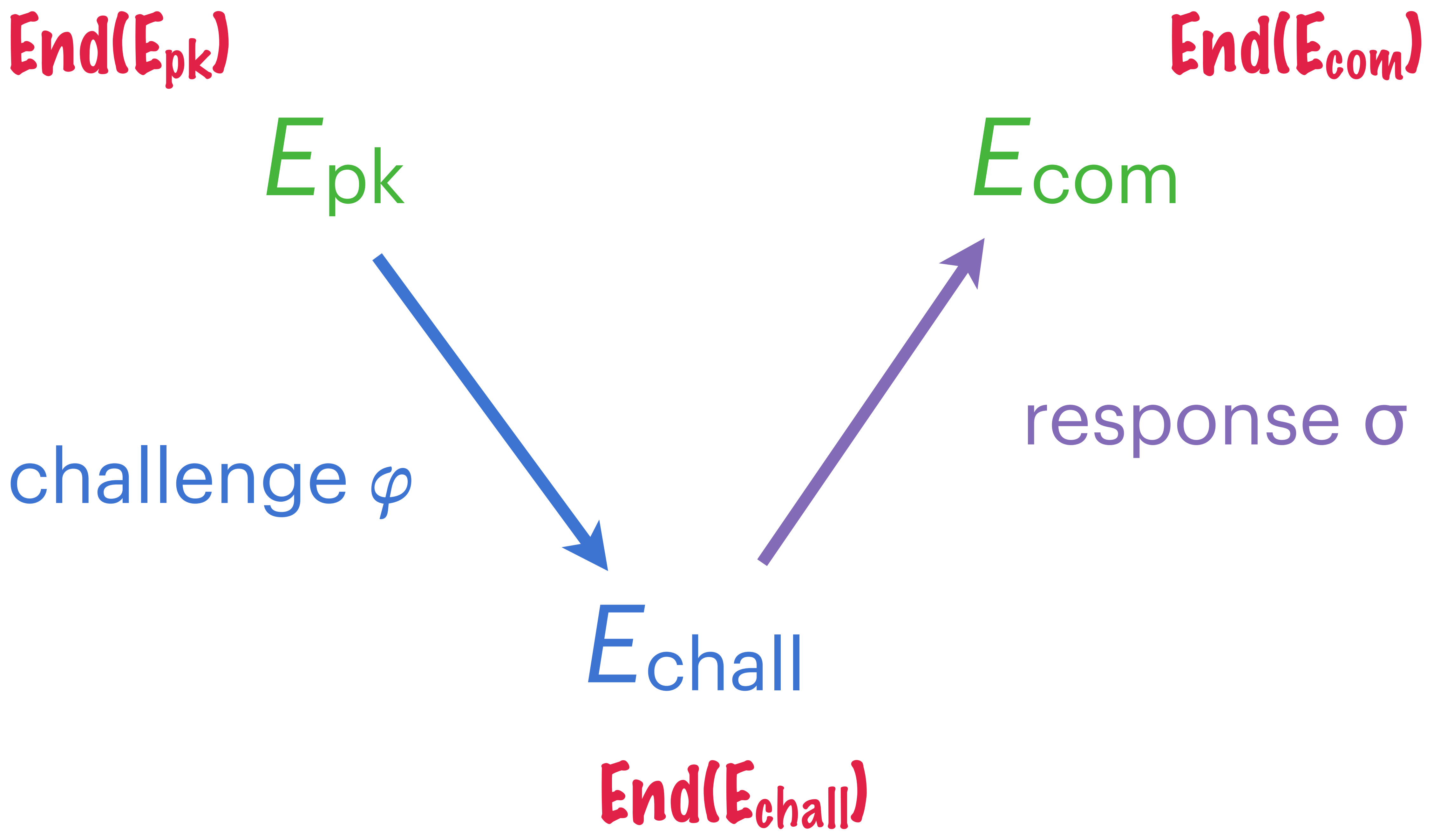
- Generate a random pair $(E_{\mathrm{pk}}, \mathrm{End}(E_{\mathrm{pk}}))$
- **public key** = E_{pk}, **secret key** = $\mathrm{End}(E_{\mathrm{pk}})$



Alice (prover)		**Bob** (verifier)
Generate random $(E_{\mathrm{com}}, \mathrm{End}(E_{\mathrm{com}}))$	$\xrightarrow{E_{\mathrm{com}}}$	
	$\xleftarrow{\varphi}$	Generate random $\varphi : E_{\mathrm{pk}} \to E_{\mathrm{chall}}$

A sigma protocol following [DKLPW20]

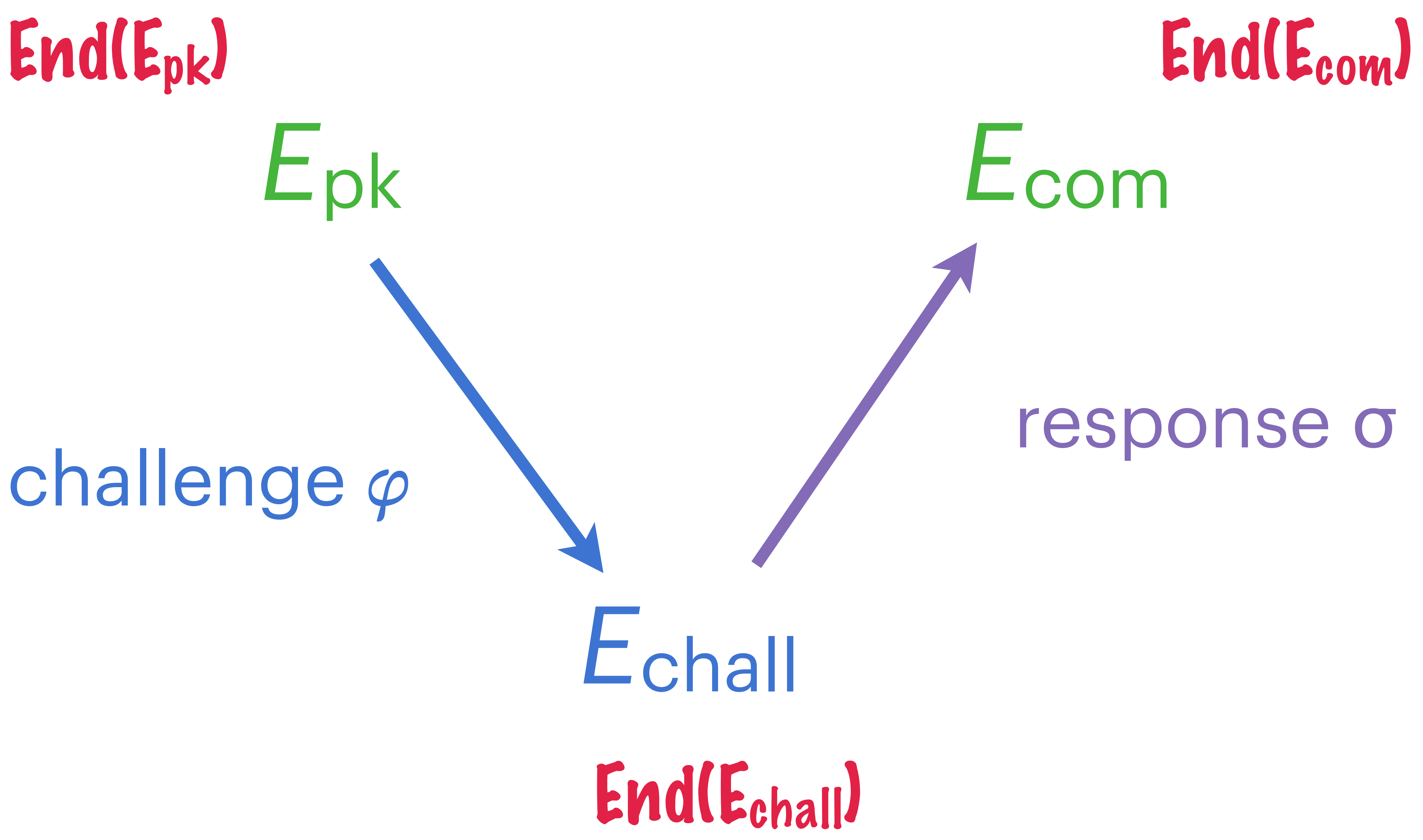
- Generate a random pair (E_{pk}, End(E_{pk}))
- **public key** = E_{pk}, **secret key** = End(E_{pk})



Alice (prover)		**Bob** (verifier)
Generate random (E_{com}, End(E_{com}))	$\xrightarrow{E_{com}}$	
	$\xleftarrow{\varphi}$	Generate random $\varphi : E_{pk} \to E_{chall}$
Compute $\sigma : E_{chall} \to E_{com}$	$\xrightarrow{\sigma}$	

A sigma protocol following [DKLPW20]

- Generate a random pair (E_{pk}, $\text{End}(E_{\text{pk}})$)
- **public key** = E_{pk}, **secret key** = $\text{End}(E_{\text{pk}})$

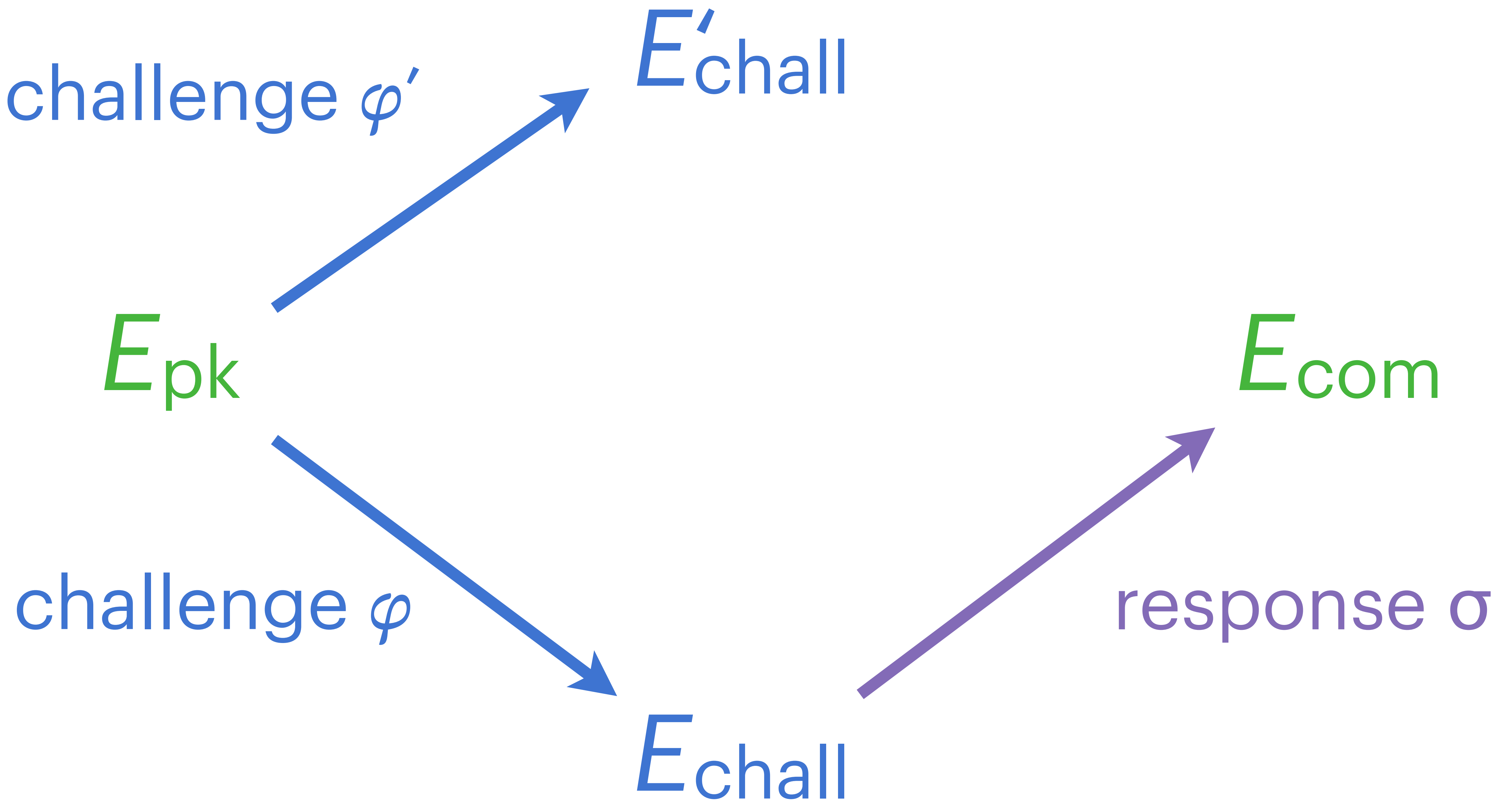


Alice (prover)		**Bob** (verifier)
Generate random (E_{com}, $\text{End}(E_{\text{com}})$)	$\xrightarrow{E_{\text{com}}}$	
	$\xleftarrow{\varphi}$	Generate random $\varphi : E_{\text{pk}} \to E_{\text{chall}}$
Compute $\sigma : E_{\text{chall}} \to E_{\text{com}}$	$\xrightarrow{\sigma}$	Check that σ is an isogeny $E_{\text{chall}} \to E_{\text{com}}$

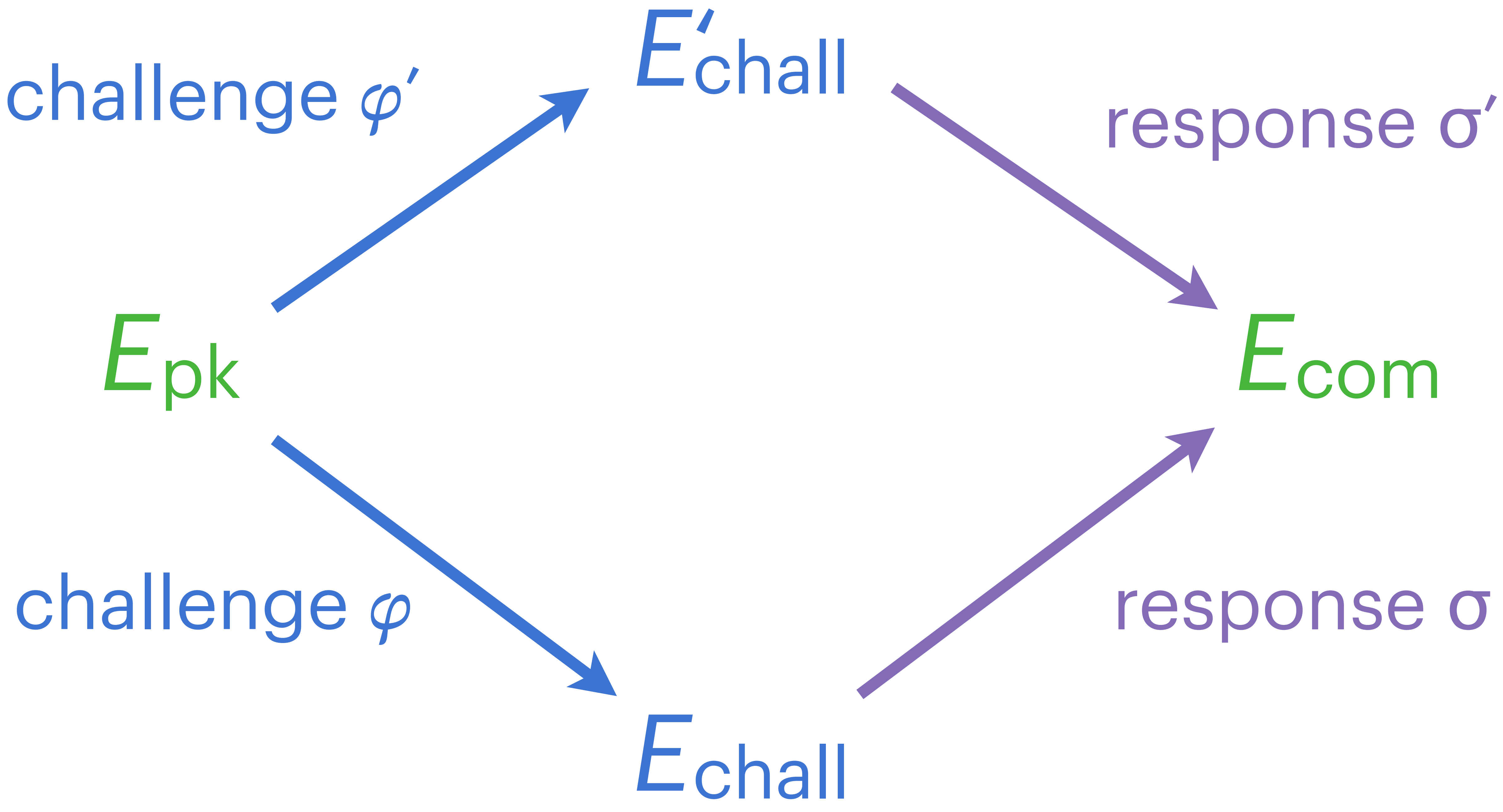
Special soundness

$$E_{\mathrm{pk}} \xrightarrow{\text{challenge } \varphi} E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

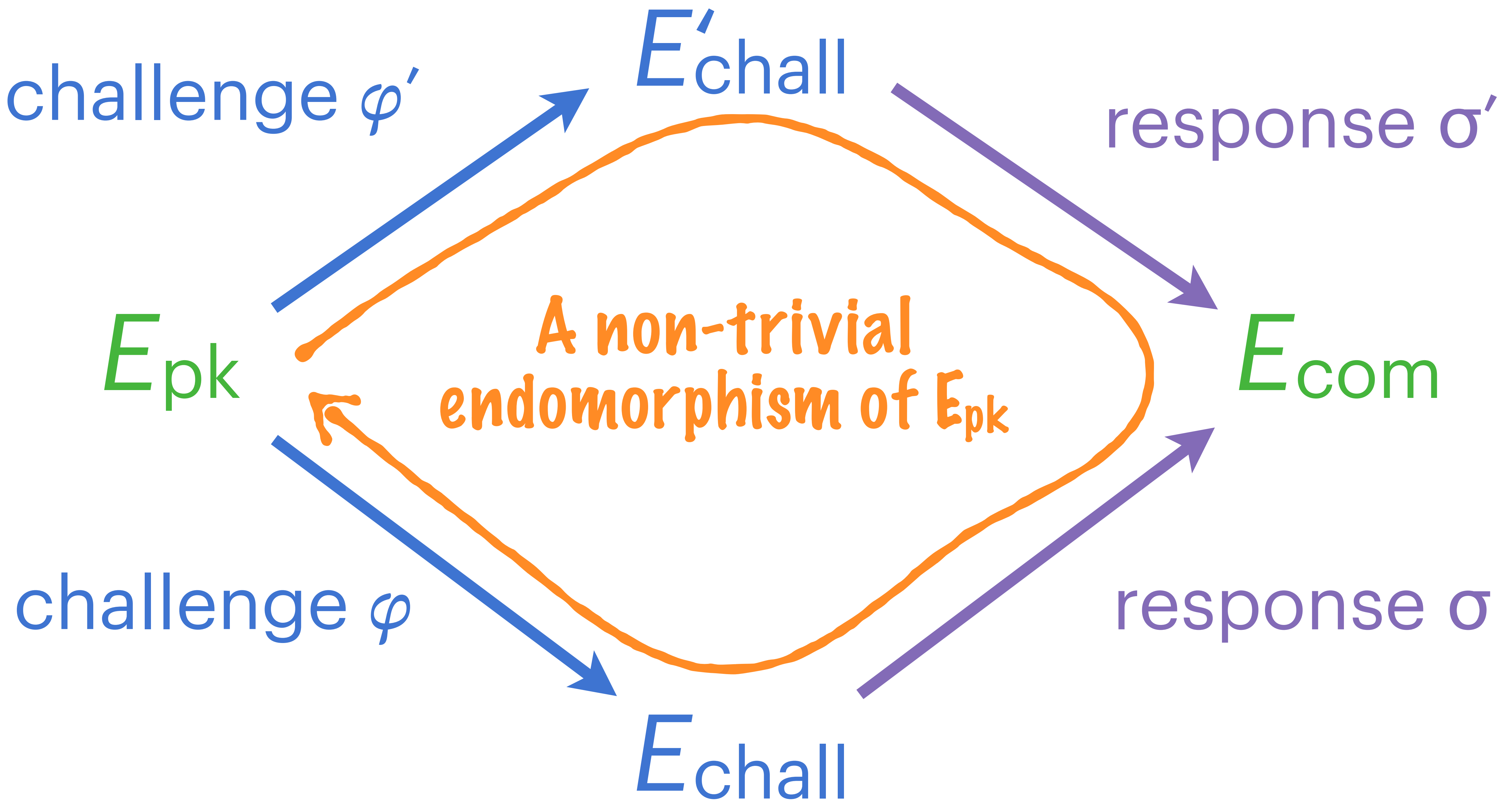
Special soundness



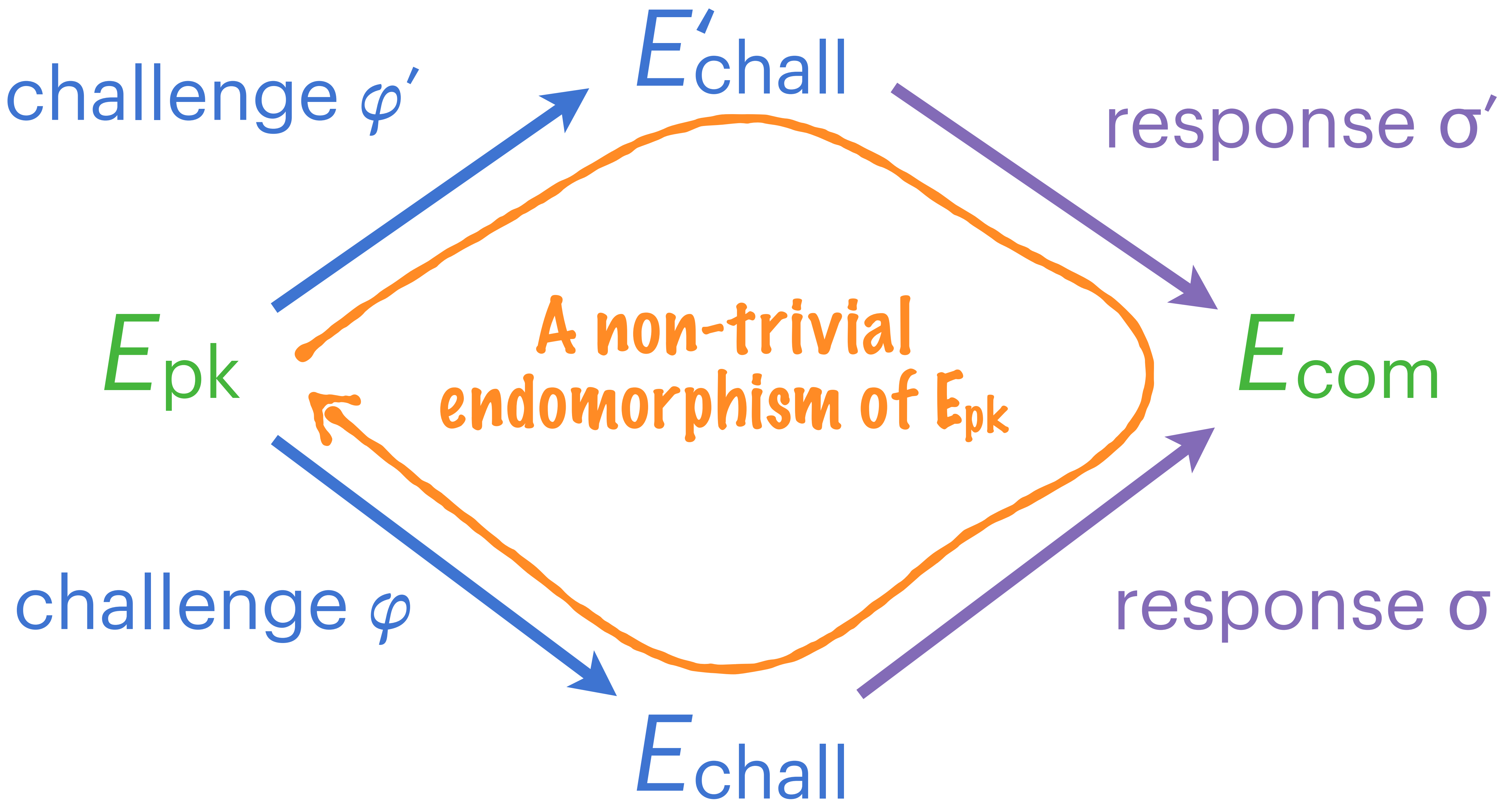
Special soundness



Special soundness

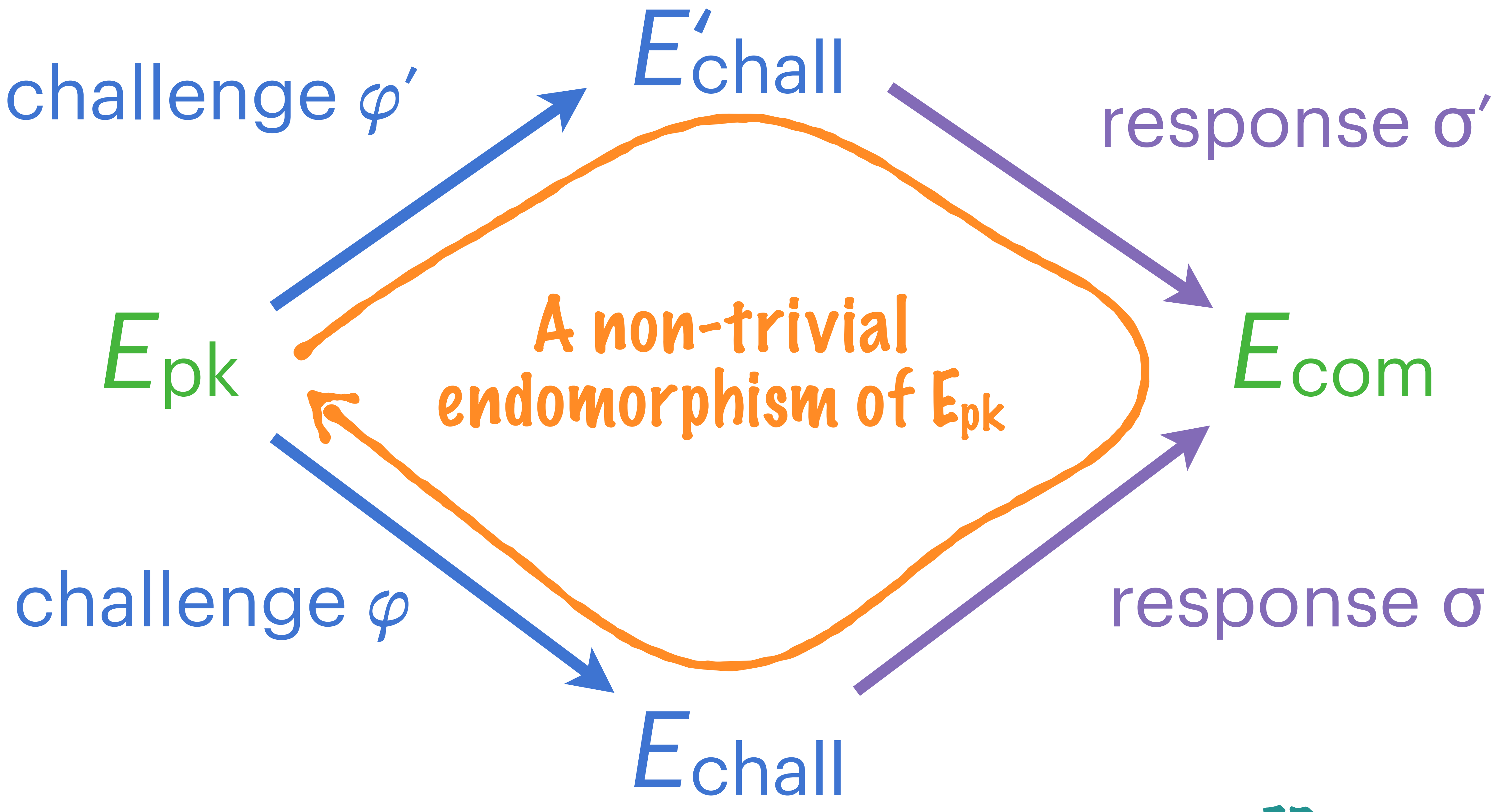


Special soundness



Can respond to 2 challenges ⇒ **Can find an endomorphism**

Special soundness



[Page, W. - Eurocrypt 2024]
Finding one endomorphism ⇔ EndRing

Can respond to 2 challenges ⇒ **Can find an endomorphism**

Computing the response isogeny

$\mathrm{End}(E_{\mathrm{chall}})$ $\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ $\mathrm{End}(E_{\mathrm{com}})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

Computing the response isogeny

$\mathrm{End}(E_{\mathrm{chall}})$ $\qquad\qquad$ $\mathrm{End}(E_{\mathrm{com}})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

1. From $\mathrm{End}(E_{\mathrm{chall}})$ and $\mathrm{End}(E_{\mathrm{com}})$, find a basis $(\varphi_1, \varphi_2, \varphi_3, \varphi_4)$ of $\mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$

Computing the response isogeny

End(E_{chall}) End(E_{com})

$$E_{chall} \xrightarrow{\text{response } \sigma} E_{com}$$

1. From End(E_{chall}) and End(E_{com}), find a basis (φ_1, φ_2, φ_3, φ_4) of Hom(E_{chall}, E_{com})
2. Return some $\sigma \in$ Hom(E_{chall}, E_{com})

Computing the response isogeny

$\mathrm{End}(E_{chall})$ $\qquad\qquad$ $\mathrm{End}(E_{com})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

1. From $\mathrm{End}(E_{\mathrm{chall}})$ and $\mathrm{End}(E_{\mathrm{com}})$, find a basis $(\varphi_1, \varphi_2, \varphi_3, \varphi_4)$ of $\mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$
2. Return some $\sigma \in \mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$

Question: how to choose σ? How to represent it?

Computing the response isogeny

$\mathrm{End}(E_{chall})$ $\mathrm{End}(E_{com})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

1. From $\mathrm{End}(E_{\mathrm{chall}})$ and $\mathrm{End}(E_{\mathrm{com}})$, find a basis $(\varphi_1, \varphi_2, \varphi_3, \varphi_4)$ of $\mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$
2. Return some $\sigma \in \mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$

Question: how to choose σ? How to represent it?

- Need σ fast to evaluate, for efficient verification

Computing the response isogeny

$\mathrm{End}(E_{\mathrm{chall}})$ $\qquad\qquad\qquad\qquad$ $\mathrm{End}(E_{\mathrm{com}})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

1. From $\mathrm{End}(E_{\mathrm{chall}})$ and $\mathrm{End}(E_{\mathrm{com}})$, find a basis $(\varphi_1, \varphi_2, \varphi_3, \varphi_4)$ of $\mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$
2. Return some $\sigma \in \mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$

Question: how to choose σ? How to represent it?

- Need σ fast to evaluate, for efficient verification
- Need σ (and its representation) not to leak the secret

Computing the response isogeny

End(E_{chall}) End(E_{com})

$$E_{chall} \xrightarrow{\text{response } \sigma} E_{com}$$

1. From End(E_{chall}) and End(E_{com}), find a basis (φ_1, φ_2, φ_3, φ_4) of Hom(E_{chall}, E_{com})
2. Return some $\sigma \in$ Hom(E_{chall}, E_{com})

Question: how to choose σ? How to represent it?

- Need σ fast to evaluate, for efficient verification
- Need σ (and its representation) not to leak the secret
  - **Warning:** typical representation of φ_i leaks End(E_{chall})

Computing the response isogeny

$$\underset{\mathrm{End}(E_{\mathrm{chall}})}{E_{\mathrm{chall}}} \xrightarrow{\text{response } \sigma} \underset{\mathrm{End}(E_{\mathrm{com}})}{E_{\mathrm{com}}}$$

Original SQIsign [DKLPW20]**:**

- Solve a norm equation [KLPT14] to find $\sigma \in \mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$ of degree 2^n,
- Convert $\sigma = a_1\varphi_1 + a_2\varphi_2 + a_3\varphi_3 + a_4\varphi_4$ to path of 2-isogenies

Computing the response isogeny

$\mathrm{End}(E_{\mathrm{chall}})$ $\qquad$ $\mathrm{End}(E_{\mathrm{com}})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

Original SQIsign [DKLPW20]:

- Solve a norm equation [KLPT14] to find $\sigma \in \mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$ of degree 2^n,
- Convert $\sigma = a_1\varphi_1 + a_2\varphi_2 + a_3\varphi_3 + a_4\varphi_4$ to path of 2-isogenies

Problems:

- [KLPT14] finds **big** solution: $\deg(\sigma) = 2^n \approx p^{3.75}$
- Conversion to chain of 2-isogenies is very **costly**: signing takes billions of cycles
- Distribution of [KLPT14] output is mysterious: **not simulatable**? not zero-knowledge?

SQIsignHD

Sqiing in higher dimensions

Picture by Beppe Rijs

Computing the response isogeny

$\mathrm{End}(E_{chall})$ $\qquad\qquad\qquad\qquad$ $\mathrm{End}(E_{com})$

$$E_{chall} \xrightarrow{\text{response } \sigma} E_{com}$$

Original SQIsign [DKLPW20]**:** [KLPT14] finds big solution $\deg(\sigma) = 2^n \approx p^{3.75}$

Computing the response isogeny

$$\overset{\mathrm{End}(E_{chall})}{E_{chall}} \xrightarrow{\text{response } \sigma} \overset{\mathrm{End}(E_{com})}{E_{com}}$$

Original SQIsign [DKLPW20]**:** [KLPT14] finds big solution $\deg(\sigma) = 2^n \approx p^{3.75}$

- Smallest isogeny in $\mathrm{Hom}(E_{chall}, E_{com})$ has degree $\approx p^{0.5}$

Computing the response isogeny

$\mathrm{End}(E_{chall})$ $\qquad\qquad\qquad\qquad\qquad\qquad$ $\mathrm{End}(E_{com})$

$$E_{\text{chall}} \xrightarrow{\text{response } \sigma} E_{\text{com}}$$

Original SQIsign [DKLPW20]**:** [KLPT14] finds big solution $\deg(\sigma) = 2^n \approx p^{3.75}$

- Smallest isogeny in $\mathrm{Hom}(E_{\text{chall}}, E_{\text{com}})$ has degree $\approx p^{0.5}$

Question: Why not output some small $\sigma \in \mathrm{Hom}(E_{\text{chall}}, E_{\text{com}})$, $\deg(\sigma) \approx p^{0.5}$?

Computing the response isogeny

$\mathrm{End}(E_{\mathrm{chall}})$ $\qquad\qquad\qquad\qquad$ $\mathrm{End}(E_{\mathrm{com}})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

Original SQIsign [DKLPW20]**:** [KLPT14] finds big solution $\deg(\sigma) = 2^n \approx p^{3.75}$

- Smallest isogeny in $\mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$ has degree $\approx p^{0.5}$

Question: Why not output some small $\sigma \in \mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$, $\deg(\sigma) \approx p^{0.5}$?

- Representation as linear combination $a_1\varphi_1 + a_2\varphi_2 + a_3\varphi_3 + a_4\varphi_4$ is dangerous:
  - $(\varphi_1, \varphi_2, \varphi_3, \varphi_4)$ leaks $\mathrm{End}(E_{\mathrm{chall}})$, so **leaks secret key** $\mathrm{End}(E_{\mathrm{pk}})$

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]**:**

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]**:**

- Let $\varphi : E_1 \longrightarrow E_2$ of degree d

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]**:**

- Let $\varphi : E_1 \to E_2$ of degree d
- Let (P, Q) is a basis of $E_1[2^n]$, with $2^{2n} > 4 \cdot \deg(\varphi)$

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]**:**

a subgroup of E_1
of order 2^{2n}

- Let $\varphi : E_1 \to E_2$ of degree d
- Let (P, Q) is a basis of $E_1[2^n]$, with $2^{2n} > 4 \cdot \deg(\varphi)$

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]**:**

a subgroup of E_1
of order 2^{2n}

- Let $\varphi : E_1 \to E_2$ of degree d
- Let (P, Q) is a basis of $E_1[2^n]$, with $2^{2n} > 4 \cdot \deg(\varphi)$
- Given $(d, P, Q, \varphi(P), \varphi(Q))$, one can compute $\varphi(R)$ for any $R \in E_1$ in poly. time

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]**:**

a subgroup of E_1
of order 2^{2n}

- Let $\varphi : E_1 \to E_2$ of degree d
- Let (P, Q) is a basis of $E_1[2^n]$, with $2^{2n} > 4 \cdot \deg(\varphi)$
- Given $(d, P, Q, \varphi(P), \varphi(Q))$, one can compute $\varphi(R)$ for any $R \in E_1$ in poly. time
- **Cost:** evaluating an isogeny of degree 2^{2n} in dimension 2, 4 or 8

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]**:**

a subgroup of E_1
of order 2^{2n}

- Let $\varphi : E_1 \to E_2$ of degree d
- Let (P, Q) is a basis of $E_1[2^n]$, with $2^{2n} > 4 \cdot \deg(\varphi)$
- Given $(d, P, Q, \varphi(P), \varphi(Q))$, one can compute $\varphi(R)$ for any $R \in E_1$ in poly. time
- **Cost:** evaluating an isogeny of degree 2^{2n} in dimension 2, 4 or 8

Interpolation: **Knowing φ on a few points $\Rightarrow$ Knowing φ everywhere**

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]:

- Let $\varphi : E_1 \to E_2$ of degree d
- Let (P, Q) is a basis of $E_1[2^n]$, with $2^{2n} > 4 \cdot \deg(\varphi)$
- Given $(d, P, Q, \varphi(P), \varphi(Q))$, one can compute $\varphi(R)$ for any $R \in E_1$ in poly. time
- **Cost:** evaluating an isogeny of degree 2^{2n} in dimension 2, 4 or 8

a subgroup of E_1 of order 2^{2n}

Interpolation: **Knowing φ on a few points $\Rightarrow$ Knowing φ everywhere**

Corollary: $(d, P, Q, \varphi(P), \varphi(Q))$ is an efficient representation of φ, the "**interpolation representation**", or "**HD representation**"

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]:

- Let $\varphi : E_1 \to E_2$ of degree d
- Let (P, Q) is a basis of $E_1[2^n]$, with $2^{2n} > 4 \cdot \deg(\varphi)$
- Given $(d, P, Q, \varphi(P), \varphi(Q))$, one can compute $\varphi(R)$ for any $R \in E_1$ in poly. time
- **Cost:** evaluating an isogeny of degree 2^{2n} in dimension 2, 4 or 8

Fastest, but requires
$2^{2n} - d = a^2$

Interpolation: **Knowing φ on a few points $\Rightarrow$ Knowing φ everywhere**

Corollary: $(d, P, Q, \varphi(P), \varphi(Q))$ is an efficient representation of φ, the "**interpolation representation**", or "**HD representation**"

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]:

- Let $\varphi : E_1 \to E_2$ of degree d
- Let (P, Q) is a basis of $E_1[2^n]$, with $2^{2n} > 4 \cdot \deg(\varphi)$
- Given $(d, P, Q, \varphi(P), \varphi(Q))$, one can compute $\varphi(R)$ for any $R \in E_1$ in poly. time
- **Cost:** evaluating an isogeny of degree 2^{2n} in dimension 2, 4 or 8

Fastest, but requires $2^{2n} - d = a^2$

Somewhat fast, but requires $2^{2n} - d = a^2 + b^2$

Interpolation: **Knowing** φ **on few points** $\Rightarrow$ **Knowing** φ **everywhere**

Corollary: $(d, P, Q, \varphi(P), \varphi(Q))$ is an efficient representation of φ, the "**interpolation representation**", or "**HD representation**"

HD representation of isogenies

Attacks against SIDH [CD23, MMPPW23, Rob23]:

- Let $\varphi : E_1 \to E_2$ of degree d
- Let (P, Q) is a basis of $E_1[2^n]$, with $2^{2n} > 4 \cdot \deg(\varphi)$
- Given $(d, P, Q, \varphi(P), \varphi(Q))$, one can compute $\varphi(R)$ for any $R \in E_1$ in poly. time
- **Cost:** evaluating an isogeny of degree 2^{2n} in dimension 2, 4 or 8

Fastest, but requires $2^{2n} - d = a^2$

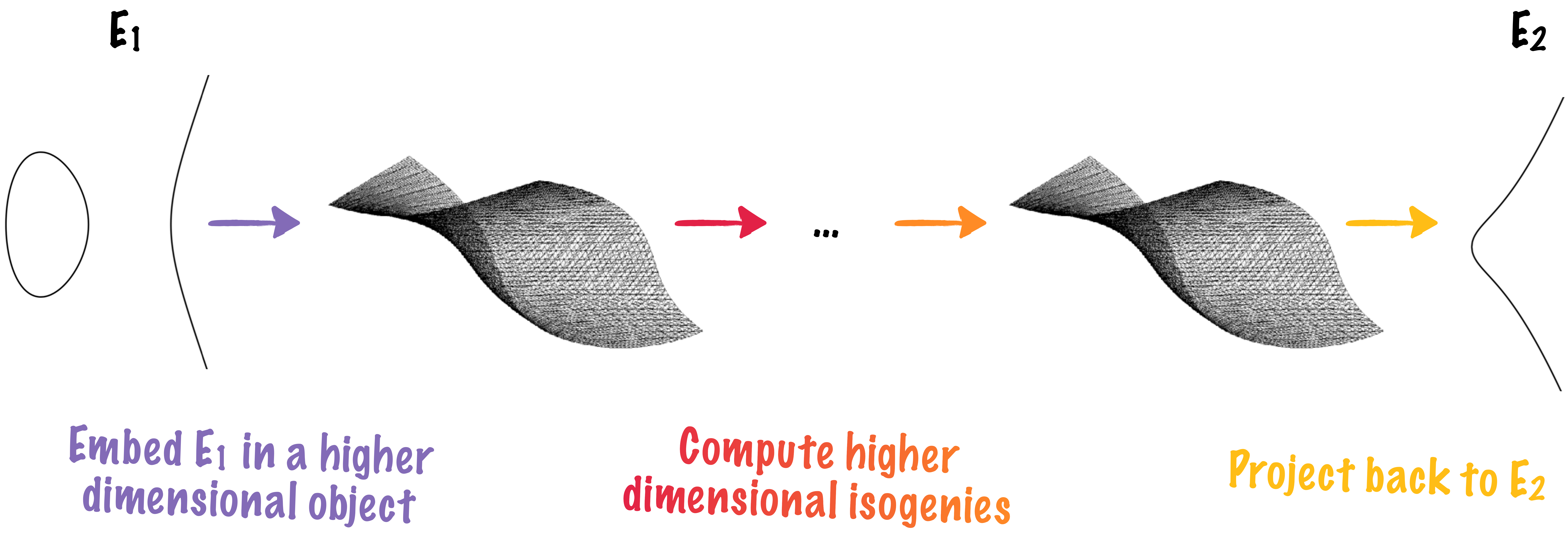
Somewhat fast, but requires $2^{2n} - d = a^2 + b^2$

Very costly, but always works

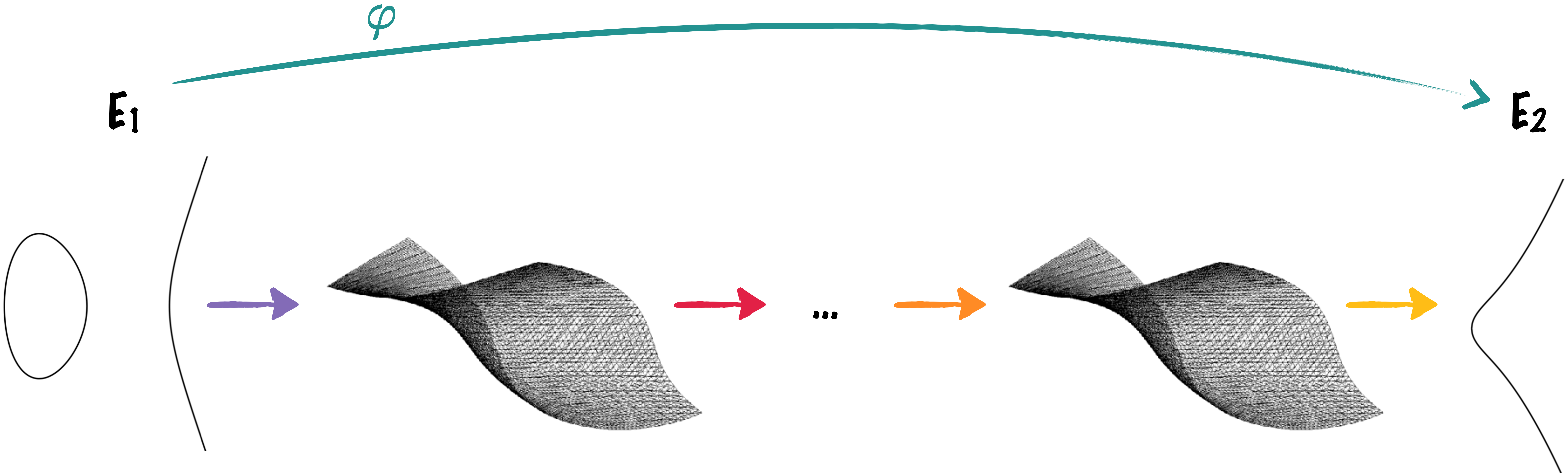
Interpolation: **Knowing φ on a few points $\Rightarrow$ Knowing φ everywhere**

Corollary: $(d, P, Q, \varphi(P), \varphi(Q))$ is an efficient representation of φ, the "**interpolation representation**", or "**HD representation**"

HD representation of isogenies



HD representation of isogenies

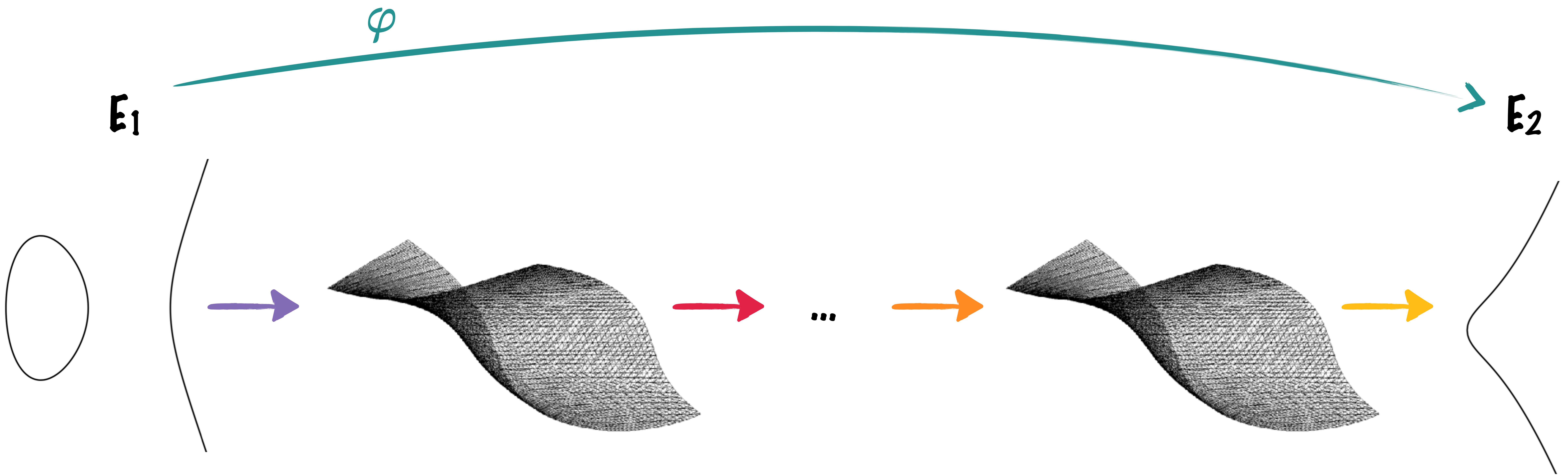


Embed E_1 in a higher dimensional object

Compute higher dimensional isogenies

Project back to E_2

HD representation of isogenies



Embed E_1 in a higher dimensional object

Compute higher dimensional isogenies

Project back to E_2

Determined by what φ does on $E_1[2^n]$

HD response isogeny

$\mathrm{End}(E_{chall})$ $\qquad$ $\mathrm{End}(E_{com})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

SQIsignHD [DLRW24]**:** constructive use of SIDH attacks

HD response isogeny

$$\begin{array}{ccc} \mathrm{End}(E_{chall}) & & \mathrm{End}(E_{com}) \\ E_{chall} & \xrightarrow{\text{response } \sigma} & E_{com} \end{array}$$

SQIsignHD [DLRW24]: constructive use of SIDH attacks

1. Pick random, **small** $\sigma \in \mathrm{Hom}(E_{chall}, E_{com})$ (say, $\deg(\sigma) \approx p^{0.5}$)

HD response isogeny

$\mathrm{End}(E_{chall})$ $\mathrm{End}(E_{com})$

$$E_{chall} \xrightarrow{\text{response } \sigma} E_{com}$$

SQIsignHD [DLRW24]**:** constructive use of SIDH attacks

1. Pick random, **small** $\sigma \in \mathrm{Hom}(E_{chall}, E_{com})$ (say, $\deg(\sigma) \approx p^{0.5}$)
2. Generate basis (P, Q) of $E_{chall}[2^n]$, for $2^{2n} > 4 \cdot \deg(\sigma)$

HD response isogeny

$\mathrm{End}(E_{chall})$ $\qquad\qquad\qquad\qquad$ $\mathrm{End}(E_{com})$

$$E_{chall} \xrightarrow{\text{response } \sigma} E_{com}$$

SQIsignHD [DLRW24]**:** constructive use of SIDH attacks

1. Pick random, **small** $\sigma \in \mathrm{Hom}(E_{chall}, E_{com})$ (say, $\deg(\sigma) \approx p^{0.5}$)
2. Generate basis (P, Q) of $E_{chall}[2^n]$, for $2^{2n} > 4 \cdot \deg(\sigma)$
3. Evaluate $P' = \sigma(P)$ and $Q' = \sigma(Q)$

HD response isogeny

$$\underset{E_{\mathrm{chall}}}{\mathrm{End}(E_{\mathrm{chall}})} \xrightarrow{\text{response } \sigma} \underset{E_{\mathrm{com}}}{\mathrm{End}(E_{\mathrm{com}})}$$

SQIsignHD [DLRW24]: constructive use of SIDH attacks

1. Pick random, **small** $\sigma \in \mathrm{Hom}(E_{\mathrm{chall}}, E_{\mathrm{com}})$ (say, $\deg(\sigma) \approx p^{0.5}$)
2. Generate basis (P, Q) of $E_{\mathrm{chall}}[2^n]$, for $2^{2n} > 4 \cdot \deg(\sigma)$
3. Evaluate $P' = \sigma(P)$ and $Q' = \sigma(Q)$
4. $(\deg(\sigma), P, Q, P', Q')$ is an **HD representation** of σ

Zero-knowledge

From *ad hoc* to rigorous security proof

Picture by Beppe Rijs

Honest verifier zero-knowledge

Alice (prover)		**Bob** (verifier)
Generate random $(E_{\text{com}}, \text{End}(E_{\text{com}}))$	$\xrightarrow{E_{\text{com}}}$	
	$\xleftarrow{\varphi}$	Generate random $\varphi : E_{\text{pk}} \to E_{\text{chall}}$
Compute $\sigma : E_{\text{chall}} \to E_{\text{com}}$	$\xrightarrow{\sigma}$	Check that σ is an isogeny $E_{\text{chall}} \to E_{\text{com}}$

Honest verifier zero-knowledge

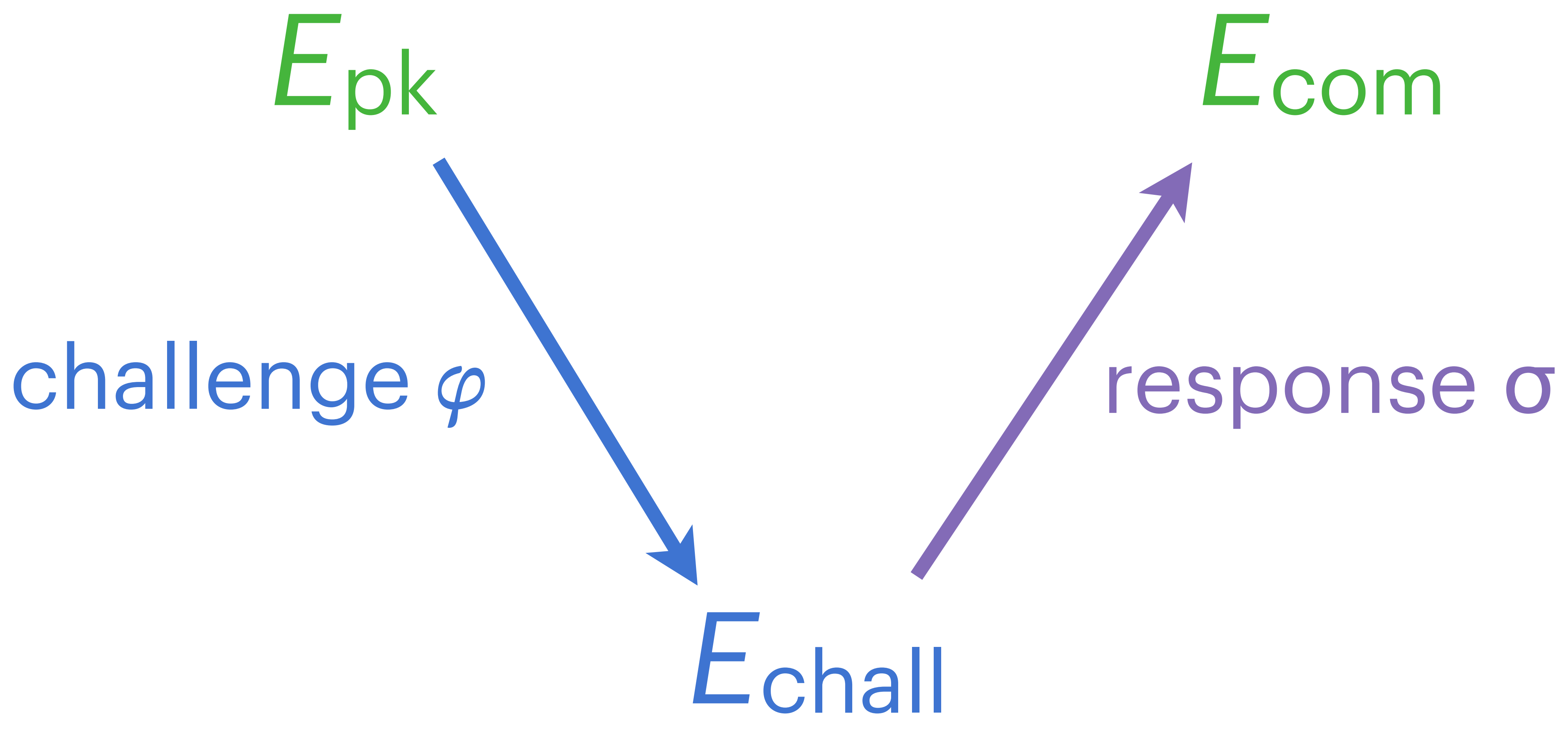
Alice (prover)		**Bob** (verifier)
Generate random $(E_{\text{com}}, \text{End}(E_{\text{com}}))$	$\xrightarrow{E_{\text{com}}}$	
	$\xleftarrow{\varphi}$	Generate random $\varphi : E_{\text{pk}} \to E_{\text{chall}}$
Compute $\sigma : E_{\text{chall}} \to E_{\text{com}}$	$\xrightarrow{\sigma}$	Check that σ is an isogeny $E_{\text{chall}} \to E_{\text{com}}$

transcript

Honest verifier zero-knowledge

Alice (prover)		**Bob** (verifier)
Generate random $(E_{\mathrm{com}}, \mathrm{End}(E_{\mathrm{com}}))$	$\xrightarrow{E_{\mathrm{com}}}$	
	$\xleftarrow{\varphi}$	Generate random $\varphi : E_{\mathrm{pk}} \to E_{\mathrm{chall}}$
Compute $\sigma : E_{\mathrm{chall}} \to E_{\mathrm{com}}$	$\xrightarrow{\sigma}$	Check that σ is an isogeny $E_{\mathrm{chall}} \to E_{\mathrm{com}}$

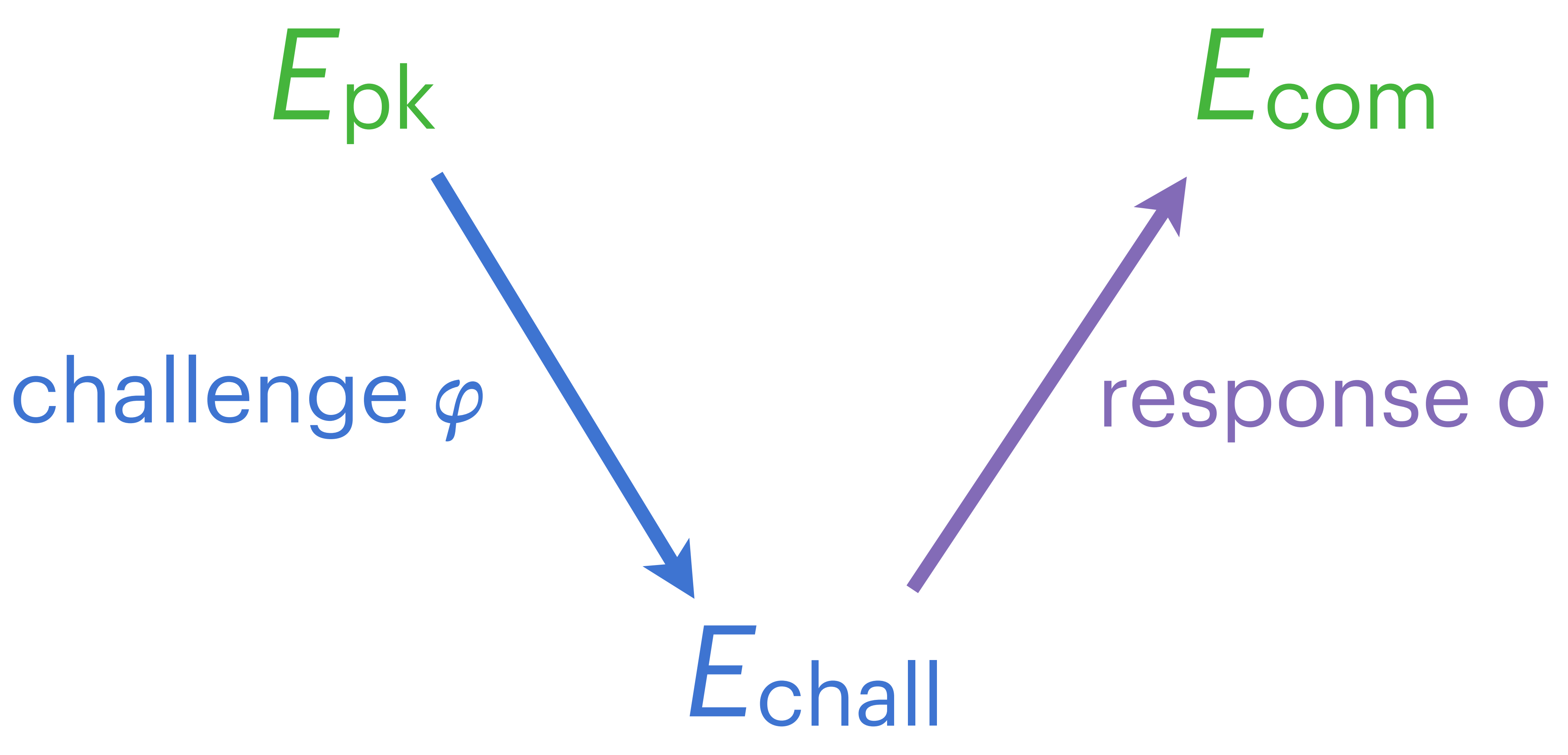
transcript



Honest verifier zero-knowledge

Alice (prover)		**Bob** (verifier)
Generate random $(E_{\mathrm{com}}, \mathrm{End}(E_{\mathrm{com}}))$	$\xrightarrow{E_{\mathrm{com}}}$	
	$\xleftarrow{\varphi}$	Generate random $\varphi : E_{\mathrm{pk}} \to E_{\mathrm{chall}}$
Compute $\sigma : E_{\mathrm{chall}} \to E_{\mathrm{com}}$	$\xrightarrow{\sigma}$	Check that σ is an isogeny $E_{\mathrm{chall}} \to E_{\mathrm{com}}$

transcript



HV Zero-knowledge = **can generate transcripts with same distribution *without the secret key***

Honest transcript

E_{pk}

Honest transcript

1) Uniformly random curve

E_{pk} E_{com}

Honest transcript

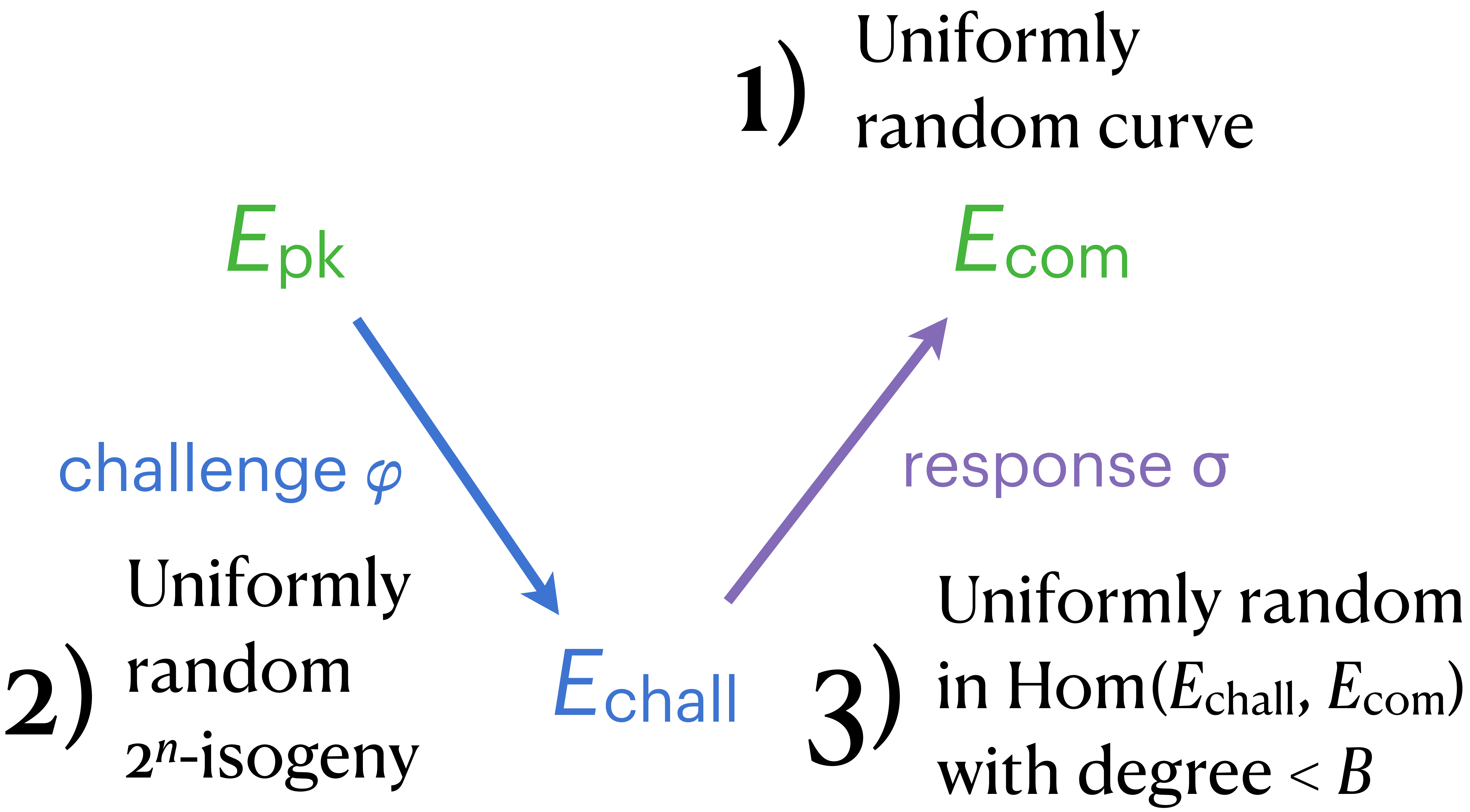
1) Uniformly random curve

E_{pk} E_{com}

$$E_{\text{pk}} \xrightarrow{\text{challenge } \varphi} E_{\text{chall}}$$

2) Uniformly random 2^n-isogeny

Honest transcript



Simulated transcript

E_{pk}

Simulated transcript

$$E_{\mathsf{pk}} \xrightarrow{\text{challenge } \varphi} E_{\mathsf{chall}}$$

1) Uniformly random 2^n-isogeny

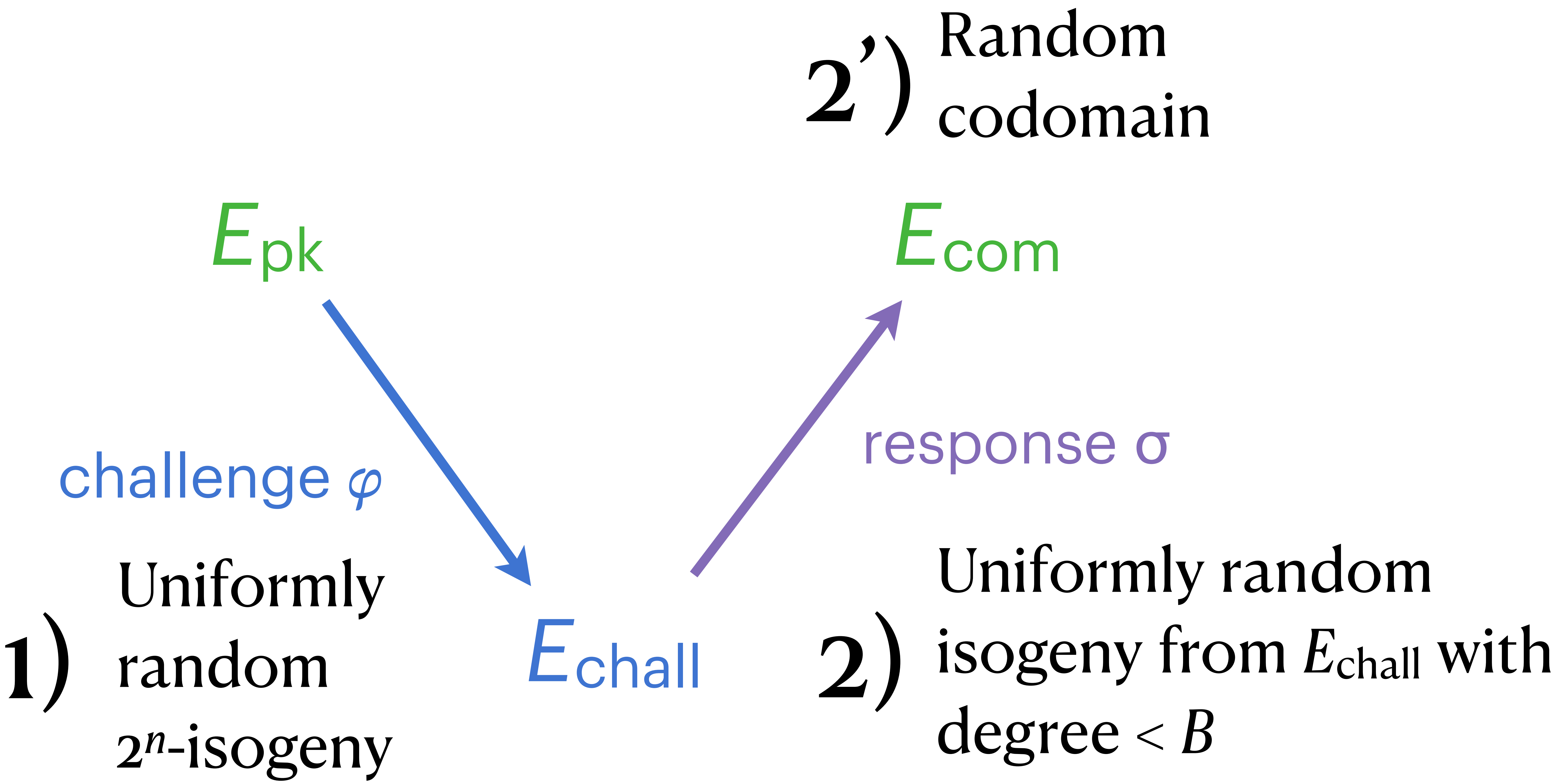
Simulated transcript

$$E_{\mathrm{pk}} \xrightarrow{\text{challenge } \varphi} E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

1) Uniformly random 2^n-isogeny

2) Uniformly random isogeny from E_{chall} with degree $< B$

Simulated transcript



Simulated transcript

2’) Random codomain

E_{pk}

E_{com}

challenge φ

response σ

1) Uniformly random 2^n-isogeny

E_{chall}

2) Uniformly random isogeny from E_{chall} with degree $< B$

With B large enough, statistically indistinguishable from honest!

Simulated transcript

2’) Random codomain

E_{pk} $\xrightarrow{\text{challenge } \varphi}$ E_{chall} $\xrightarrow{\text{response } \sigma}$ E_{com}

1) Uniformly random 2^n-isogeny

???

2) Uniformly random isogeny from E_{chall} with degree $< B$

With B large enough, statistically indistinguishable from honest!

RADIO

Random Any-Degree Isogeny Oracle:

- **Input:** an elliptic curve E, a bound $B > 0$
- **Output:** An efficient representation of uniformly distributed isogeny in

$$\{\varphi : E \to ? \mid \deg(\varphi) < B\}$$

RADIO

Random Any-Degree Isogeny Oracle:

- **Input:** an elliptic curve E, a bound $B > 0$
- **Output:** An efficient representation of uniformly distributed isogeny in

$$\{\varphi : E \rightarrow ? \mid \deg(\varphi) < B\}$$

We know how to sample random isogenies of smooth degree…

RADIO only extends that power to any degree. Powerful oracle?

RADIO

Random Any-Degree Isogeny Oracle:

- **Input:** an elliptic curve E, a bound $B > 0$
- **Output:** An efficient representation of uniformly distributed isogeny in

$$\{\varphi : E \to ? \mid \deg(\varphi) < B\}$$

We know how to sample random isogenies of smooth degree...

RADIO only extends that power to any degree. Powerful oracle?

Assumption: EndRing still hard given a RADIO

Dimensions

2? 4? 8?

Picture by Beppe Rijs

How to verify?

$$\overset{\mathrm{End}(E_{chall})}{E_{chall}} \xrightarrow{\text{response } \sigma} \overset{\mathrm{End}(E_{com})}{E_{com}}$$

Response: σ represented as (d, P, Q, P', Q') where

- (P, Q) is a basis of $E_{chall}[2^n]$, with $2^{2n} > 4 \cdot \deg(\sigma)$
- $(P', Q') = (\sigma(P), \sigma(Q))$

How to verify?

$\mathrm{End}(E_{\text{chall}})$ $\qquad\qquad\qquad\qquad$ $\mathrm{End}(E_{\text{com}})$

$$E_{\text{chall}} \xrightarrow{\text{response } \sigma} E_{\text{com}}$$

Response: σ represented as (d, P, Q, P', Q') where

- (P, Q) is a basis of $E_{\text{chall}}[2^n]$, with $2^{2n} > 4 \cdot \deg(\sigma)$
- $(P', Q') = (\sigma(P), \sigma(Q))$

Verification: Check that (d, P, Q, P', Q') repr. an isogeny $E_{\text{chall}} \longrightarrow E_{\text{com}}$ of deg $< 2^{2n-2}$:

How to verify?

$\mathrm{End}(E_{chall})$ $\qquad\qquad\qquad\qquad$ $\mathrm{End}(E_{com})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

Response: σ represented as (d, P, Q, P', Q') where

- (P, Q) is a basis of $E_{\mathrm{chall}}[2^n]$, with $2^{2n} > 4 \cdot \deg(\sigma)$
- $(P', Q') = (\sigma(P), \sigma(Q))$

Verification: Check that (d, P, Q, P', Q') repr. an isogeny $E_{\mathrm{chall}} \longrightarrow E_{\mathrm{com}}$ of deg $< 2^{2n-2}$:

- In general: evaluate an isogeny in **dimension 8**

How to verify?

$\mathrm{End}(E_{\mathrm{chall}})$ $\qquad\qquad$ $\mathrm{End}(E_{\mathrm{com}})$

$$E_{\mathrm{chall}} \xrightarrow{\text{response } \sigma} E_{\mathrm{com}}$$

Response: σ represented as (d, P, Q, P', Q') where

- (P, Q) is a basis of $E_{\mathrm{chall}}[2^n]$, with $2^{2n} > 4 \cdot \deg(\sigma)$
- $(P', Q') = (\sigma(P), \sigma(Q))$

Verification: Check that (d, P, Q, P', Q') repr. an isogeny $E_{\mathrm{chall}} \longrightarrow E_{\mathrm{com}}$ of deg $< 2^{2n-2}$:

- In general: evaluate an isogeny in **dimension 8**
- If $2^{2n} - \deg(\sigma) = a^2 + b^2$: an isogeny in **dimension 4** is sufficient

How to verify?

$\mathrm{End}(E_{chall})$ $\qquad$ $\mathrm{End}(E_{com})$

$$E_{chall} \xrightarrow{\text{response } \sigma} E_{com}$$

Response: σ represented as (d, P, Q, P', Q') where

- (P, Q) is a basis of $E_{chall}[2^n]$, with $2^{2n} > 4 \cdot \deg(\sigma)$
- $(P', Q') = (\sigma(P), \sigma(Q))$

Verification: Check that (d, P, Q, P', Q') repr. an isogeny $E_{chall} \longrightarrow E_{com}$ of deg $< 2^{2n-2}$:

- In general: evaluate an isogeny in **dimension 8**
- If $2^{2n} - \deg(\sigma) = a^2 + b^2$: an isogeny in **dimension 4** is sufficient

Force $2^{2n} - \deg(\sigma)$ to be a prime $\equiv 1 \bmod 4$

Two versions…

Two versions...

SQIsign8D: no restriction on deg(σ) + degrees large enough for “stat. indisting.”

- **Provably** secure if **EndRing is hard given a RADIO**
- Verification needs isogenies in dimension 8, **impractical**

Two versions…

SQIsign8D: no restriction on deg(σ) + degrees large enough for "stat. indisting."

- **Provably** secure if **EndRing is hard given a RADIO**
- Verification needs isogenies in dimension 8, **impractical**

SQIsign4D: Force 2^n – deg(σ) to be a prime ≡ 1 mod 4 + use smaller degrees

- Security needs **heuristics**, but more compelling, simpler than original SQIsign
- NIST-I level (128 bits security): **64 bytes public key**, **109 bytes signature** (9x smaller than Falcon)
- Verification needs isogenies in dim 4; getting good [Dartois — eprint 2024/1180]
- Scales well to higher security

Dimension 2

SQIsign2D-West

Picture by Beppe Rijs

2D embedding of an isogeny

- Let $\sigma : E_{\mathrm{chall}} \to E_{\mathrm{com}}$ with $d = \deg(\sigma) < 2^{2n-2}$

2D embedding of an isogeny

- Let $\sigma : E_{\text{chall}} \rightarrow E_{\text{com}}$ with $d = \deg(\sigma) < 2^{2n-2}$
- Given an isogeny $\eta : E_{\text{chall}} \rightarrow ?$ of degree $2^{2n} - d$, we can interpolate in dimension 2

2D embedding of an isogeny

- Let $\sigma : E_{\text{chall}} \to E_{\text{com}}$ with $d = \deg(\sigma) < 2^{2n-2}$
- Given an isogeny $\eta : E_{\text{chall}} \to ?$ of degree $2^{2n} - d$, we can interpolate in dimension 2
- If $2^{2n} - d = a^2$ is square, $[a] : E_{\text{chall}} \to E_{\text{chall}}$ is an isogeny of degree $a^2 = 2^{2n} - d$

2D embedding of an isogeny

- Let $\sigma : E_{\text{chall}} \to E_{\text{com}}$ with $d = \deg(\sigma) < 2^{2n-2}$
- Given an isogeny $\eta : E_{\text{chall}} \to ?$ of degree $2^{2n} - d$, we can interpolate in dimension 2
- If $2^{2n} - d = a^2$ is square, $[a] : E_{\text{chall}} \to E_{\text{chall}}$ is an isogeny of degree $a^2 = 2^{2n} - d$

$2^{2n} - d$ is unlikely
to be square ☹️

2D embedding of an isogeny

- Let $\sigma : E_{\text{chall}} \to E_{\text{com}}$ with $d = \deg(\sigma) < 2^{2n-2}$
- Given an isogeny $\eta : E_{\text{chall}} \to ?$ of degree $2^{2n} - d$, we can interpolate in dimension 2
- If $2^{2n} - d = a^2$ is square, $[a] : E_{\text{chall}} \to E_{\text{chall}}$ is an isogeny of degree $a^2 = 2^{2n} - d$

$2^{2n} - d$ is unlikely to be square ☹️

Can look for other isogenies η 🧐

2D embedding of an isogeny

- Let $\sigma : E_{\mathrm{chall}} \to E_{\mathrm{com}}$ with $d = \deg(\sigma) < 2^{2n-2}$
- Given an isogeny $\eta : E_{\mathrm{chall}} \to ?$ of degree $2^{2n} - d$, we can interpolate in dimension 2
- If $2^{2n} - d = a^2$ is square, $[a] : E_{\mathrm{chall}} \to E_{\mathrm{chall}}$ is an isogeny of degree $a^2 = 2^{2n} - d$

$2^{2n} - d$ is unlikely to be square ☹️

Can look for other isogenies η 🧐

Use secret knowledge of $\mathrm{End}(E_{\mathrm{chall}})$...

2D embedding of an isogeny

- Let $\sigma : E_{\text{chall}} \to E_{\text{com}}$ with $d = \deg(\sigma) < 2^{2n-2}$
- Given an isogeny $\eta : E_{\text{chall}} \to ?$ of degree $2^{2n} - d$, we can interpolate in dimension 2
- If $2^{2n} - d = a^2$ is square, $[a] : E_{\text{chall}} \to E_{\text{chall}}$ is an isogeny of degree $a^2 = 2^{2n} - d$

$2^{2n} - d$ is unlikely to be square ☹️

Can look for other isogenies η 🧐

Use secret knowledge of $\text{End}(E_{\text{chall}})$...

Made fast with techniques from QFESTA [Nakagawa, Onuki - 2023] and Clapoti [Page, Robert - 2023]

SQIsign2D-West

[Basso, Dartois, De Feo, Leroux, Maino, Pope, Robert, W. — Asiacrypt 2024]

SQIsign2D-West

[Basso, Dartois, De Feo, Leroux, Maino, Pope, Robert, W. — Asiacrypt 2024]

- Verification: evaluation of an isogeny in dimension 2 (of degree 2^{2n}), much faster than all SQIsign predecessors

SQIsign2D-West

[Basso, Dartois, De Feo, Leroux, Maino, Pope, Robert, W. — Asiacrypt 2024]

- Verification: evaluation of an isogeny in dimension 2 (of degree 2^{2n}), much faster than all SQIsign predecessors
- Still very **compact**: NIST-I level **66 bytes public key**, **148 bytes signature**

SQIsign2D-West

[Basso, Dartois, De Feo, Leroux, Maino, Pope, Robert, W. — Asiacrypt 2024]

- Verification: evaluation of an isogeny in dimension 2 (of degree 2^{2n}), much faster than all SQIsign predecessors
- Still very **compact**: NIST-I level **66 bytes public key**, **148 bytes signature**
- Security proof similar to most conservative predecessor (SQIsign8D)
  - **No heuristics**
  - Computational assumption: ***EndRing*** *is hard given access to an oracle that produces random isogenies of large degree (random-walks-on-steroids oracle)*
  - **Public key is uniformly distributed**: benefit from full force of worst-case to average-case reductions!

Performance

Performance in MCycles, for level of security NIST-I

Caution: non-uniform levels of optimizations... should only be indicative of an order of magnitude... timings in ms are extrapolated for ~3GHz

	Key gen.	Signing	Verif.
Original SQIsign	2800	4600	93
Optimized SQIsign	400	1880 620ms	29 10ms
SQIsignHD	190	115 38ms	?
SQIsign2D-West	60	160 53ms	9 3ms
SQIsign2D-West + heuristics	58	100 33ms	9

For NIST-V security level: cost x6