Isogeny-Based Cryptography

What is an isogeny? A map $\varphi: E_1 \longrightarrow E_2$ given by rational functions such that $\varphi(P+Q) = \varphi(P)+\varphi(Q)$

(looks like polynomial numerator/denominator)

(implies $\varphi(\infty) = \infty$)

elliptic curves

"homomorphism of groups"

Example. $E_1: y^2 = x^3 + 1 \mod 11$

$\downarrow \varphi$

$E_2: y^2 = x^3 + 6 \mod 11$

$(x,y) \in E_1$

$\downarrow$

$\left(\frac{x^3+4}{x^2}, \frac{x^3y+3y}{x^3}\right) \in E_2$

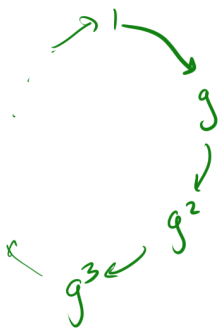
Eg. $(0,1) \longmapsto \left(\frac{4}{0}, \frac{3}{0}\right) = \infty \in E_2$

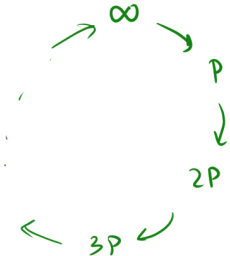
$(1,1) \longmapsto \left(\frac{5}{1}, \frac{4}{1}\right) = (5,4) \in E_2$ on E_1

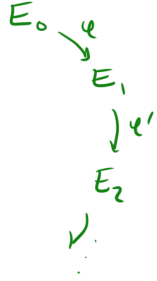
kernel $= \{P \in E_1 : \varphi(P) = \infty\}$

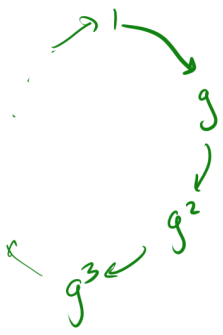
The degree of an isogeny is the # of pts in the kernel

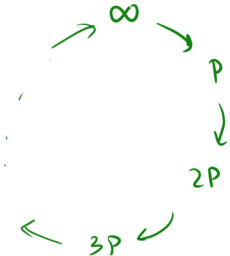
For this ex, $\{\infty, (0,1), (0,10)\} \Rightarrow \deg(\varphi) = 3$. "3-isogeny"

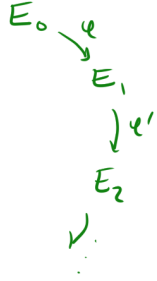

1
g
g^2
g^3


∞
P
2P
3P


E_0
φ
E_1
φ'
E_2

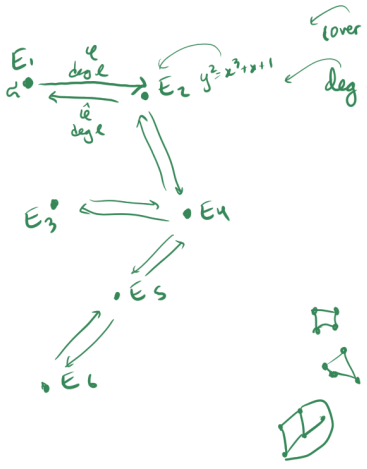

1
g
g^2
g^3


∞
P
2P
3P


E_0
φ
E_1
φ'
E_2

Crucial fact: If $\varphi: E_1 \to E_2$ has degree ℓ

then there exists an isogeny $\hat{\varphi}: E_2 \to E_1$ of degree ℓ.



The Isogeny Graph

Set p = prime (huge), ℓ = small prime (2, 3, 5...)

Vertices: all elliptic curves mod p,

having $p+1$ points.

("j-invariant")

(up to change of coordinates / isomorphism)

Edges: E_1 — E_2

when $\exists$ $\varphi: E_1 \to E_2$ isogeny of degree ℓ.

CSIDH (seaside)

red edges

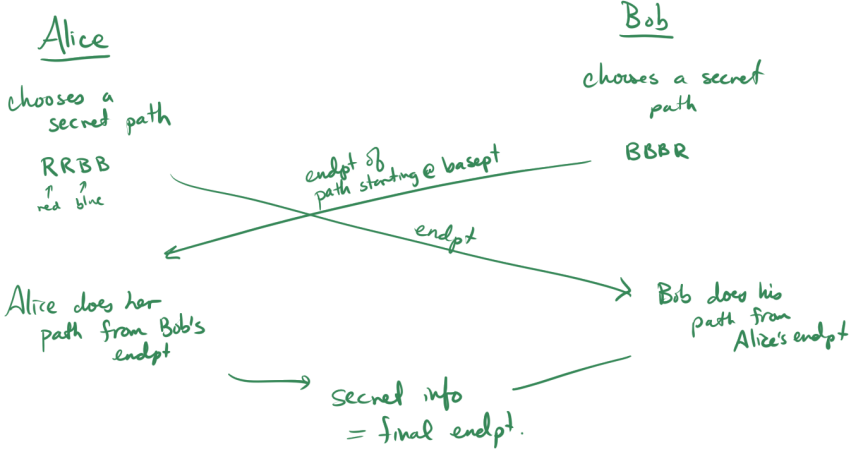
blue edges

- use 2 small primes ℓ_1, ℓ_2
- fix a base point E_0

CSIDH (seaside)

red edges → ℓ_1, blue edges → ℓ_2

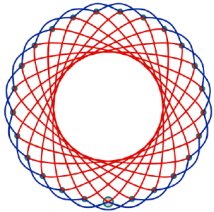
- use 2 small primes ℓ_1, ℓ_2
- fix a basepoint E_0

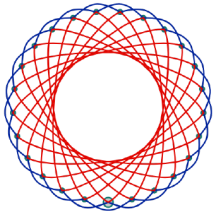


Isogeny-based cryptography: CSIDH

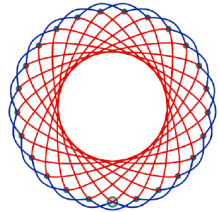
key exchange

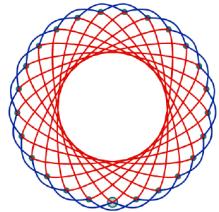
Alice





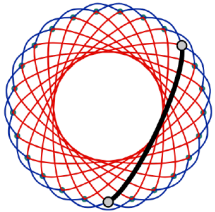
Bob

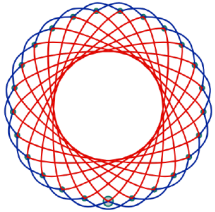




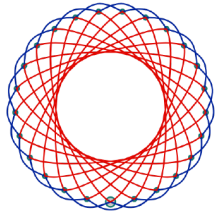
Isogeny-based cryptography: CSIDH

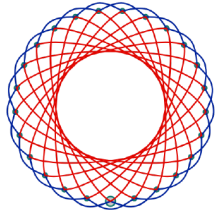
Alice





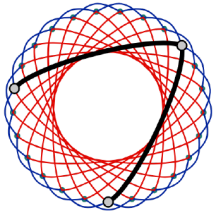
Bob

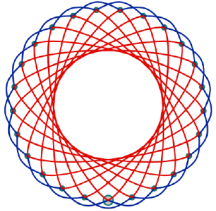




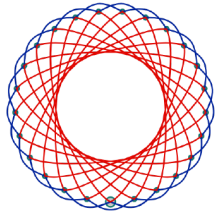
Isogeny-based cryptography: CSIDH

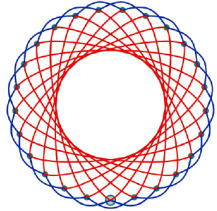
Alice





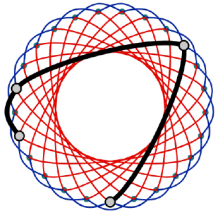
Bob

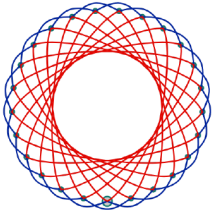




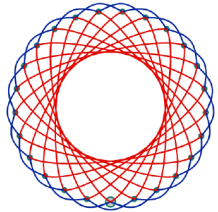
Isogeny-based cryptography: CSIDH

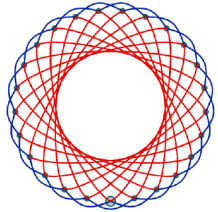
Alice





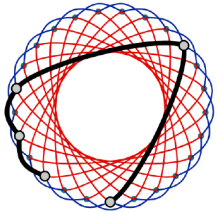
Bob

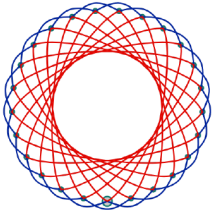




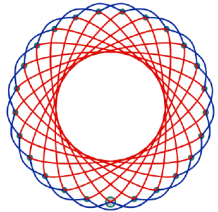
Isogeny-based cryptography: CSIDH

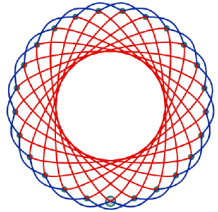
Alice





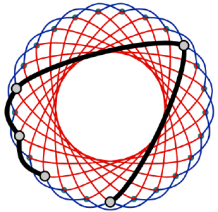
Bob

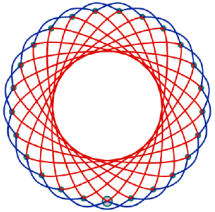




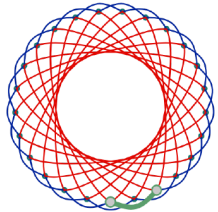
Isogeny-based cryptography: CSIDH

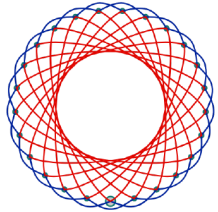
Alice





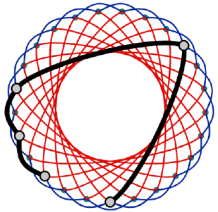
Bob

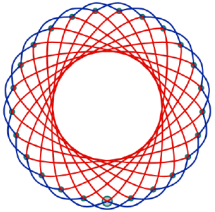




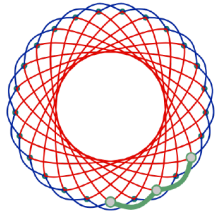
Isogeny-based cryptography: CSIDH

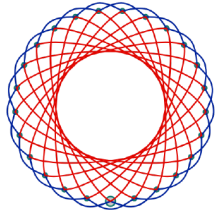
Alice





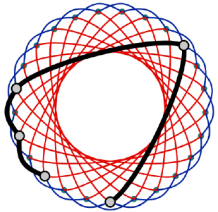
Bob

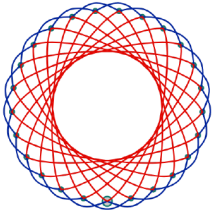




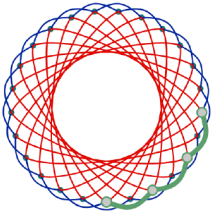
Isogeny-based cryptography: CSIDH

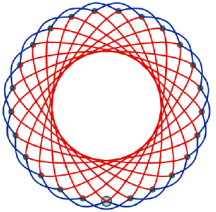
Alice



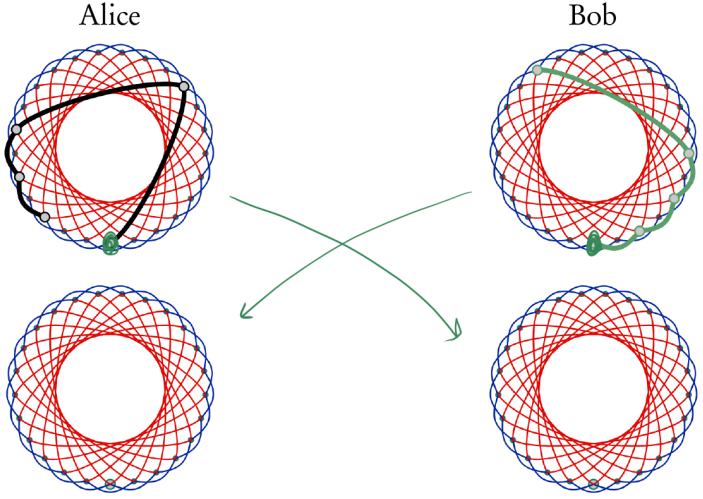


Bob



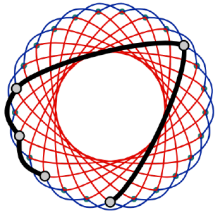


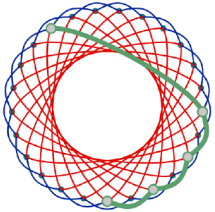
Isogeny-based cryptography: CSIDH



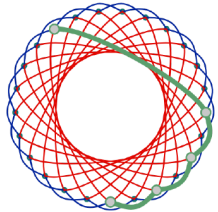
Isogeny-based cryptography: CSIDH

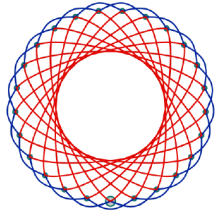
Alice





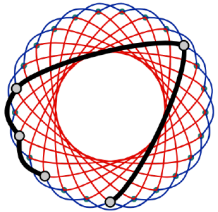
Bob

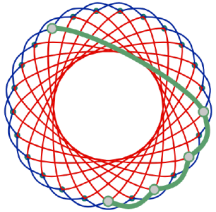




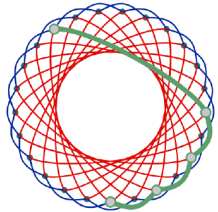
Isogeny-based cryptography: CSIDH

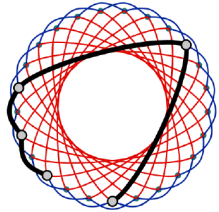
Alice





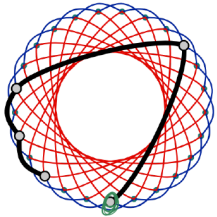
Bob

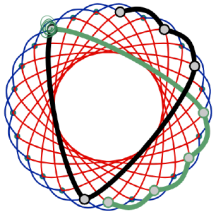




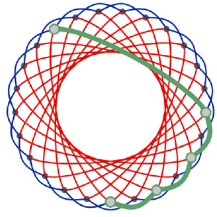
Isogeny-based cryptography: CSIDH

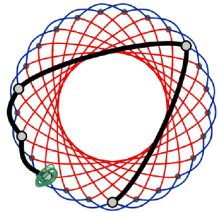
Alice



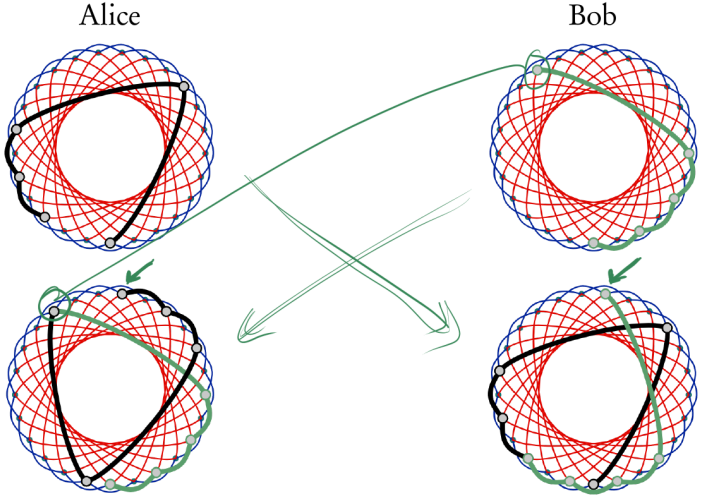


Bob

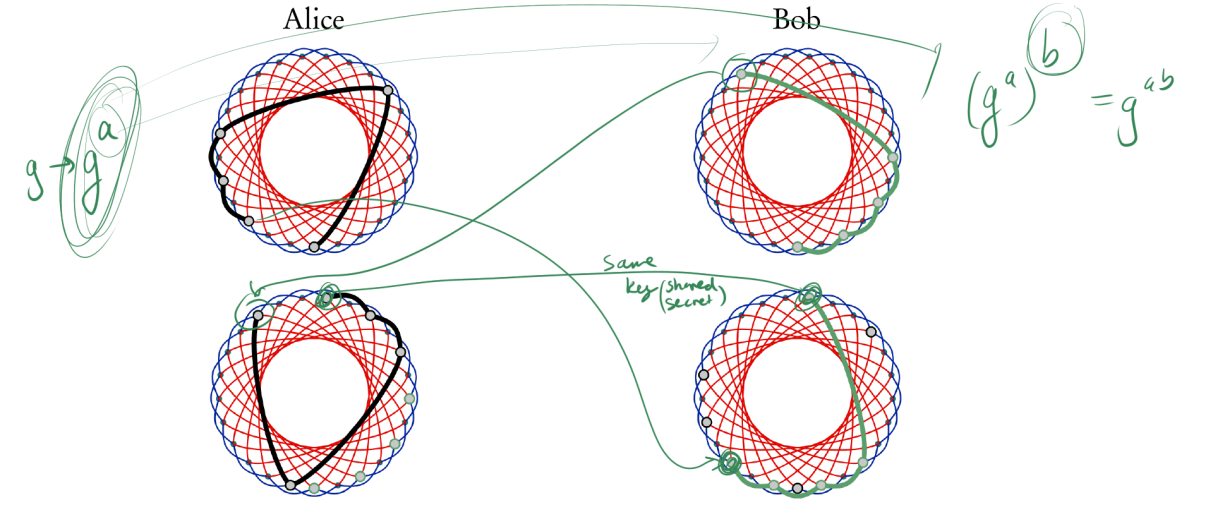




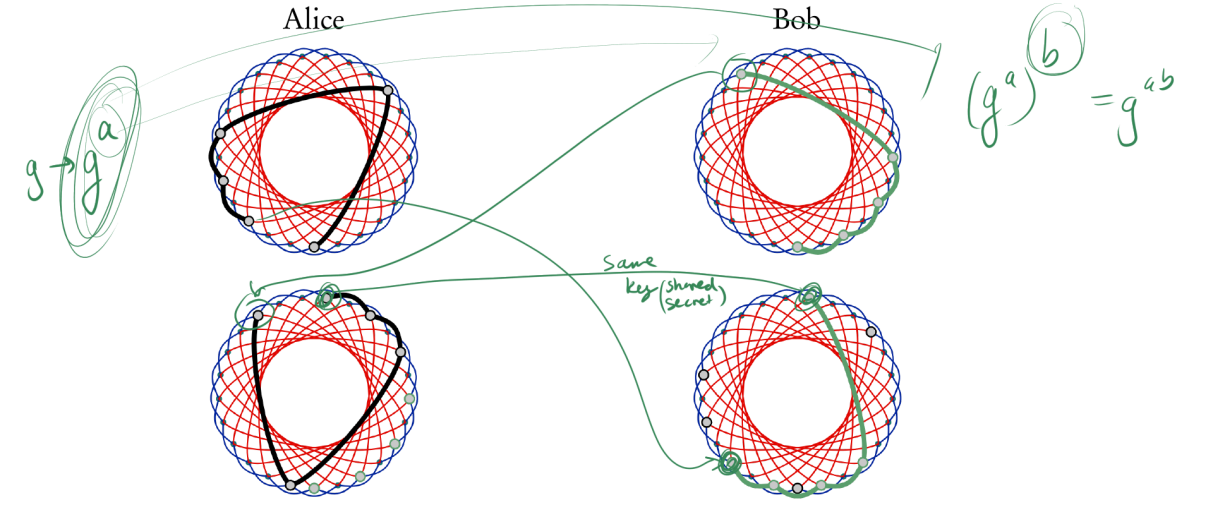
Isogeny-based cryptography: CSIDH



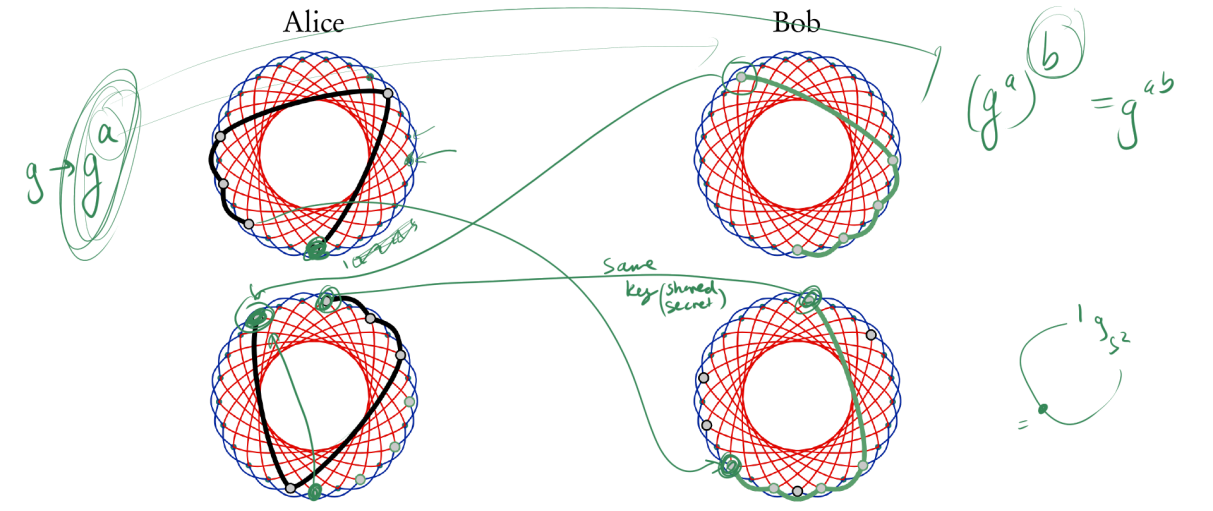
Isogeny-based cryptography: CSIDH



Isogeny-based cryptography: CSIDH



Isogeny-based cryptography: CSIDH



Hard Problem

Given two vertices, construct an isogeny/path between them. "path-finding"