Side channel protections for CSIDH

Luca De Feo

IBM Research Zürich

October 16, 2019, PHISIC, Gardanne

based on joint work with
D. Cervantes-Vázquez, M. Chenu, J.J. Chi-Domínguez, F. Rodríguez-Henríquez, B. Smith

Slides online at `https://defeo.lu/docet`

Why isogenies?

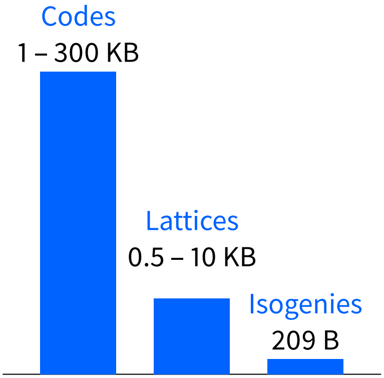
Six families still in NIST post-quantum competition:

Lattices	9 encryption	3 signature
Codes	7 encryption	
Multivariate		4 signature
Isogenies	1 encryption	
Hash-based		1 signature
MPC		1 signature

Why isogenies?

Six families still in NIST post-quantum competition:

Lattices	9 encryption	3 signature
Codes	7 encryption	
Multivariate		4 signature
Isogenies	1 encryption	
Hash-based		1 signature
MPC		1 signature

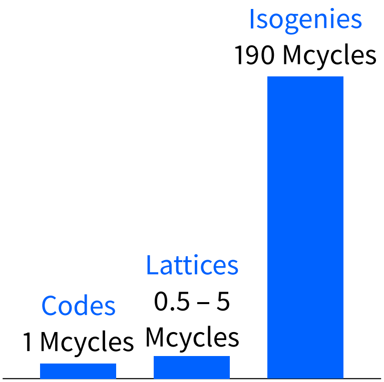


Public key size
NIST-1 level (AES128)
(not to scale)

Why isogenies?

Six families still in NIST post-quantum competition:

Lattices	9 encryption	3 signature
Codes	7 encryption	
Multivariate		4 signature
Isogenies	1 encryption	
Hash-based		1 signature
MPC		1 signature



Encryption performance

NIST-1 level (AES128)

(not to scale)

Iso-what?!

Keywords

- An isogeny is a map between two elliptic curves;

Iso-what?!

Keywords

- An isogeny is a map between two elliptic curves;
- It is a group morphism:
$$\phi(P + Q) = \phi(P) + \phi(Q);$$

Iso-what?!

Keywords

- An isogeny is a map between two elliptic curves;
- It is a group morphism:

$$\phi(P+Q) = \phi(P) + \phi(Q);$$

- It is an algebraic map:

$$\phi(x,y) = \left(\frac{g(x)}{h(x)}, y\left(\frac{g(x)}{h(x)}\right)'\right);$$

Iso-what?!

Keywords

- An isogeny is a map between two elliptic curves;
- It is a group morphism:
$$\phi(P+Q)=\phi(P)+\phi(Q);$$
- It is an algebraic map:
$$\phi(x,y)=\left(\frac{g(x)}{h(x)},\, y\left(\frac{g(x)}{h(x)}\right)'\right);$$
- It is entirely determined by its kernel (i.e., by a single point);

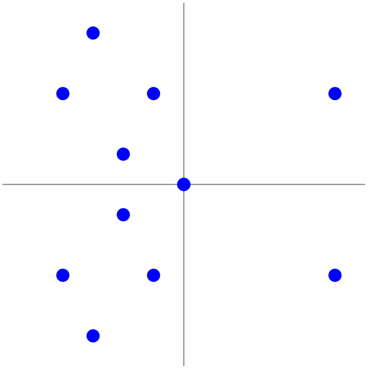
Iso-what?!

Keywords

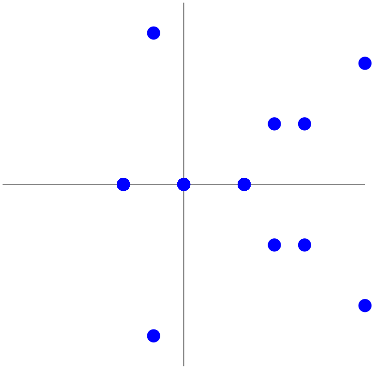
- An isogeny is a map between two elliptic curves;
- It is a group morphism:
$$\phi(P+Q) = \phi(P) + \phi(Q);$$
- It is an algebraic map:
$$\phi(x,y) = \left(\frac{g(x)}{h(x)}, y\left(\frac{g(x)}{h(x)}\right)'\right);$$
- It is entirely determined by its kernel (i.e., by a single point);
- Isogeny degree $=$ size of the kernel $=$ order of kernel generator $\approx$ size of the polynomials;

Isogenies: an example over $\mathbb{F}_{11}$

$$E \;:\; y^2 = x^3 + x$$

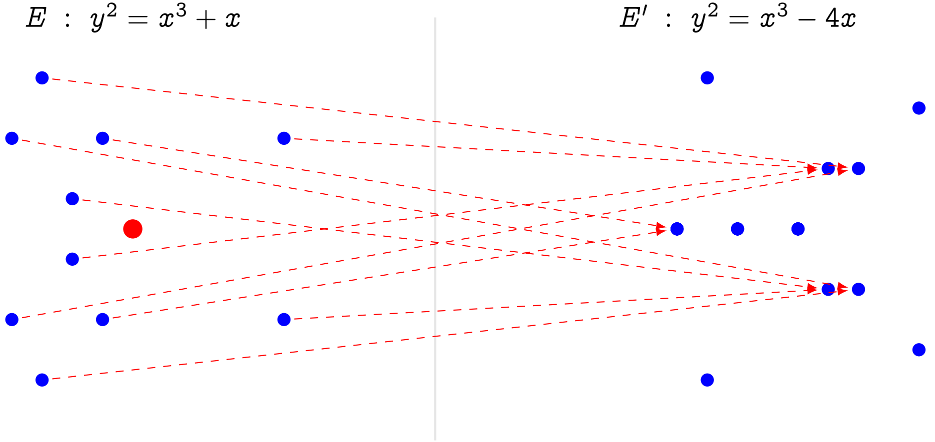


$$E' \;:\; y^2 = x^3 - 4x$$



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

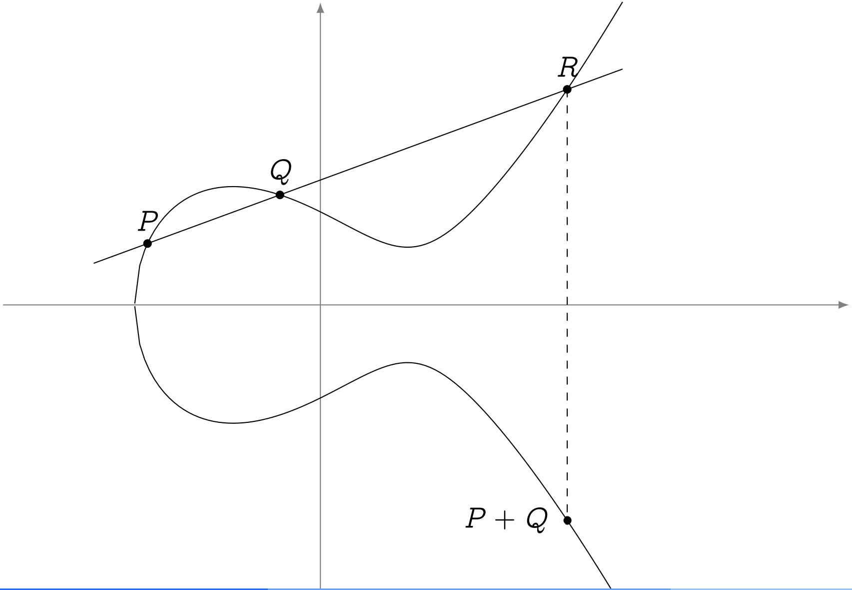
Isogenies: an example over $\mathbb{F}_{11}$



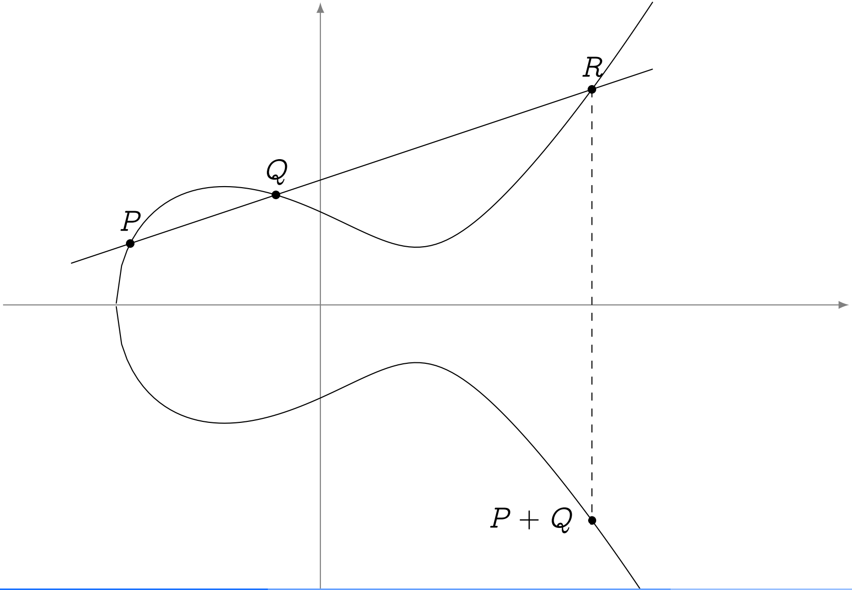
$$\phi(x,y) = \left(\frac{x^2+1}{x},\quad y\frac{x^2-1}{x^2}\right)$$

- Kernel generator in red.
- This is a degree 2 map.
- Analogous to $x \mapsto x^2$ in $\mathbb{F}_q^*$.

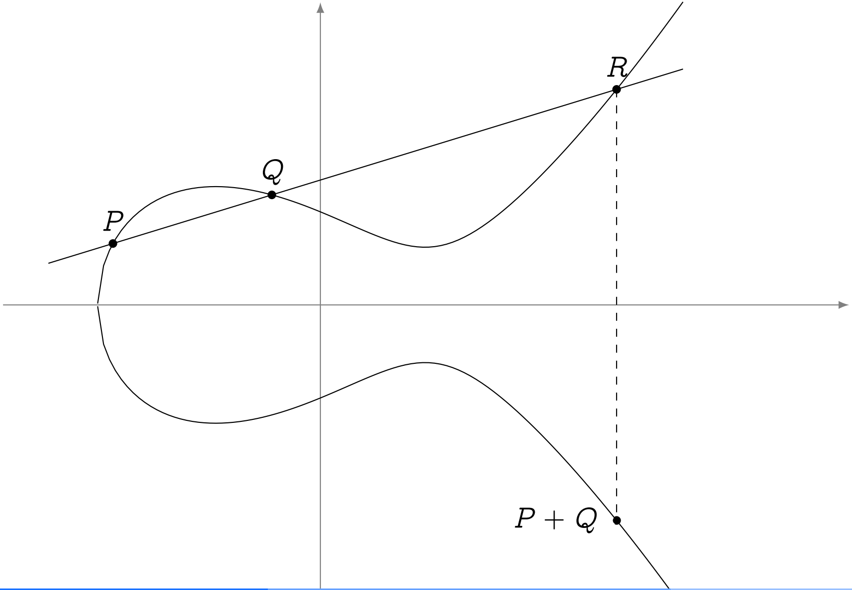
Isogeny graphs



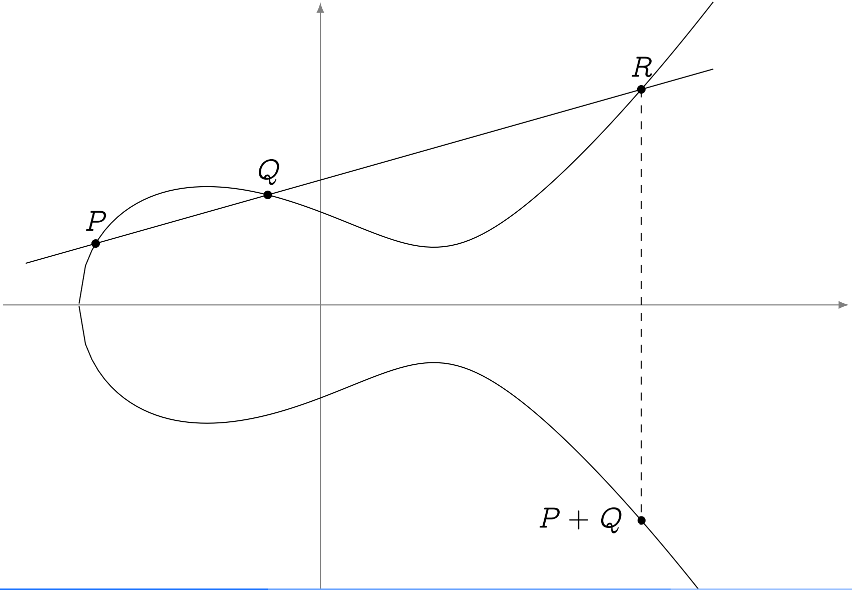
Isogeny graphs



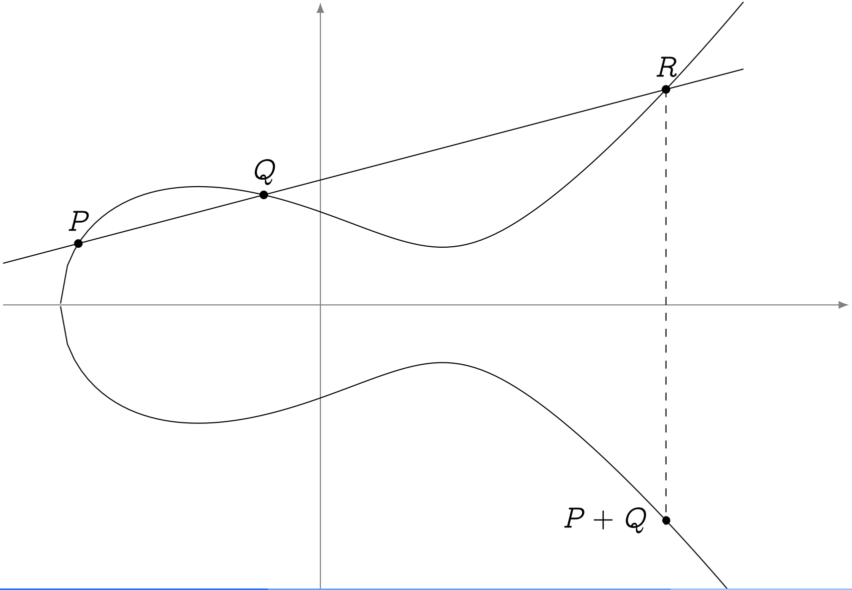
Isogeny graphs



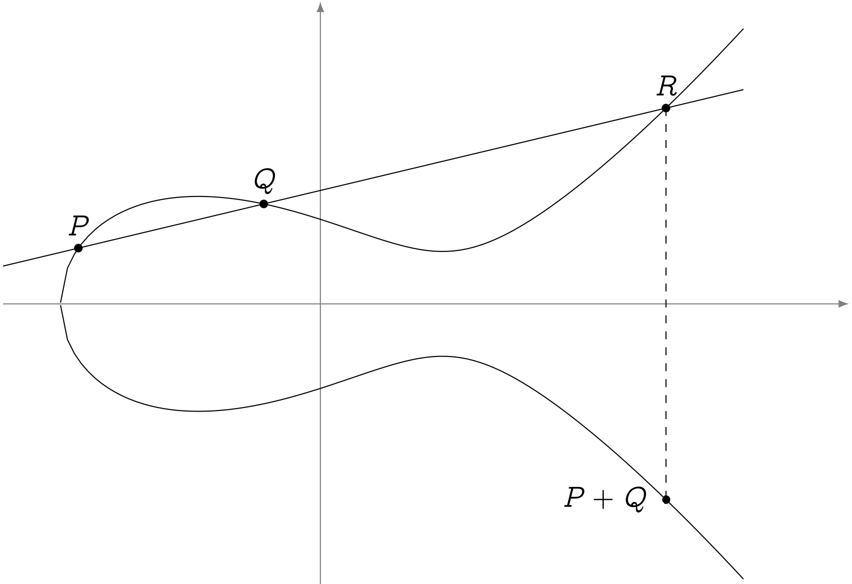
Isogeny graphs



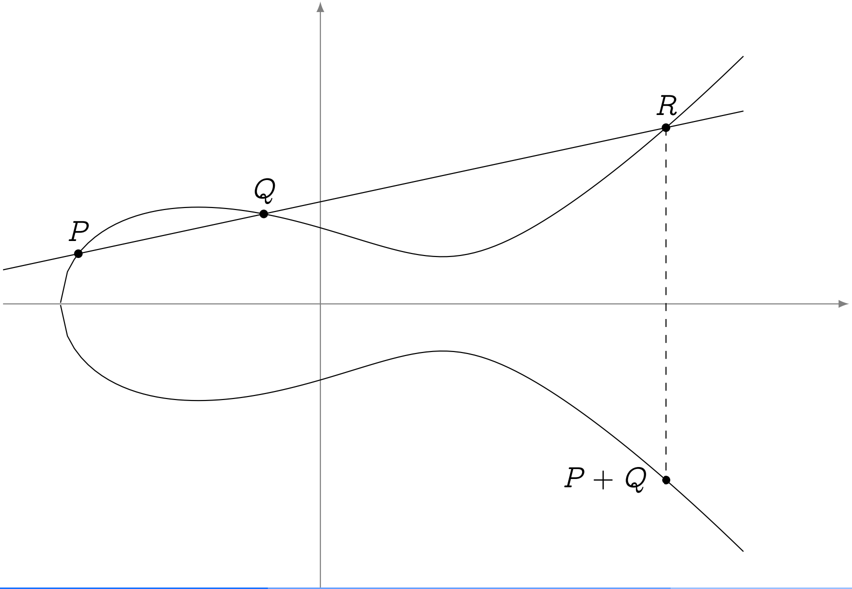
Isogeny graphs



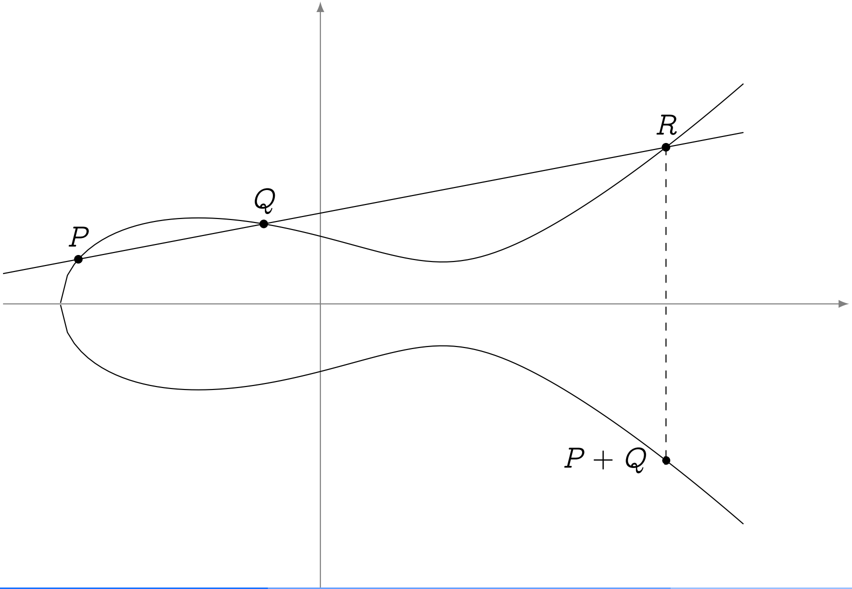
Isogeny graphs



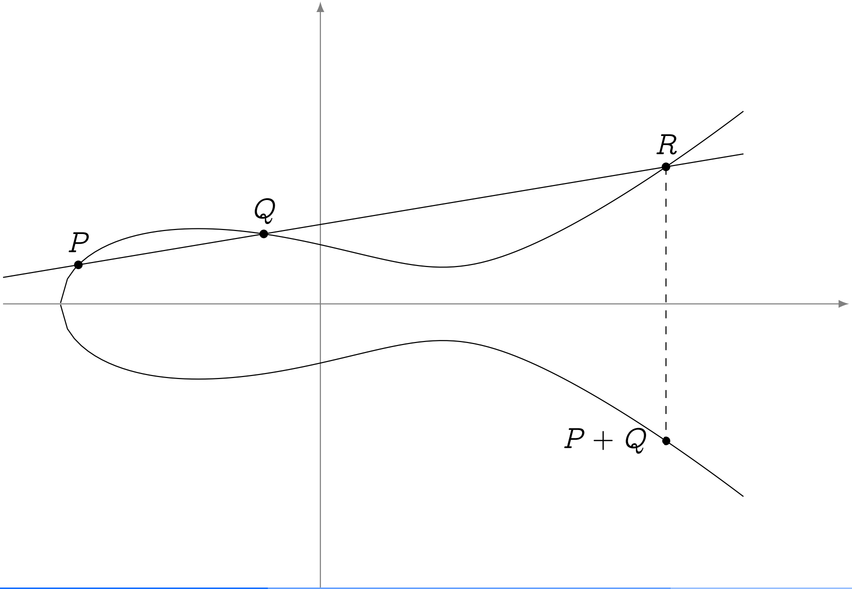
Isogeny graphs



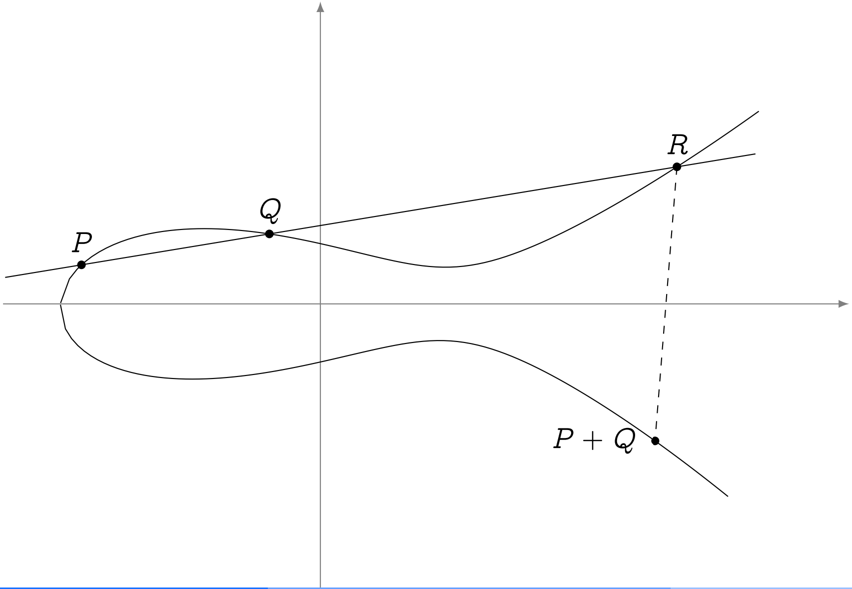
Isogeny graphs



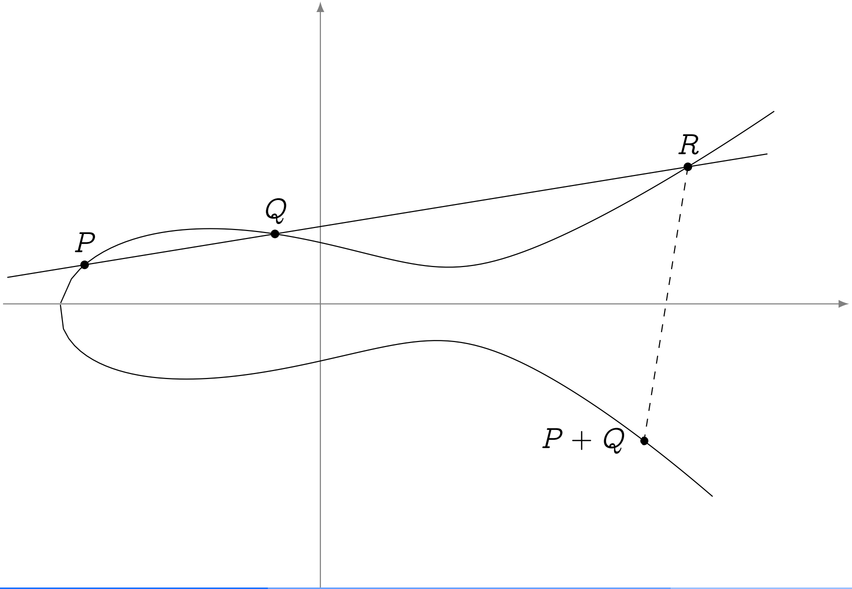
Isogeny graphs



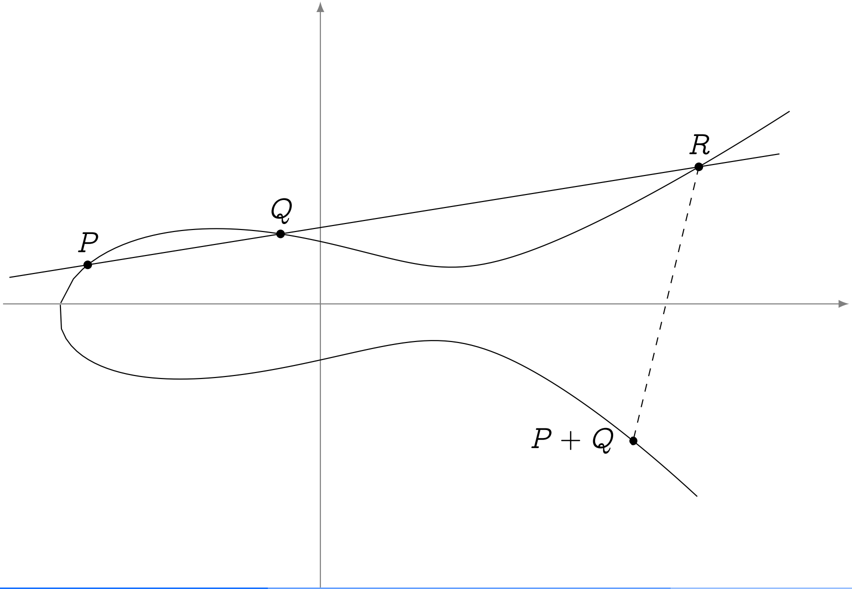
Isogeny graphs



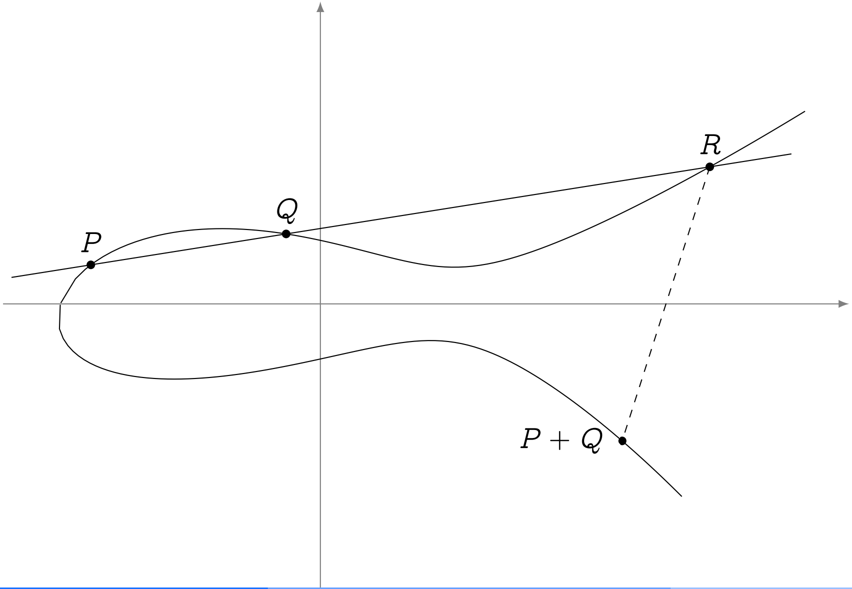
Isogeny graphs



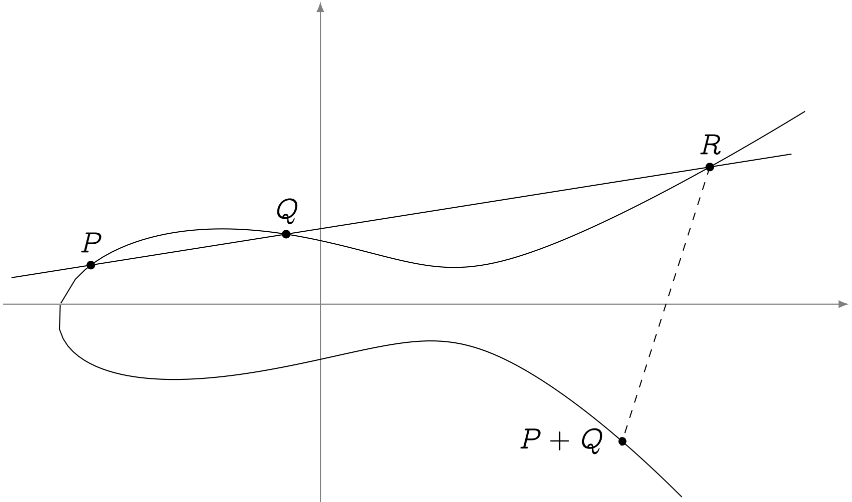
Isogeny graphs



Isogeny graphs

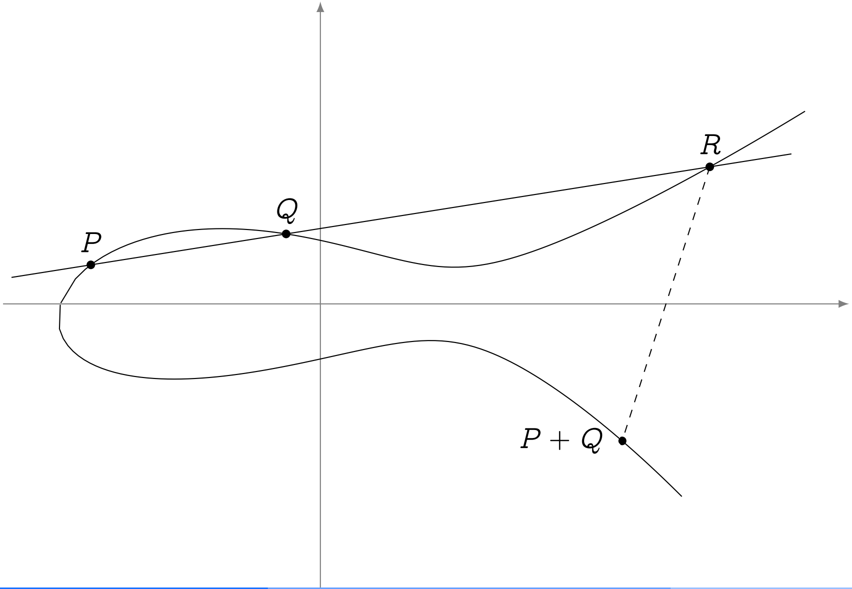


Isogeny graphs

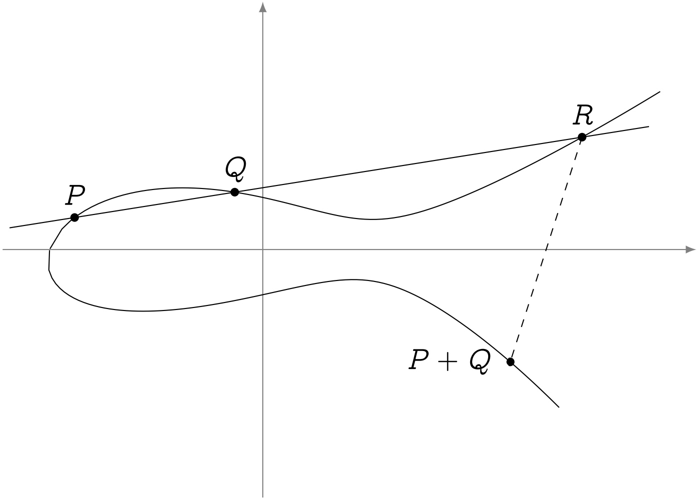


$$y^2 = x^3 + ax + b \quad \longrightarrow \quad j = 1728\frac{4a^3}{4a^3+27b^2}$$

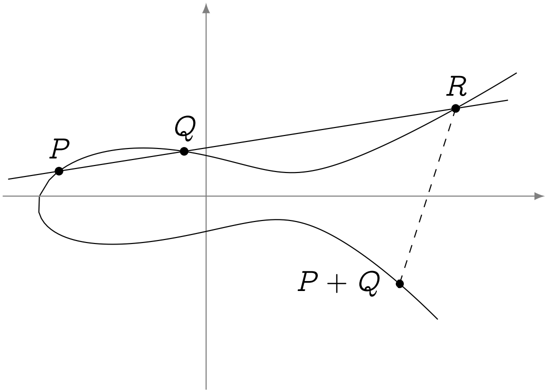
Isogeny graphs



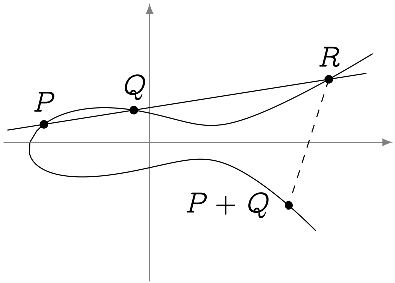
Isogeny graphs



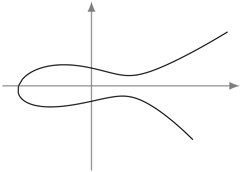
Isogeny graphs



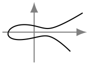
Isogeny graphs



Isogeny graphs



Isogeny graphs



Isogeny graphs

$j = 1728$

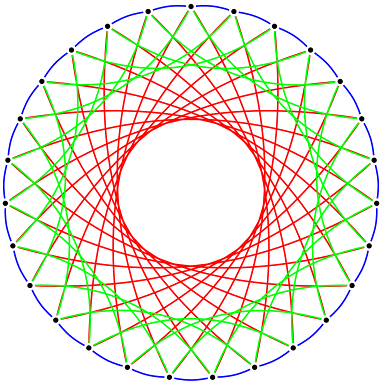
Isogeny graphs

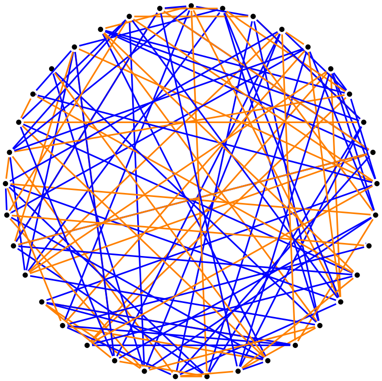
$$j = 1728 \xrightarrow{\phi} j = 287496$$

Isogeny graphs

The beauty and the beast (credit: Lorenz Panny)

Components of particular isogeny graphs look like this:

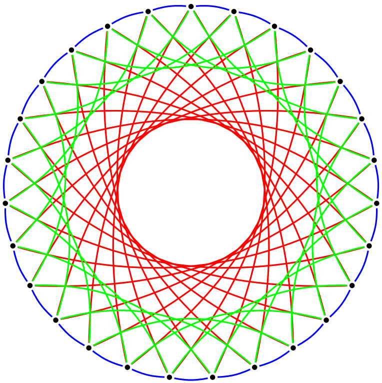


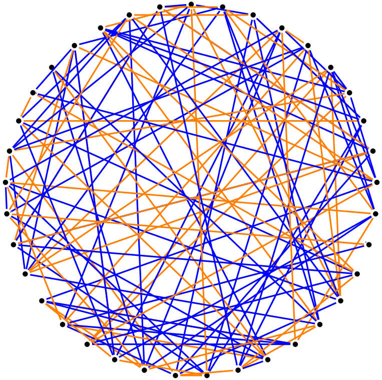


Which of these is good for crypto?

The beauty and the beast (credit: Lorenz Panny)

Components of particular isogeny graphs look like this:

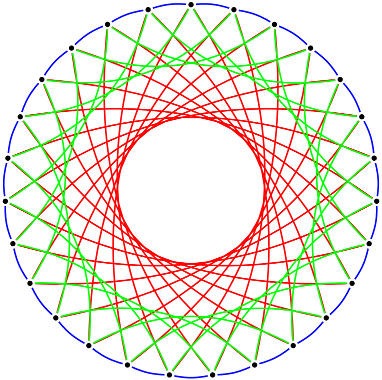




Which of these is good for crypto? **Both.**

The beauty and the beast (credit: Lorenz Panny)

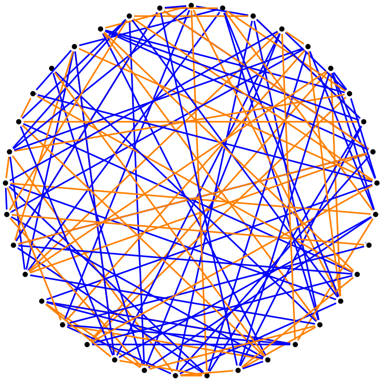
At this time, there are <u>two distinct families</u> of systems:



$\mathbb{F}_p$

CSIDH [pron.: sea-side]

`https://csidh.isogeny.org`



$\mathbb{F}_{p^2}$

SIDH

`https://sike.org`

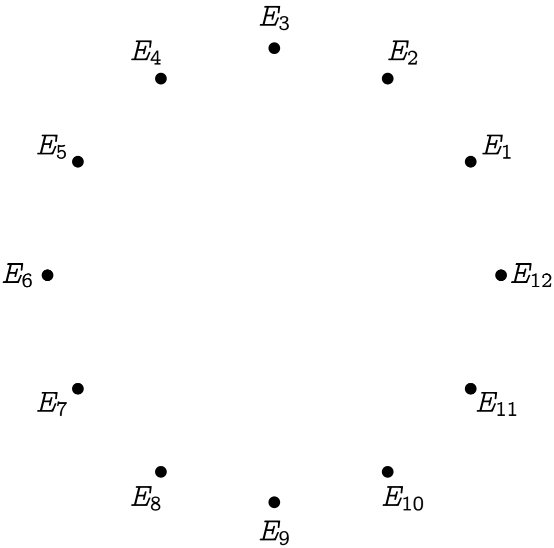
CSIDH vs SIDH

	CSIDH	SIDH
Speed (on x64 arch., NIST 1)	$\sim$ 70ms	$\sim$ 6ms
Public key size (NIST 1)	64B	346B
Key compression		
↳ speed		$\sim$ 11ms
↳ size		209B
Submitted to NIST	no	yes
TRL	4	6
Best classical attack	$p^{1/4}$	$p^{1/4}$ $(p^{3/8})$
Best quantum attack	$\tilde{O}\left(3^{\sqrt{\log_3 p}}\right)$	$p^{1/6}$ $(p^{3/8})$
Key size scales	quadratically	linearly
CPA security	yes	yes
CCA security	yes	Fujisaki-Okamoto
Constant time	it's complicated	yes
Non-interactive key exchange	yes	no
Signatures	short but (slow | do not scale)	big and slow

CSIDH vs SIDH

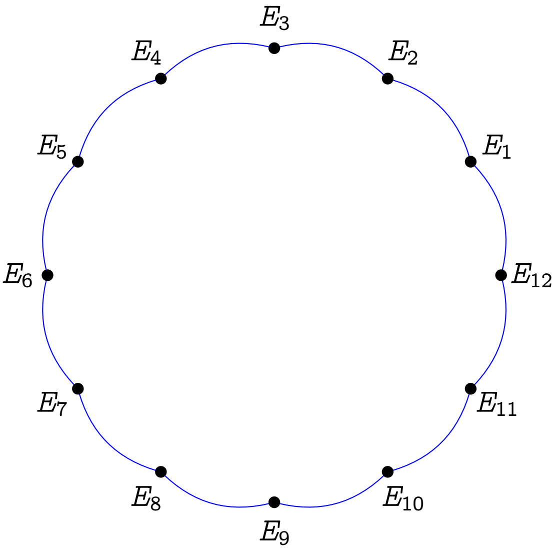
	CSIDH	SIDH
Speed (on x64 arch., NIST 1)	$\sim$ 70ms	$\sim$ 6ms
Public key size (NIST 1)	64B	346B
Key compression		
↳ speed		$\sim$ 11ms
↳ size		209B
Submitted to NIST	no	yes
TRL	4	6
Best classical attack	$p^{1/4}$	$p^{1/4} \quad (p^{3/8})$
Best quantum attack	$\tilde{O}\left(3^{\sqrt{\log_3 p}}\right)$	$p^{1/6} \quad (p^{3/8})$
Key size scales	quadratically	linearly
CPA security	yes	yes
CCA security	yes	Fujisaki-Okamoto
Constant time	it's complicated	yes
Non-interactive key exchange	yes	no
Signatures	short but (slow | do not scale)	big and slow

The CSIDH graph



Vertices are
supersingular elliptic curves
over $\mathbb{F}_p$.

The CSIDH graph

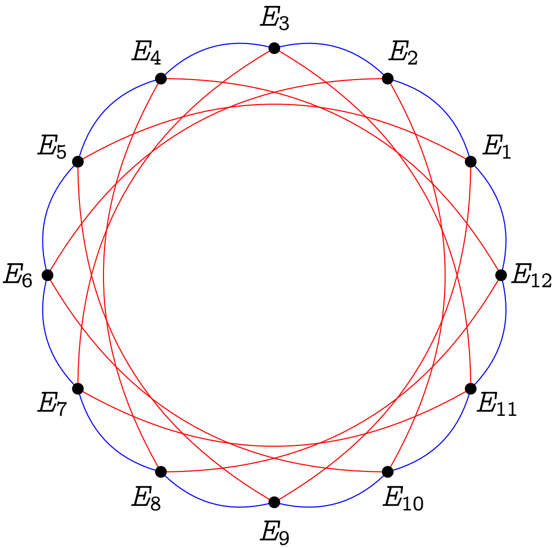


Vertices are supersingular elliptic curves over $\mathbb{F}_p$.

Edges are isogenies of bounded prime degree.

— degree 3

The CSIDH graph



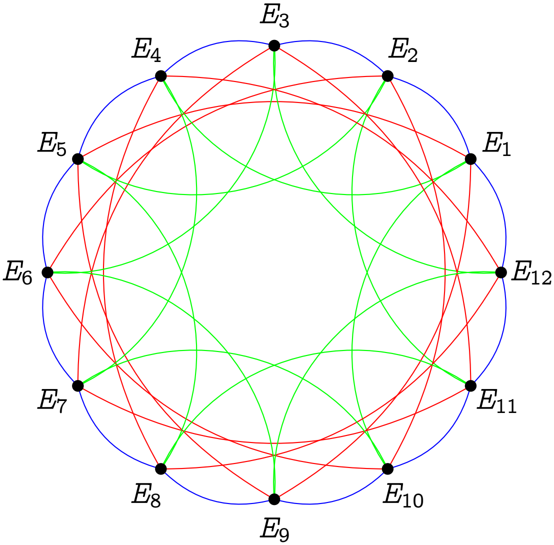
Vertices are supersingular elliptic curves over $\mathbb{F}_p$.

Edges are isogenies of bounded prime degree.

— degree 3

— degree 5

The CSIDH graph



Vertices are supersingular elliptic curves over $\mathbb{F}_p$.

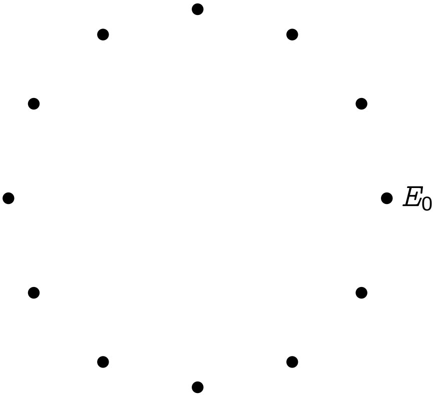
Edges are isogenies of bounded prime degree.

— degree 3

— degree 5

— degree 7

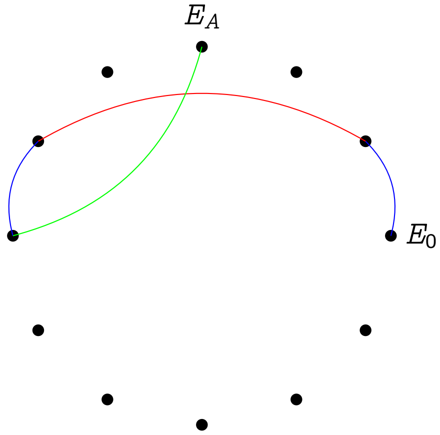
CSIDH key exchange



Public parameters:

- A supersingular curve $E_0/\mathbb{F}_p$;
- A set of small prime degree isogenies.

CSIDH key exchange

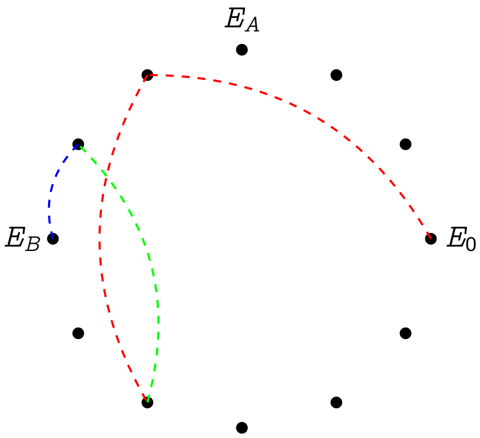


Public parameters:

- A supersingular curve $E_0/\mathbb{F}_p$;
- A set of small prime degree isogenies.

1. **Alice** takes a secret random walk $\phi_A : E_0 \to E_A$ of length $O(\log p)$;

CSIDH key exchange

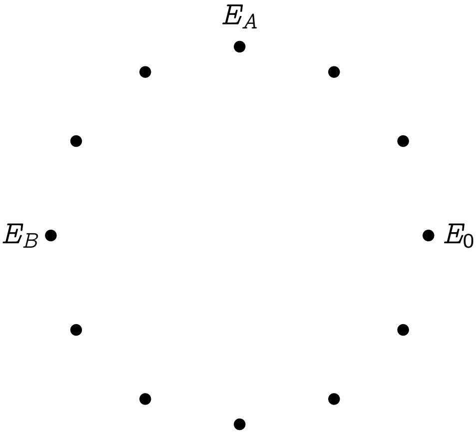


Public parameters:

- A supersingular curve $E_0/\mathbb{F}_p$;
- A set of small prime degree isogenies.

1. **Alice** takes a secret random walk $\phi_A : E_0 \to E_A$ of length $O(\log p)$;
2. **Bob** does the same;

CSIDH key exchange

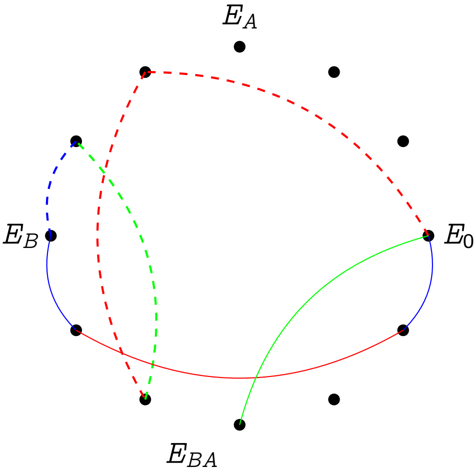


Public parameters:

- A supersingular curve $E_0/\mathbb{F}_p$;
- A set of small prime degree isogenies.

1. **Alice** takes a secret random walk $\phi_A : E_0 \to E_A$ of length $O(\log p)$;
2. **Bob** does the same;
3. They publish E_A and E_B;

CSIDH key exchange

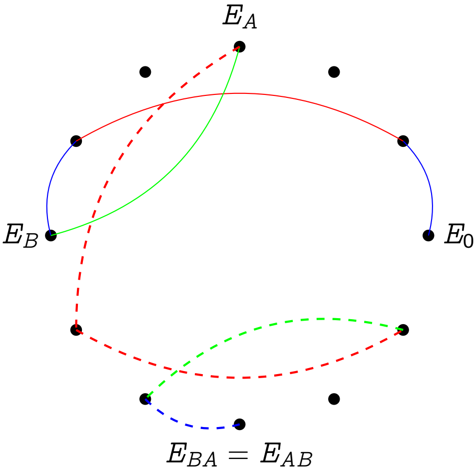


Public parameters:

- A supersingular curve $E_0/\mathbb{F}_p$;
- A set of small prime degree isogenies.

1. **Alice** takes a secret random walk $\phi_A : E_0 \to E_A$ of length $O(\log p)$;
2. **Bob** does the same;
3. They publish E_A and E_B;
4. **Alice** repeats her secret walk ϕ_A starting from E_B.

CSIDH key exchange



Public parameters:

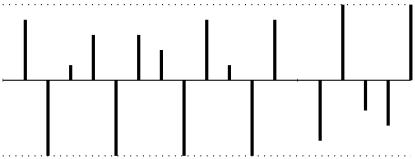
- A supersingular curve $E_0/\mathbb{F}_p$;
- A set of small prime degree isogenies.

1. **Alice** takes a secret random walk $\phi_A : E_0 \rightarrow E_A$ of length $O(\log p)$;
2. **Bob** does the same;
3. They publish E_A and E_B;
4. **Alice** repeats her secret walk ϕ_A starting from E_B.
5. **Bob** repeats his secret walk ϕ_B starting from E_A.

CSIDH data flow

Your secret: a vector of number of isogeny steps for each degree

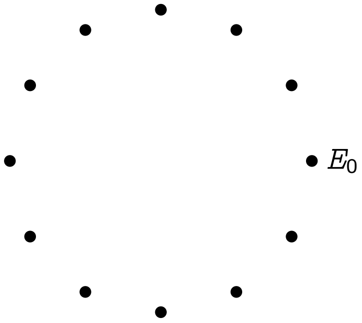
$$(5, 1, -4, \ldots)$$



Your public key: (the j-invariant of) a supersingular elliptic curve

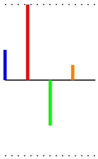
$$j = \texttt{0x23baf75419531a44f3b97cc9d8291a275047fcdae0c9a0c0ebb993964f821f2}$$
$$\texttt{0c11058a4200ff38c4a85e208345300033b0d3119ff4a7c1be0acd62a622002a9}$$

Isogeny evaluation



Repeat:

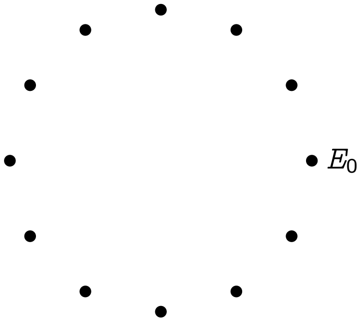
- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.



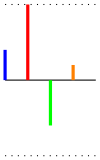
Isogeny evaluation

Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.



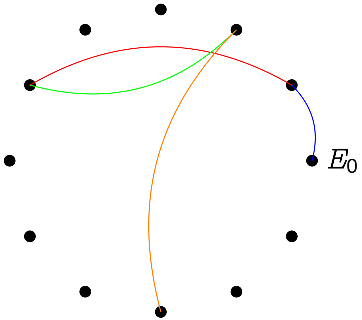
$$\#\langle P \rangle = 3 \cdot 5 \cdot 7 \cdot 11$$



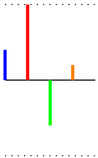
Isogeny evaluation

Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.



$$\#\langle P \rangle = 3 \cdot 5 \cdot 7 \cdot 11$$

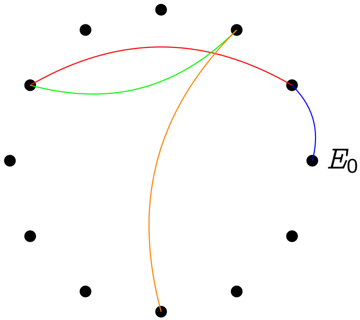


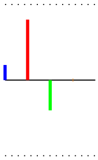
$$Q = P$$

Isogeny evaluation

Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q\rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q\rangle$.

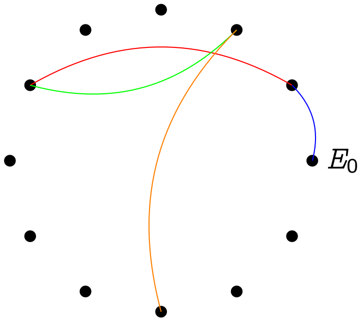




Isogeny evaluation

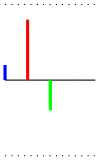
Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$,
  where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.



$$\#\langle P \rangle = 3 \cdot 5 \cdot 7 \cdot 11$$

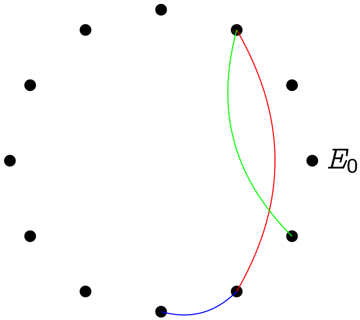
$$Q = [11]P$$



Isogeny evaluation

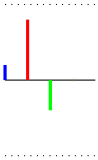
Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$,
  where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.



$$\#\langle P \rangle = 3 \cdot 5 \cdot 7 \cdot 11$$

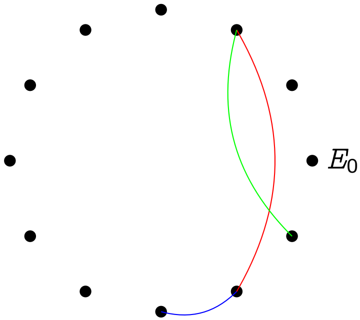
$$Q = [11]P$$

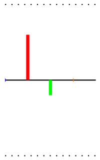


Isogeny evaluation

Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$,
  where c is an appropriate cofactor, so that $N = \#\langle Q\rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q\rangle$.

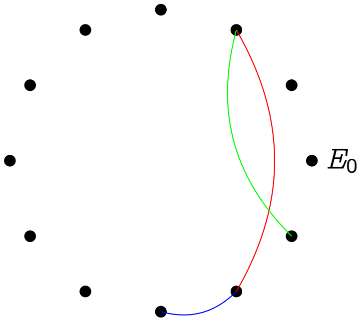




Isogeny evaluation

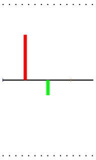
Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.



$$\#\langle P \rangle = 3 \cdot 7 \cdot 11$$

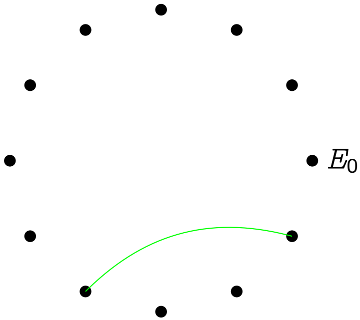
$$Q = [3 \cdot 11]P$$



Isogeny evaluation

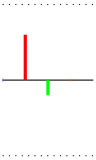
Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q\rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q\rangle$.



$$\#\langle P\rangle = 3 \cdot 7 \cdot 11$$

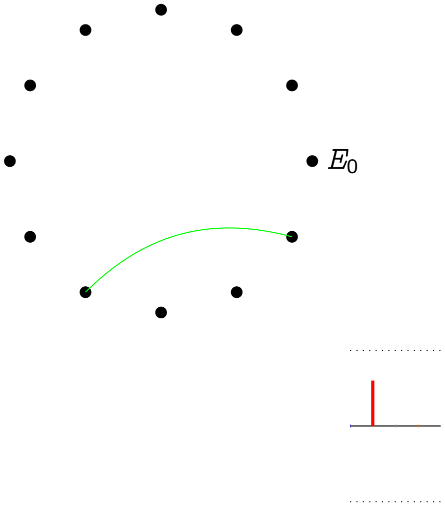
$$Q = [3 \cdot 11]P$$



Isogeny evaluation

Repeat:

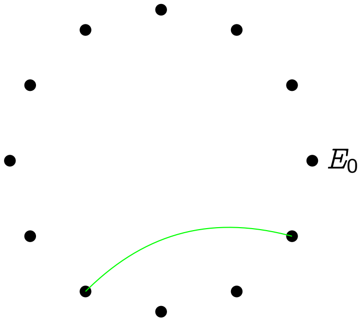
- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.



Isogeny evaluation

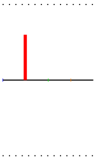
Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$,
  where c is an appropriate cofactor, so that $N = \#\langle Q\rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q\rangle$.



$$\#\langle P\rangle = 3 \cdot 5 \cdot 7 \cdot 11$$

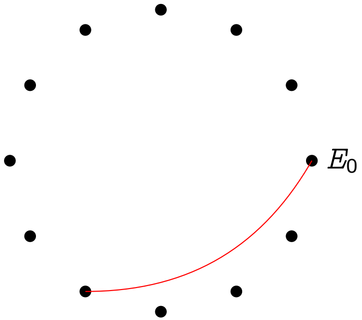
$$Q = [3 \cdot 7 \cdot 11]P$$

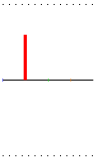


Isogeny evaluation

Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.





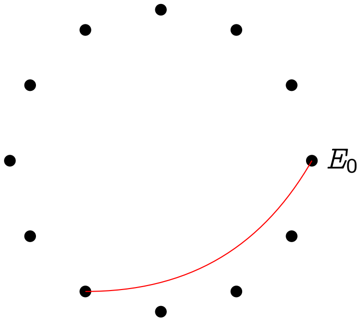
$$\#\langle P \rangle = 3 \cdot 5 \cdot 7 \cdot 11$$

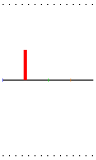
$$Q = [3 \cdot 7 \cdot 11]P$$

Isogeny evaluation

Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.

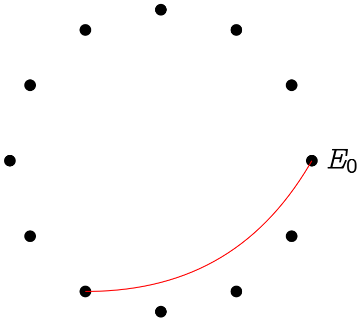




Isogeny evaluation

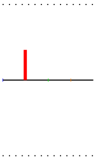
Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q\rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q\rangle$.



$$\#\langle P\rangle = 3 \cdot 5 \cdot 7 \cdot 11$$

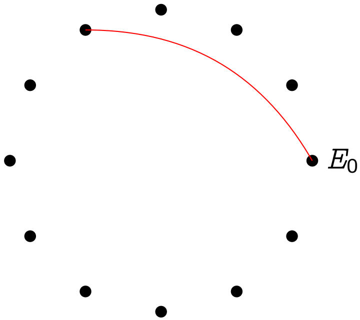
$$Q = [3 \cdot 7 \cdot 11]P$$

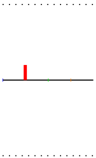


Isogeny evaluation

Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q \rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q \rangle$.

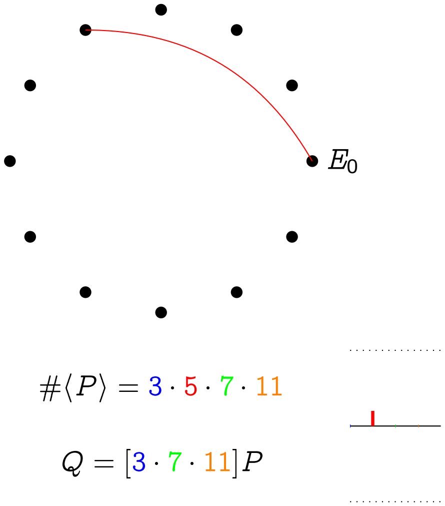




Isogeny evaluation

Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$, where c is an appropriate cofactor, so that $N = \#\langle Q\rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q\rangle$.



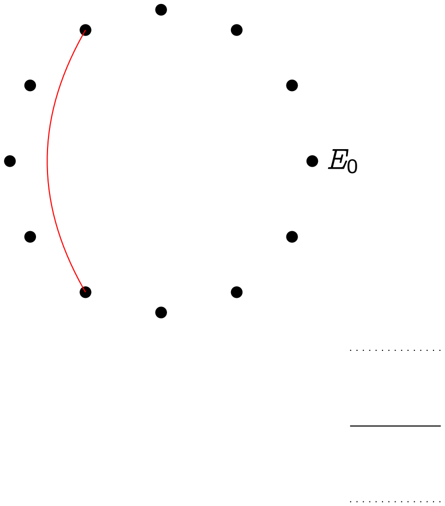
$$\#\langle P\rangle = 3 \cdot 5 \cdot 7 \cdot 11$$

$$Q = [3 \cdot 7 \cdot 11]P$$

Isogeny evaluation

Repeat:

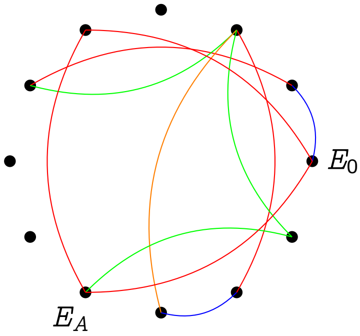
- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$,
  where c is an appropriate cofactor, so that $N = \#\langle Q\rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q\rangle$.

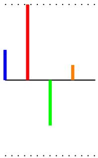


Isogeny evaluation

Repeat:

- Take a random point $P \in E(\mathbb{F}_p)$;
- Set $Q = [c]P$,
  where c is an appropriate cofactor, so that $N = \#\langle Q\rangle$ contains only useful prime factors;
- Advance by the degree N isogeny of kernel $\langle Q\rangle$.





In seek of constant time

Two obstacles for constant time:

1. Some random points P may lack some factors;
2. Number of isogeny evaluations dependent on secret key.

In seek of constant time

Two obstacles for constant time:

1. ~~Some random points P may lack some factors;~~ Unrelated to secret key if truly random.
2. Number of isogeny evaluations dependent on secret key.

In seek of constant time

Two obstacles for constant time:

1. ~~Some random points P may lack some factors;~~ Unrelated to secret key if truly random.
2. Number of isogeny evaluations dependent on secret key.

Meyer, Campos, Reith 2018; Onuki, Aikawa, Yamazaki, Takagi 2019

- "Dummy" isogenies:
  - Always do exactly the same number of isogeny evaluations per prime degree,
  - discard computations in excess;
- $4\times$ slowdown (MCR) / $2.5\times$ slowdown (OAYT).
- Protected against SPA…

In seek of constant time

Two obstacles for constant time:

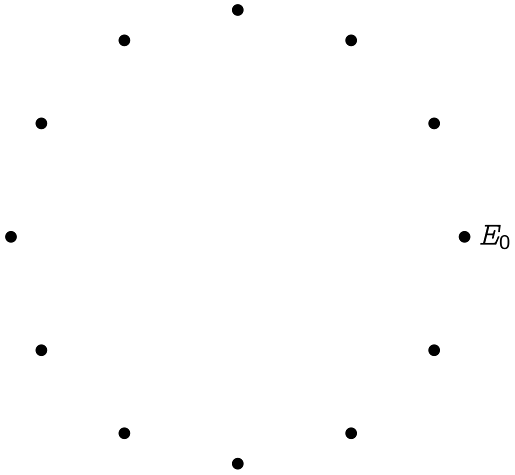
1. ~~Some random points P may lack some factors;~~ Unrelated to secret key if truly random.
2. Number of isogeny evaluations dependent on secret key.

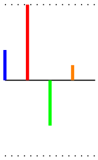
Meyer, Campos, Reith 2018; Onuki, Aikawa, Yamazaki, Takagi 2019

- "Dummy" isogenies:
  - Always do exactly the same number of isogeny evaluations per prime degree,
  - discard computations in excess;
- $4\times$ slowdown (MCR) / $2.5\times$ slowdown (OAYT).
- Protected against SPA... but very easy to attack by fault!

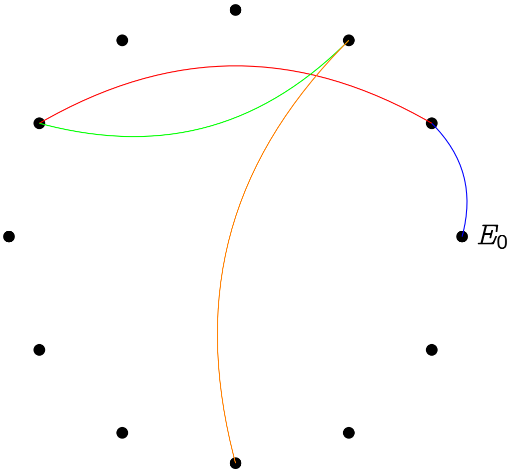

FO4305OZ02

Isogeny evaluation with dummies

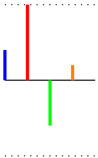




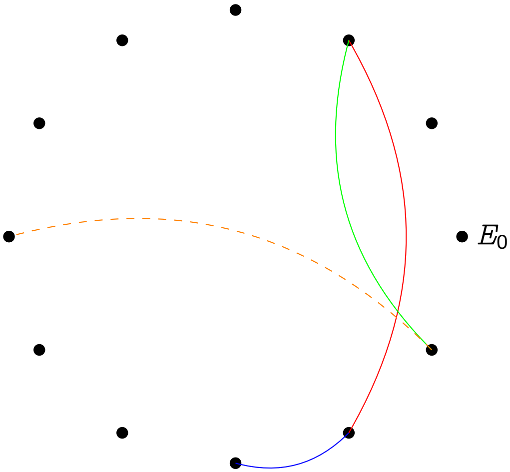
Isogeny evaluation with dummies



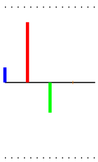
$$\#\langle P\rangle = 3\cdot 5\cdot 7\cdot 11$$



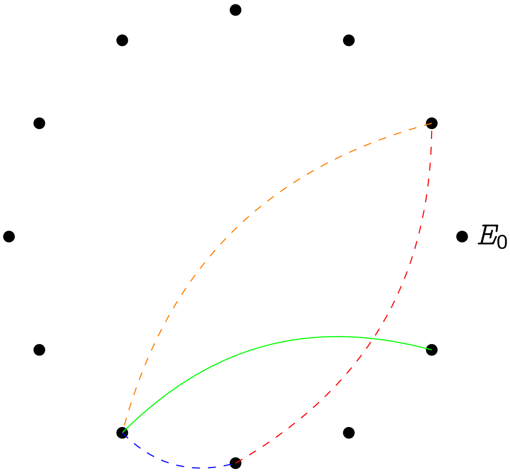
Isogeny evaluation with dummies



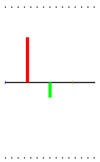
$$\#\langle P\rangle = 3 \cdot 5 \cdot 7 \cdot 11$$



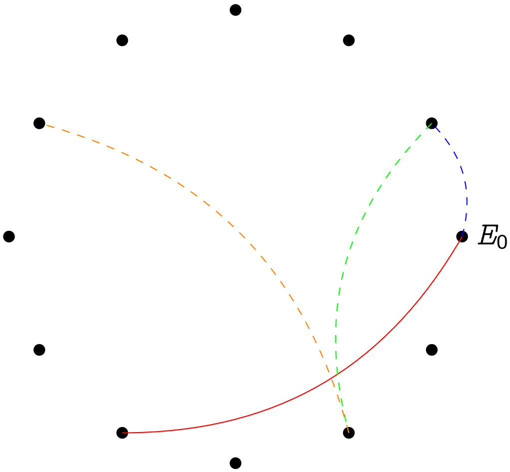
Isogeny evaluation with dummies



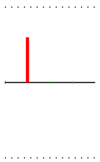
$\#\langle P\rangle = 3 \cdot 7 \cdot 11$



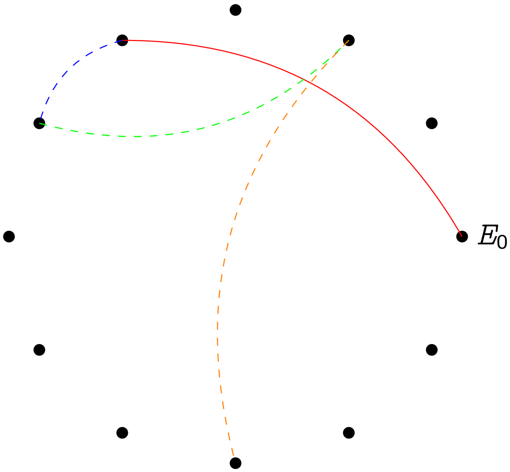
Isogeny evaluation with dummies



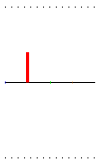
$$\#\langle P\rangle = 3 \cdot 5 \cdot 7 \cdot 11$$



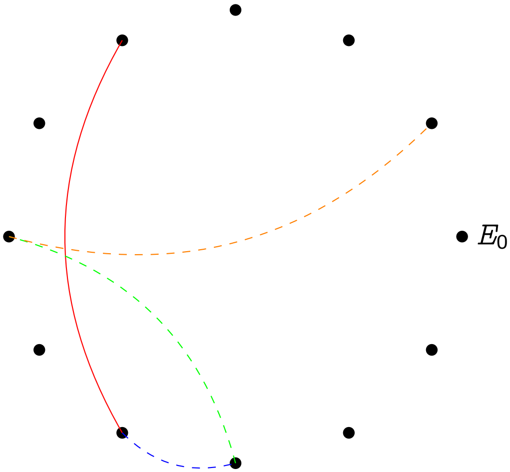
Isogeny evaluation with dummies



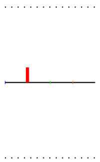
$\#\langle P\rangle = 3 \cdot 5 \cdot 7 \cdot 11$



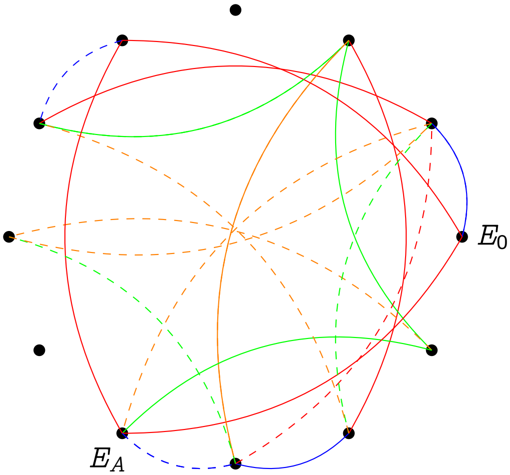
Isogeny evaluation with dummies



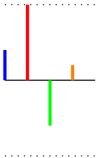
$$\#\langle P\rangle = 3\cdot 5\cdot 7\cdot 11$$



Isogeny evaluation with dummies



$\#\langle P\rangle = 3 \cdot 5 \cdot 7 \cdot 11$



Our work (Latincrypt 2019)

- Fixed a leak related to the sampling of random points.

Our work (Latincrypt 2019)

- Fixed a leak related to the sampling of random points.
- Speed-up both MCR and OAYT constant time implementations:
  - Fully Twisted Edwards implementation;
  - Use of Shortest Differential Addition Chains;

Our work (Latincrypt 2019)

- Fixed a leak related to the sampling of random points.
- Speed-up both MCR and OAYT constant time implementations:
  - Fully Twisted Edwards implementation;
  - Use of Shortest Differential Addition Chains;
- Protection against fault attack at the cost of a $2\times$ slowdown:
  - Got rid of "dummy isogenies".

Our work (Latincrypt 2019)

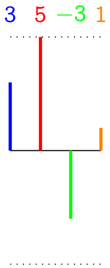
- Fixed a leak related to the sampling of random points.
- Speed-up both MCR and OAYT constant time implementations:
  - Fully Twisted Edwards implementation;
  - Use of Shortest Differential Addition Chains;
- Protection against fault attack at the cost of a $2\times$ slowdown:
  - Got rid of "dummy isogenies".
- Initiated study of fully constant time variant (very expensive, though).

Avoiding dummies

We change the format of the secret key:

Original: vectors with coefficients in $[-B, B]$.

Modified: vectors with odd[1] coefficients in $[-B, B]$.



[1]Or even, all the same.

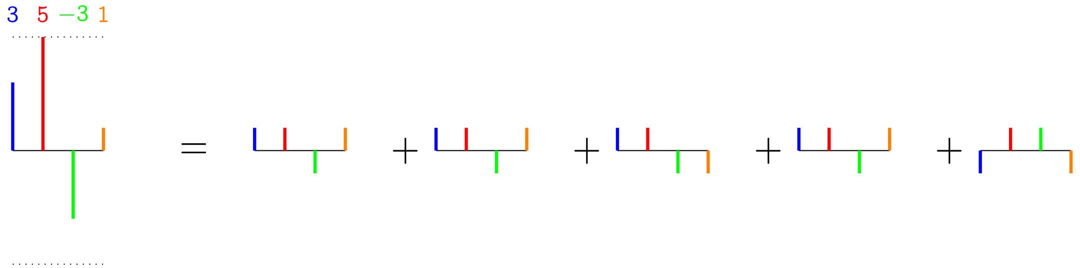
Avoiding dummies

We change the format of the secret key:

Original: vectors with coefficients in $[-B, B]$.

Modified: vectors with odd[1] coefficients in $[-B, B]$.

- Translate vector to sum of ± 1 vectors;
- Each vector costs exactly one isogeny evaluation per degree.



[1]Or even, all the same.

Running-time: measured clock cycles

Clock cycle counts for constant-time CSIDH implementations, averaged over 1024 experiments. The ratio is computed using MCR 2018 as baseline implementation.

Implementation	**CSIDH algorithm**	**Mcycles**	**Ratio**
Castryck et al.	unprotected, unmodified	155	0.39
Meyer–Campos–Reith	unmodified	395	1.00
This work	MCR-style	337	0.85
	OAYT-style	239	0.61
	No-dummy	481	1.22

Summary

- Repeat with me: I need isogeny-based crypto!
- CSIDH is the new Diffie–Hellman:
  Very short keys, easy key validation, ...
- Implementing isogeny-based crypto efficiently is challenging, even more so with side-channel protections.

Thank you

https://defeo.lu/

@luca_defeo