
CSIDH
['siːˌsaɪd]
Lorenz Panny, TU/e

Post-quantum Diffie–Hellman?

Traditionally, Diffie–Hellman works in a group G via the map

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x. \end{array}$$

Post-quantum Diffie–Hellman?

Traditionally, Diffie–Hellman works in a group G via the map

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x. \end{array}$$

Shor's algorithm quantumly computes discrete logarithms in any group in polynomial time.

Post-quantum Diffie–Hellman!

Traditionally, Diffie–Hellman works in a group G via the map

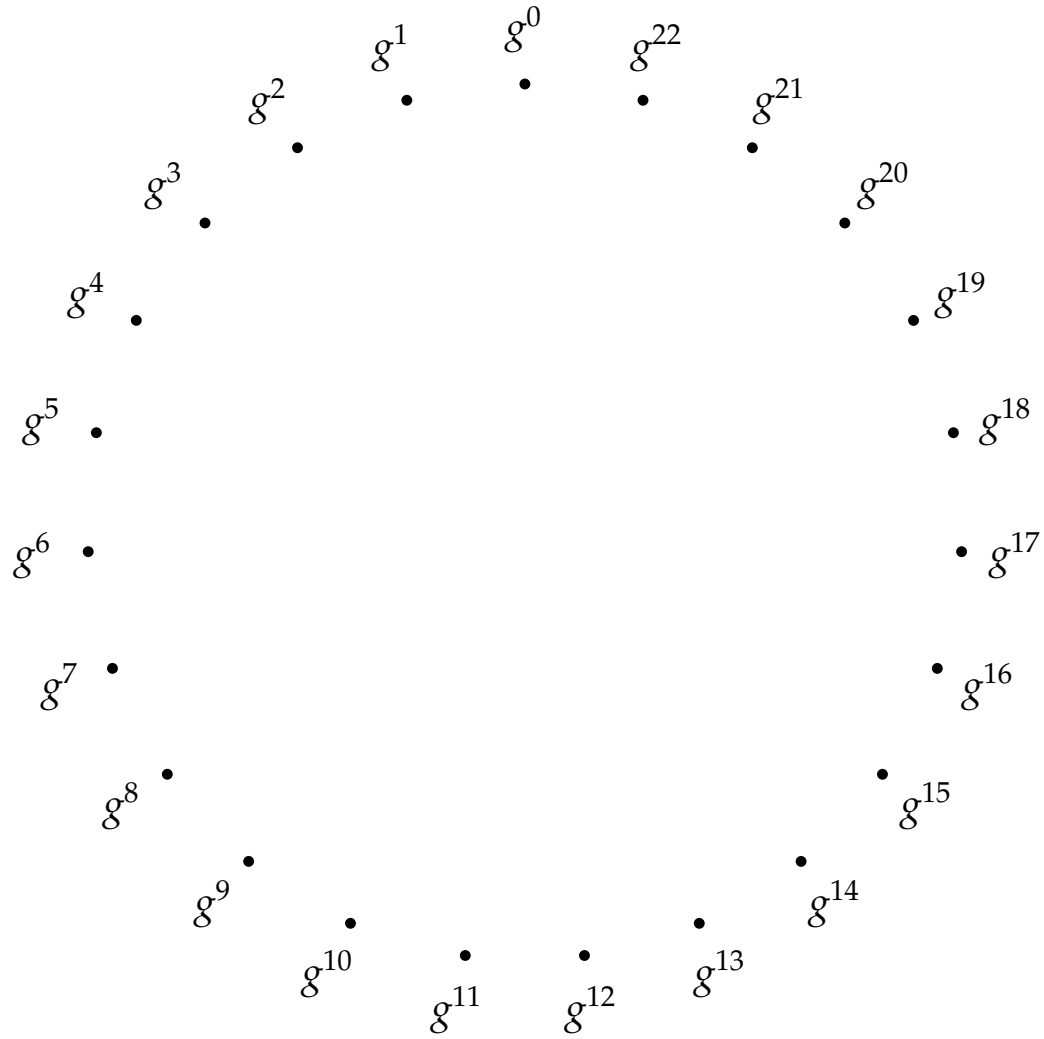
$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x. \end{array}$$

Shor's algorithm quantumly computes discrete logarithms in any group in polynomial time.

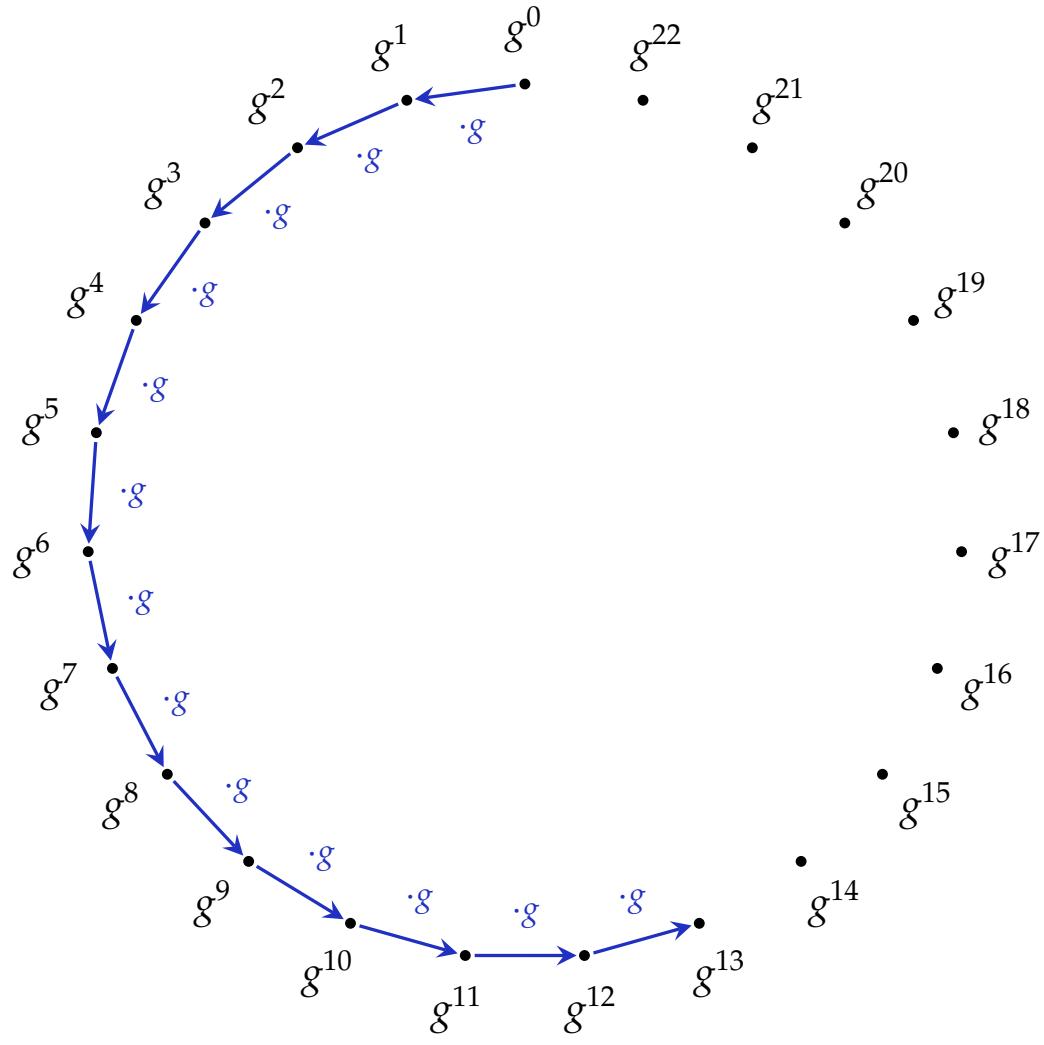
Shor relies on composing public keys: $g^x \cdot g^y = g^{x+y}$. $\rightsquigarrow$ Idea:

Replace exponentiation on the group G by a group action mof a group H on a set S:

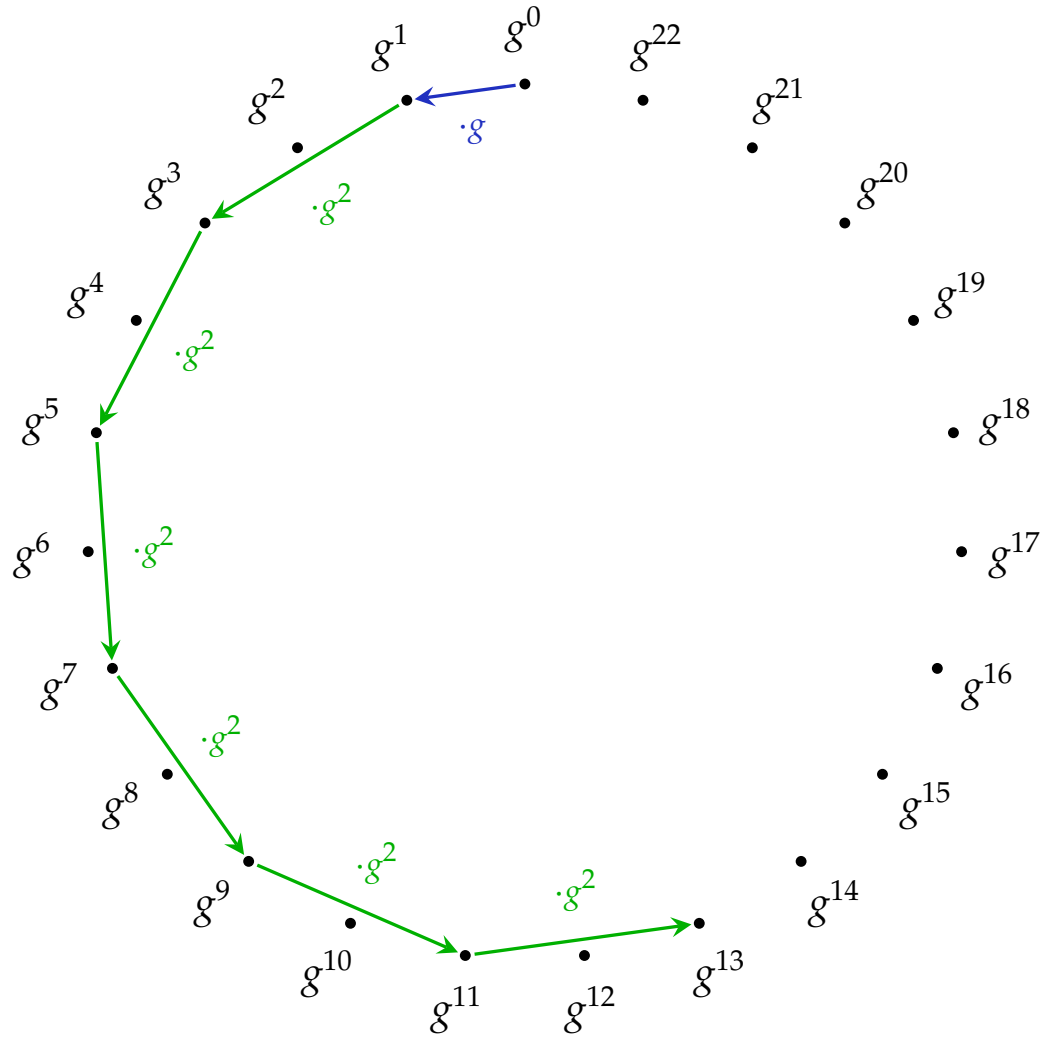
$$H \times S \to S.$$


g^0
g^1
g^2
g^3
g^4
g^5
g^6
g^7
g^8
g^9
g^{10}
g^{11}
g^{12}
g^{13}
g^{14}
g^{15}
g^{16}
g^{17}
g^{18}
g^{19}
g^{20}
g^{21}
g^{22}

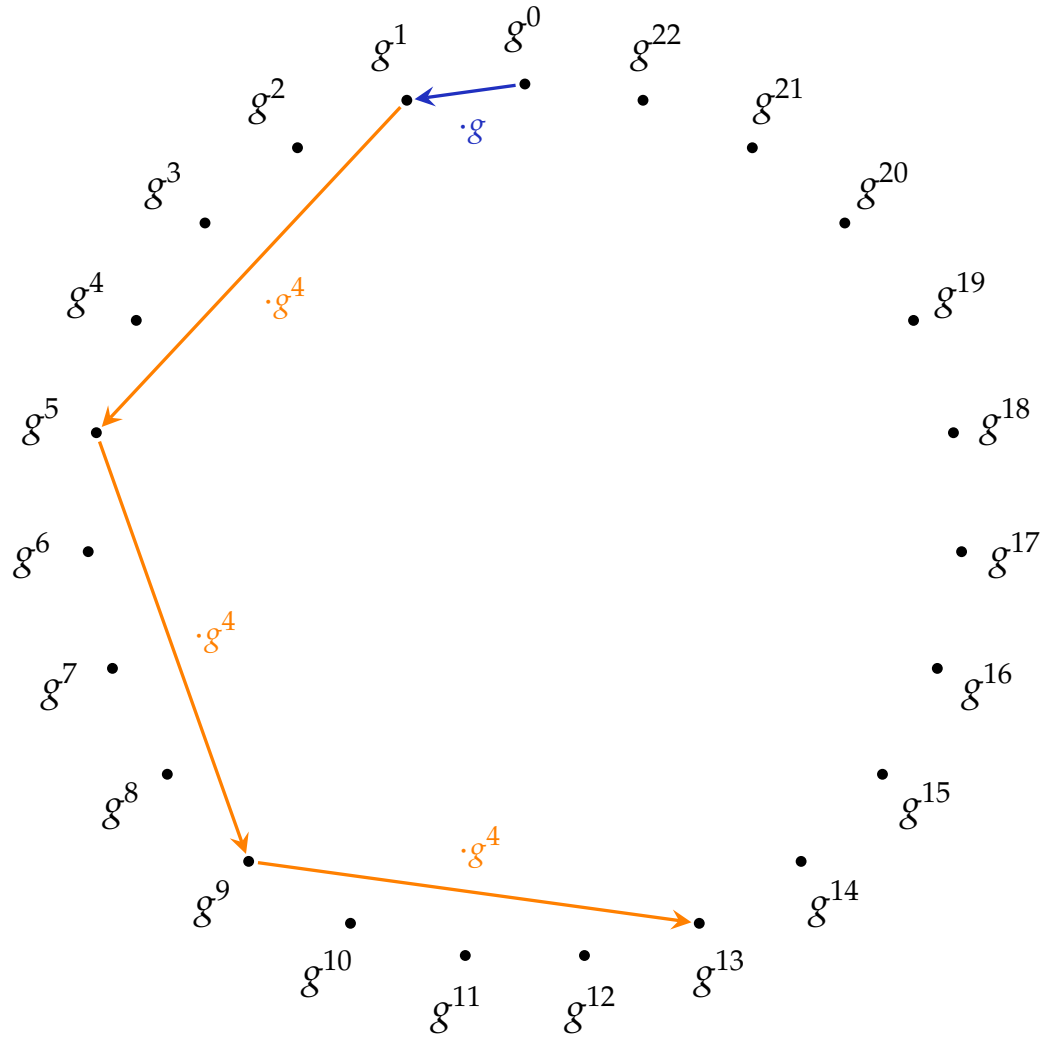
multiply


g^0
g^1
g^2
g^3
g^4
g^5
g^6
g^7
g^8
g^9
g^{10}
g^{11}
g^{12}
g^{13}
g^{14}
g^{15}
g^{16}
g^{17}
g^{18}
g^{19}
g^{20}
g^{21}
g^{22}
$\cdot g$

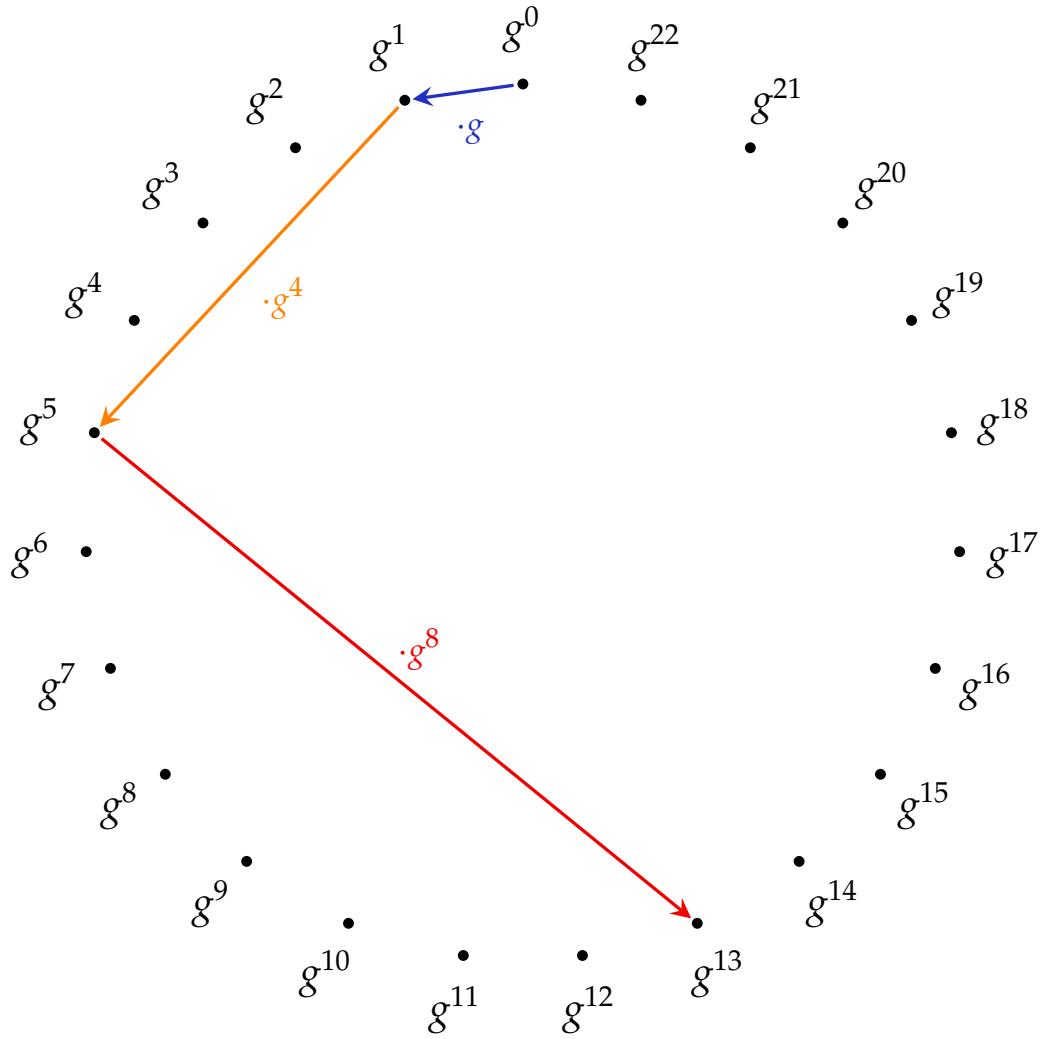
Square-and-multiply



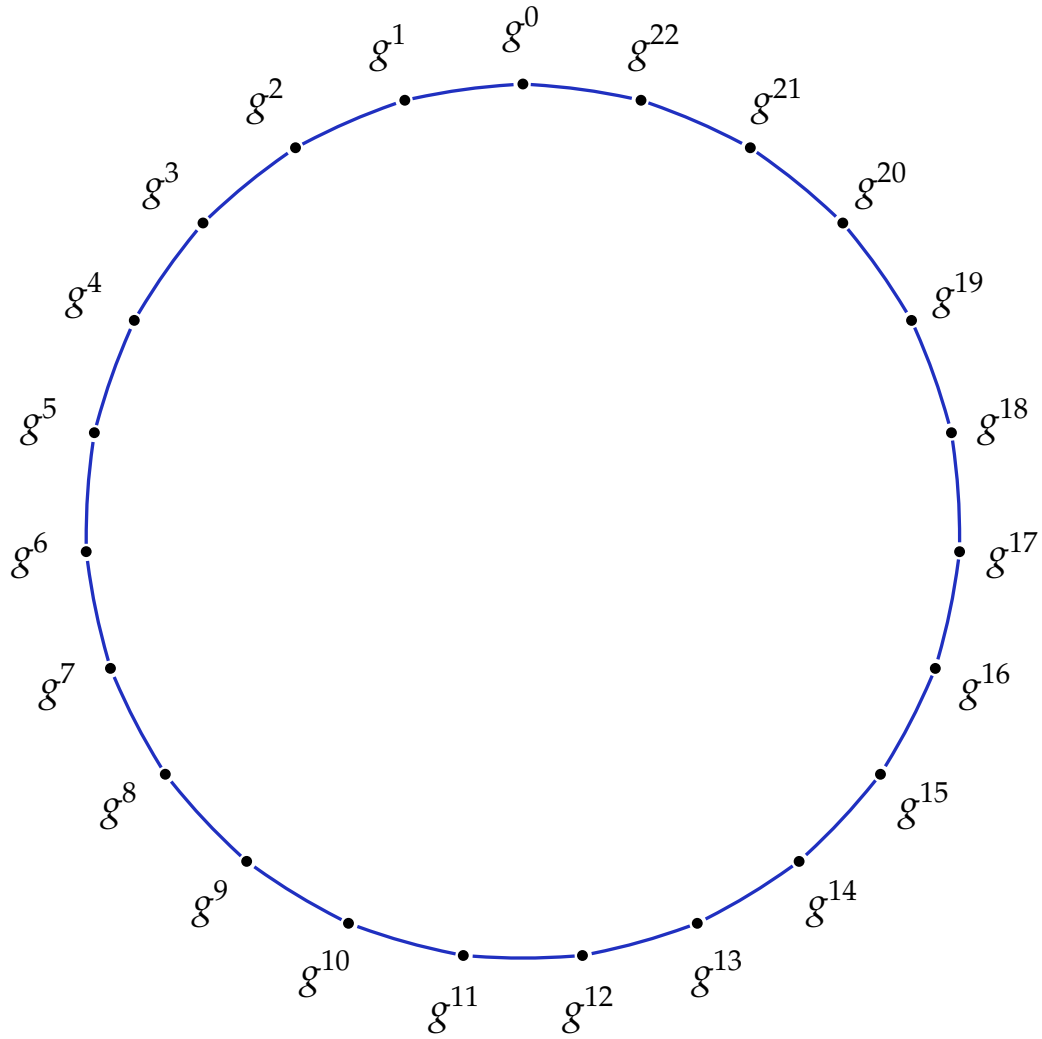
Square-and-multiply-and-square-and-multiply



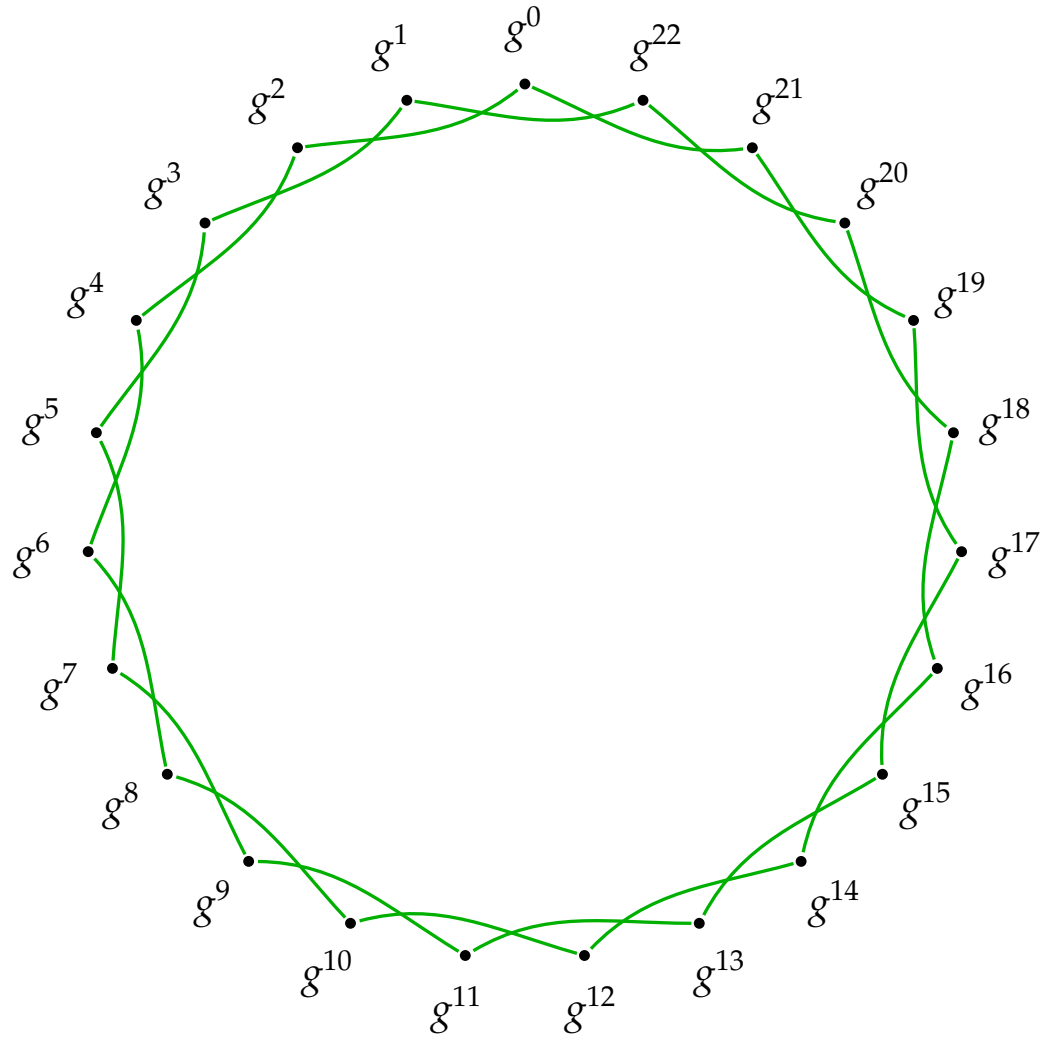
Square-and-multiply-and-square-and-multiply-and-squ



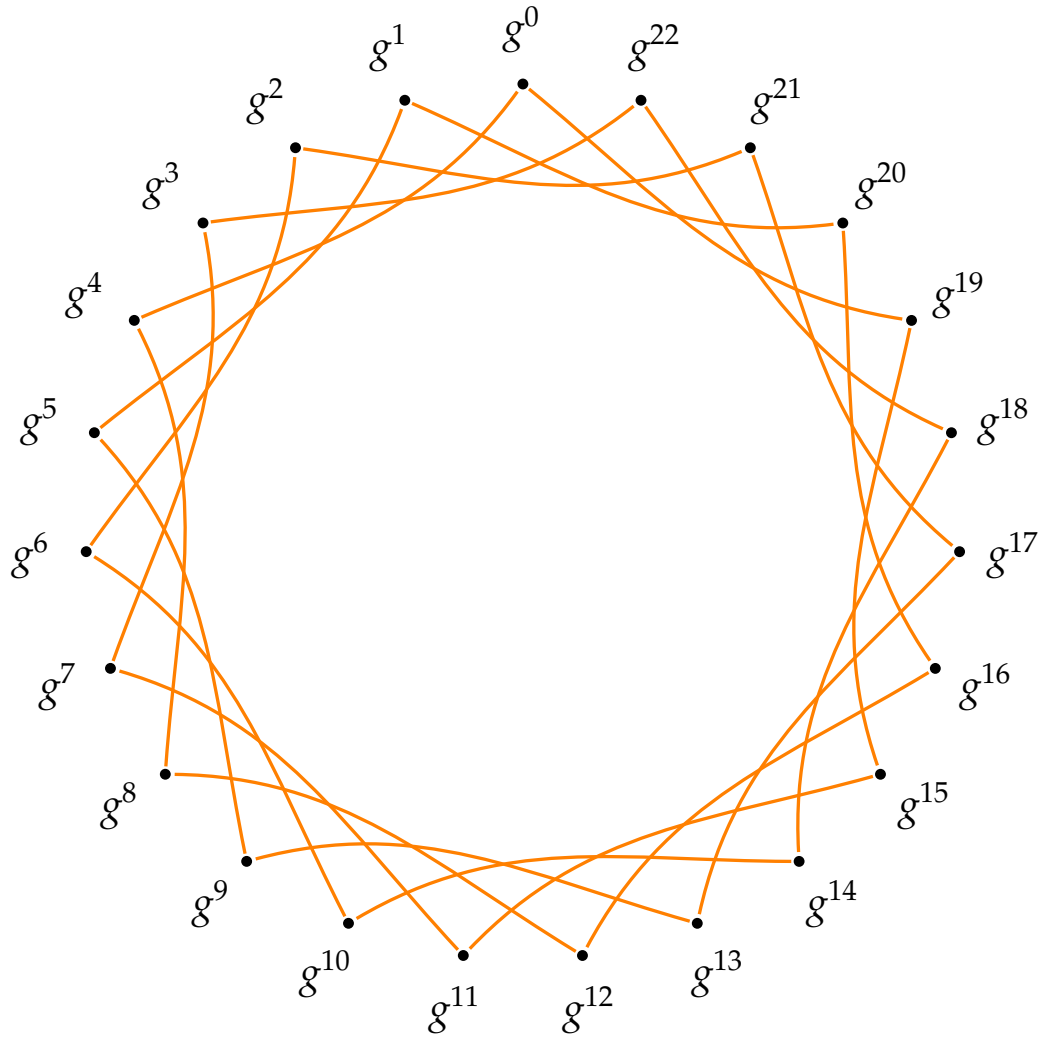
cycles


g^0
g^1
g^2
g^3
g^4
g^5
g^6
g^7
g^8
g^9
g^{10}
g^{11}
g^{12}
g^{13}
g^{14}
g^{15}
g^{16}
g^{17}
g^{18}
g^{19}
g^{20}
g^{21}
g^{22}

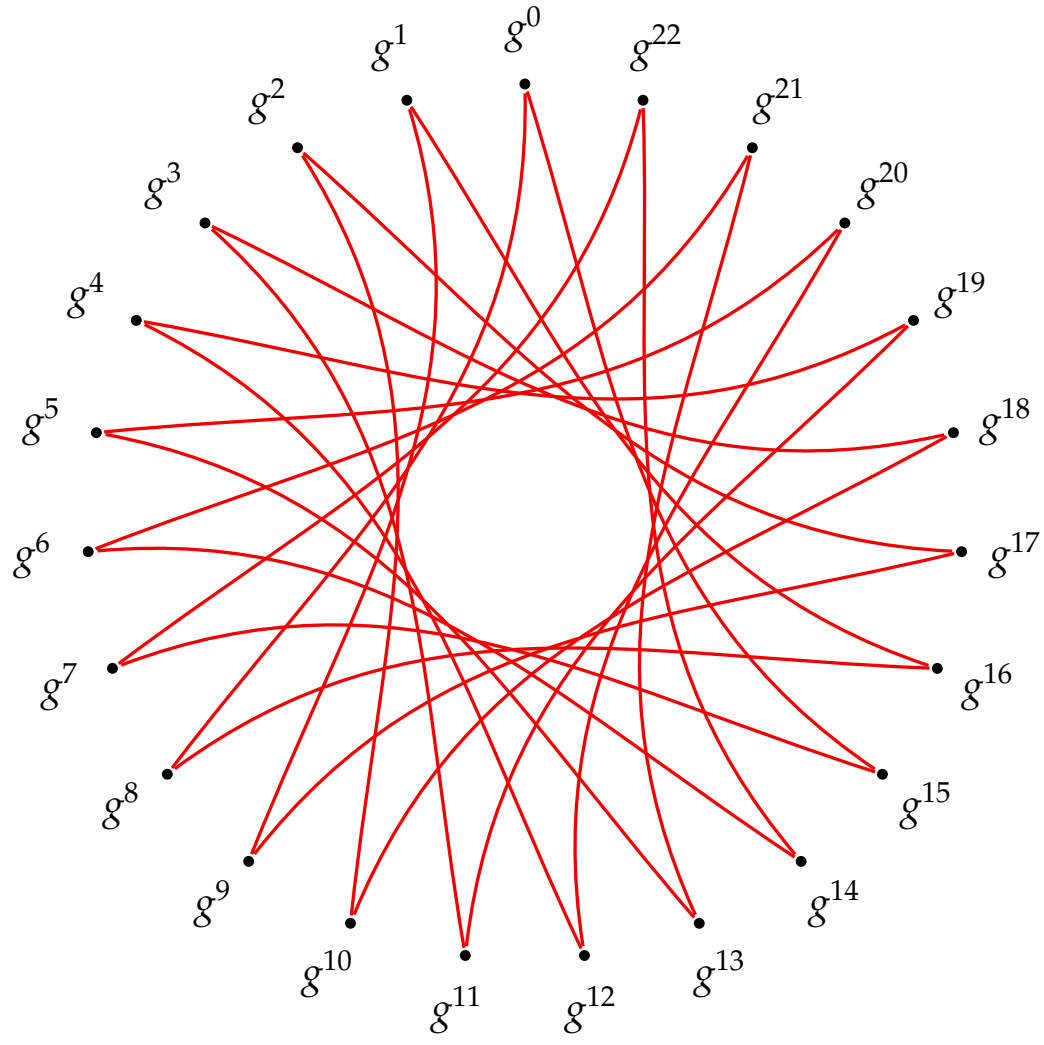
cycles


g^0
g^1
g^2
g^3
g^4
g^5
g^6
g^7
g^8
g^9
g^{10}
g^{11}
g^{12}
g^{13}
g^{14}
g^{15}
g^{16}
g^{17}
g^{18}
g^{19}
g^{20}
g^{21}
g^{22}

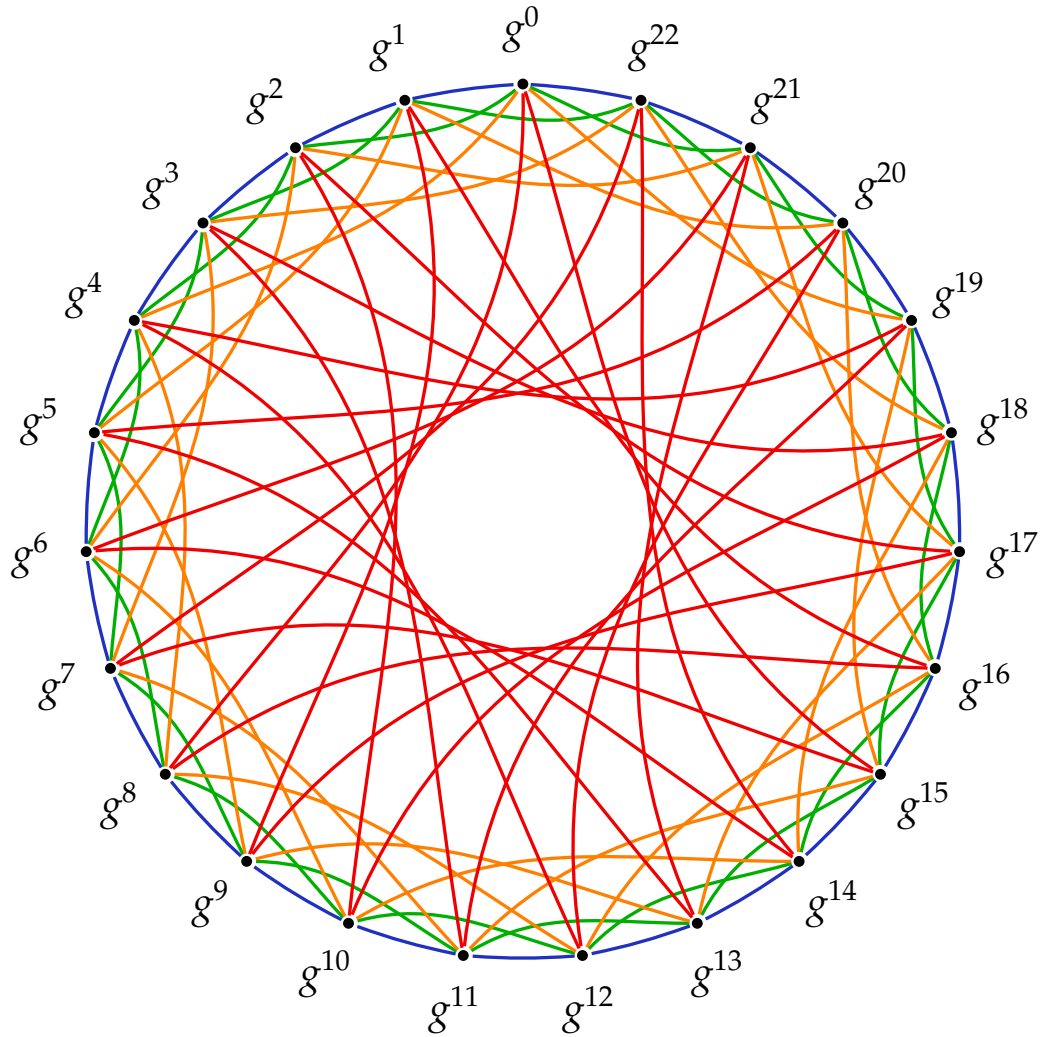
cycles


g^0
g^1
g^2
g^3
g^4
g^5
g^6
g^7
g^8
g^9
g^{10}
g^{11}
g^{12}
g^{13}
g^{14}
g^{15}
g^{16}
g^{17}
g^{18}
g^{19}
g^{20}
g^{21}
g^{22}

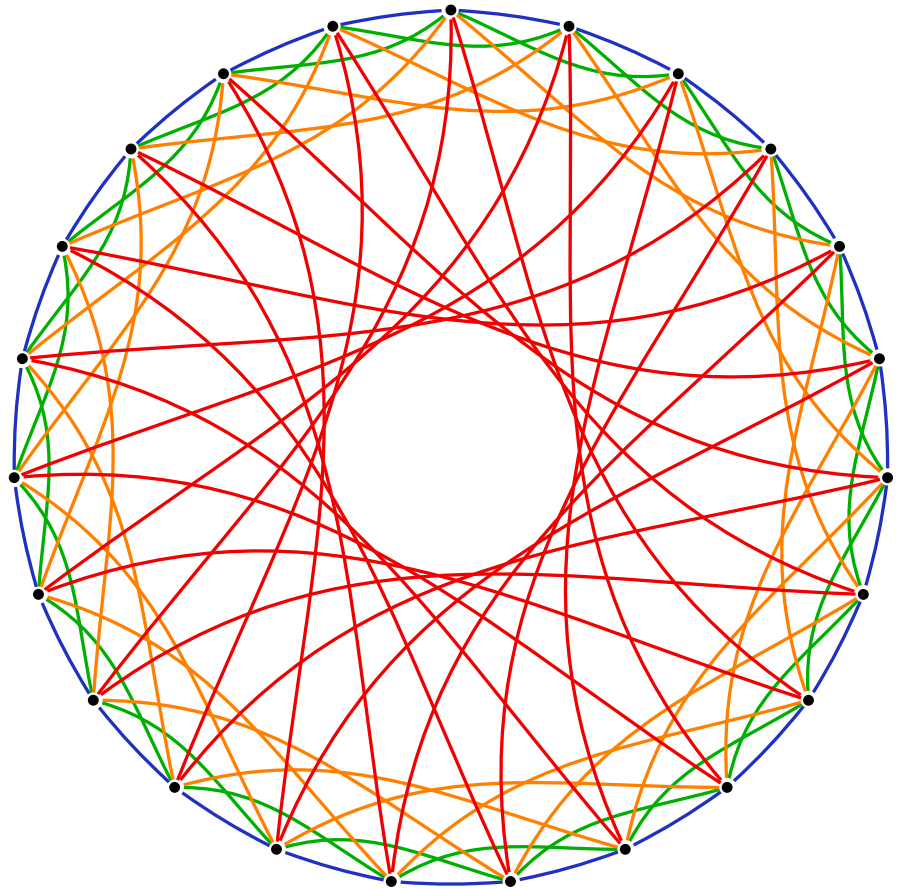
cycles


g^0
g^1
g^2
g^3
g^4
g^5
g^6
g^7
g^8
g^9
g^{10}
g^{11}
g^{12}
g^{13}
g^{14}
g^{15}
g^{16}
g^{17}
g^{18}
g^{19}
g^{20}
g^{21}
g^{22}

A union of cycles

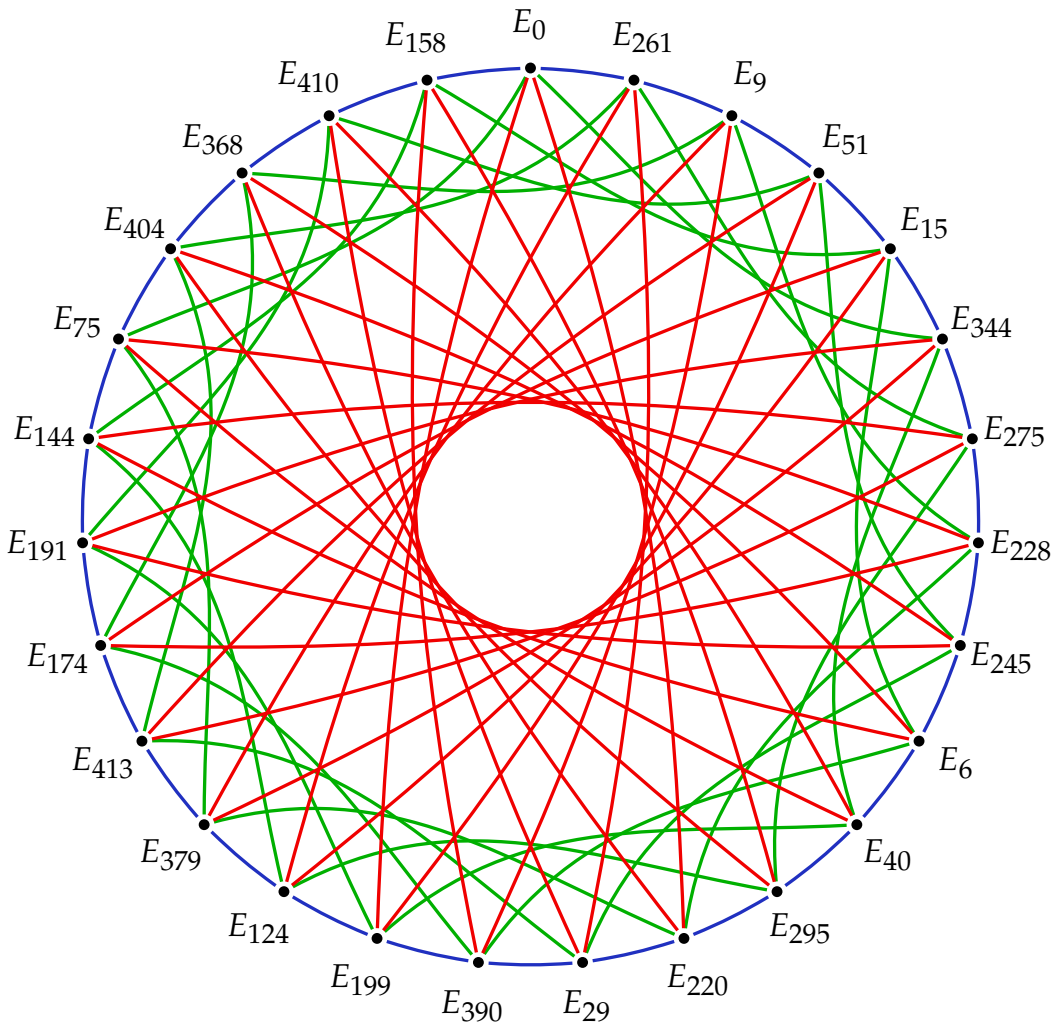


A union of cycles



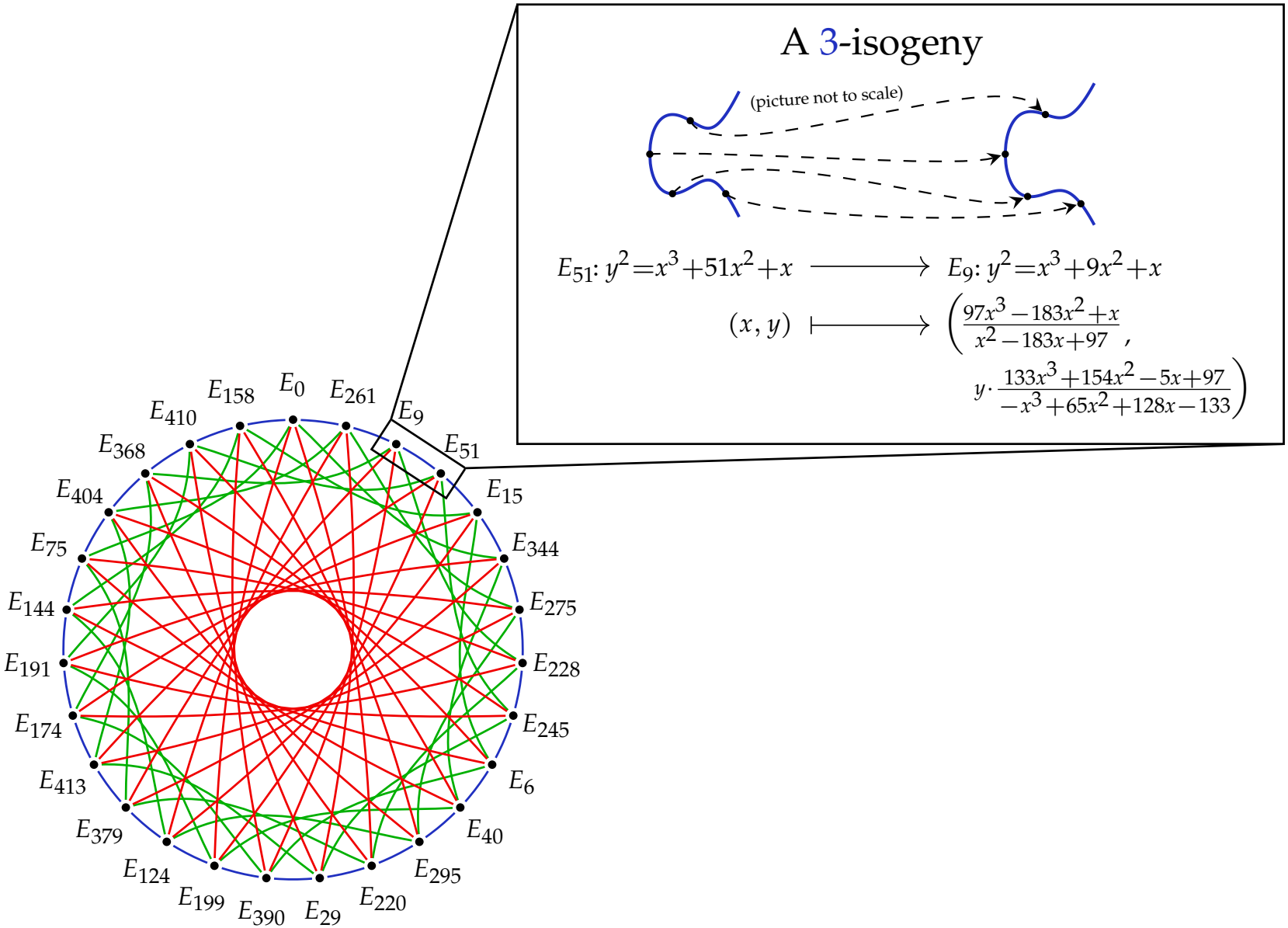
CSIDH: Nodes are now elliptic curves and edges are isogenies.

A union of cycles of elliptic curves



Nodes: Supersingular curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
Edges: 3-, 5-, and 7-isogenies (certain kinds of maps).

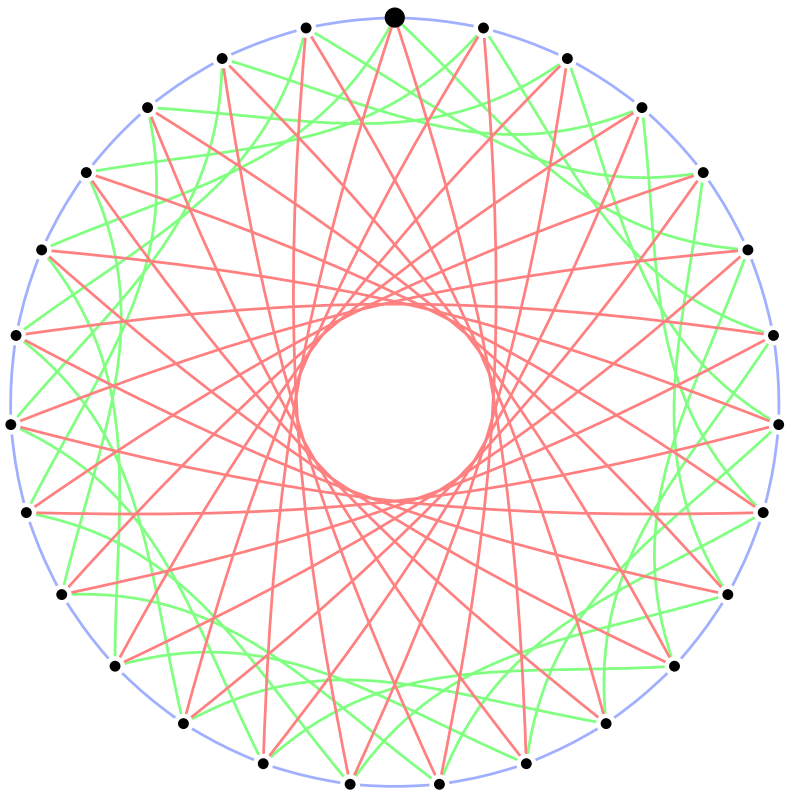
A union of cycles of elliptic curves and isogenies



Diffie–Hellman on graphs!

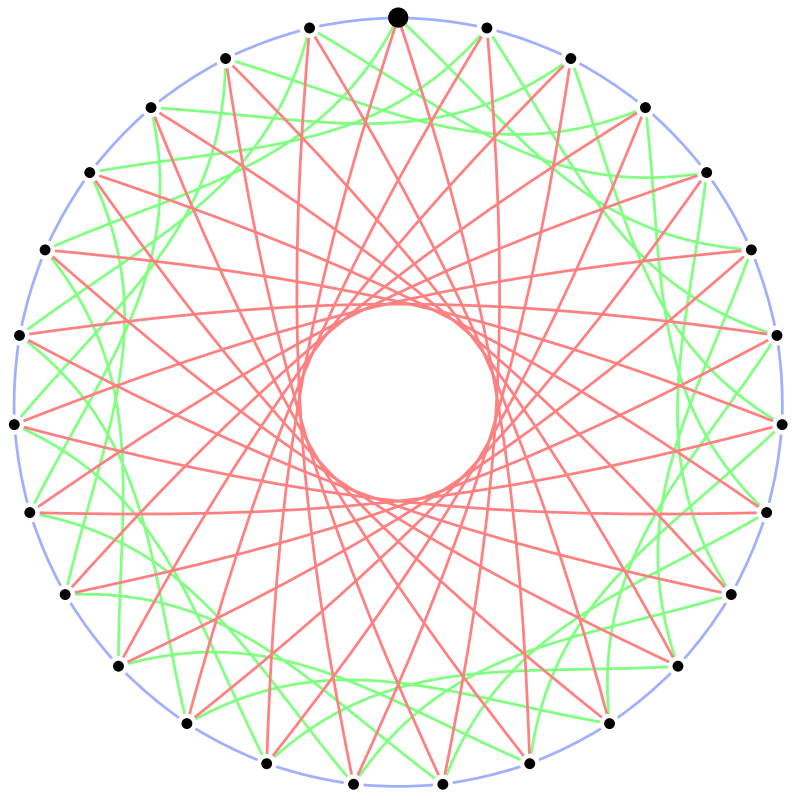
Alice

$[+, -, +, -]$



Bob

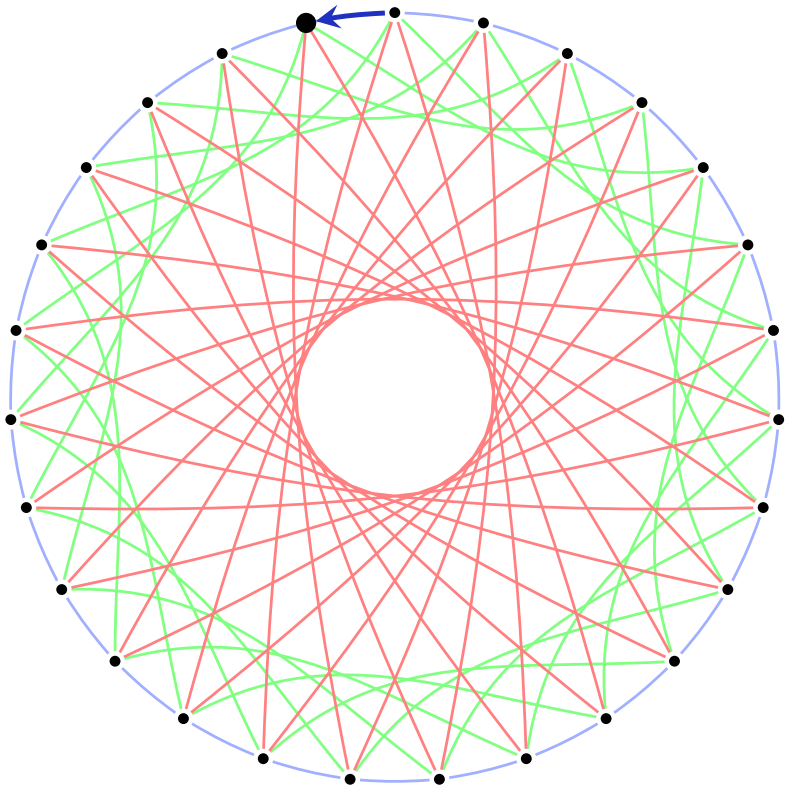
$[+, +, -, +]$



Diffie–Hellman on graphs!

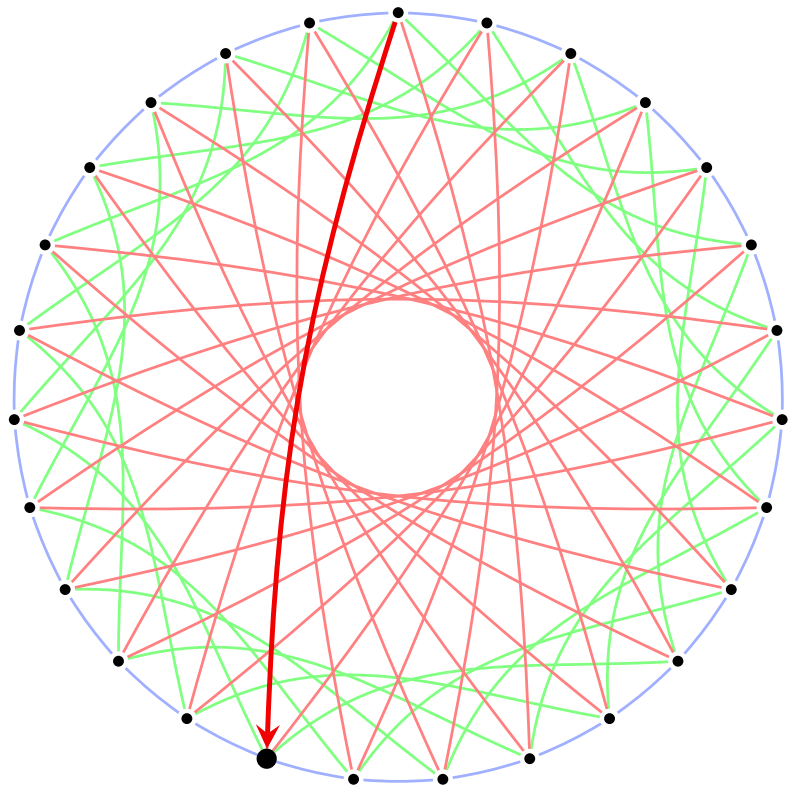
Alice

$[+, -, +, -]$



Bob

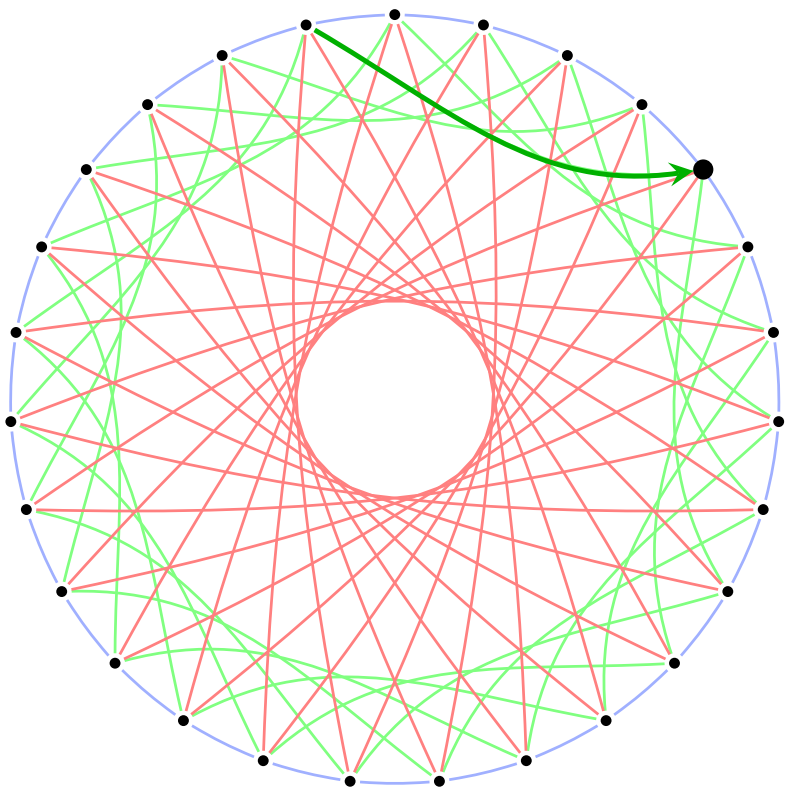
$[+, +, -, +]$



Diffie–Hellman on graphs!

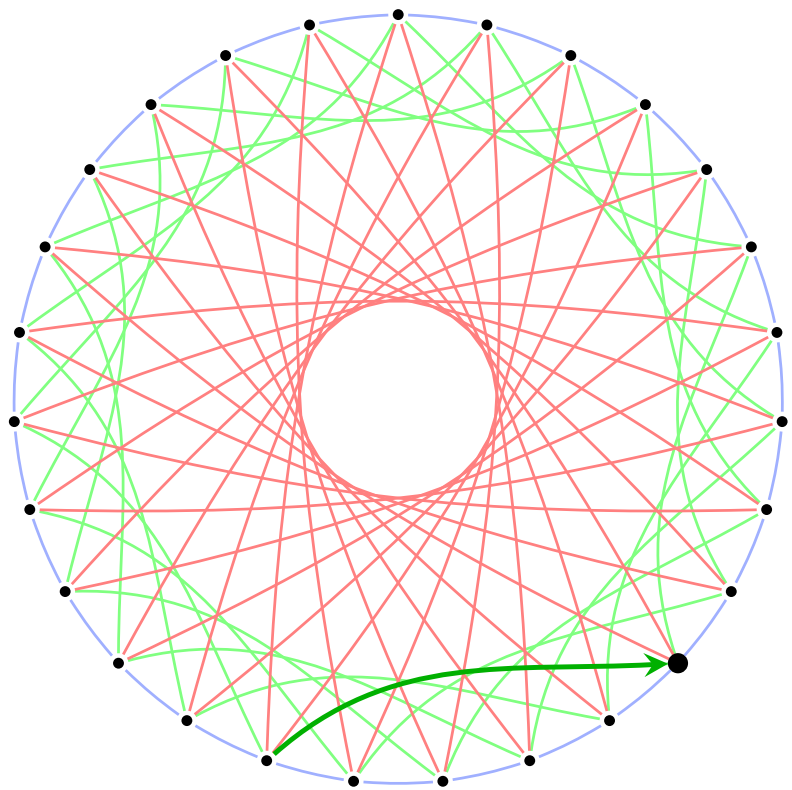
Alice

$[+, -, +, -]$



Bob

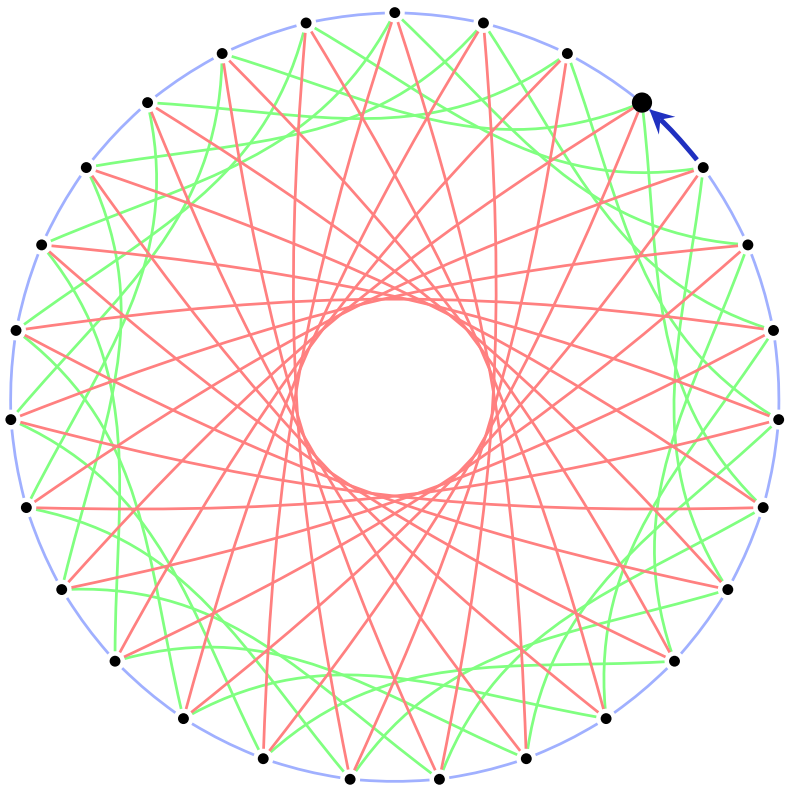
$[+, +, -, +]$



Diffie–Hellman on graphs!

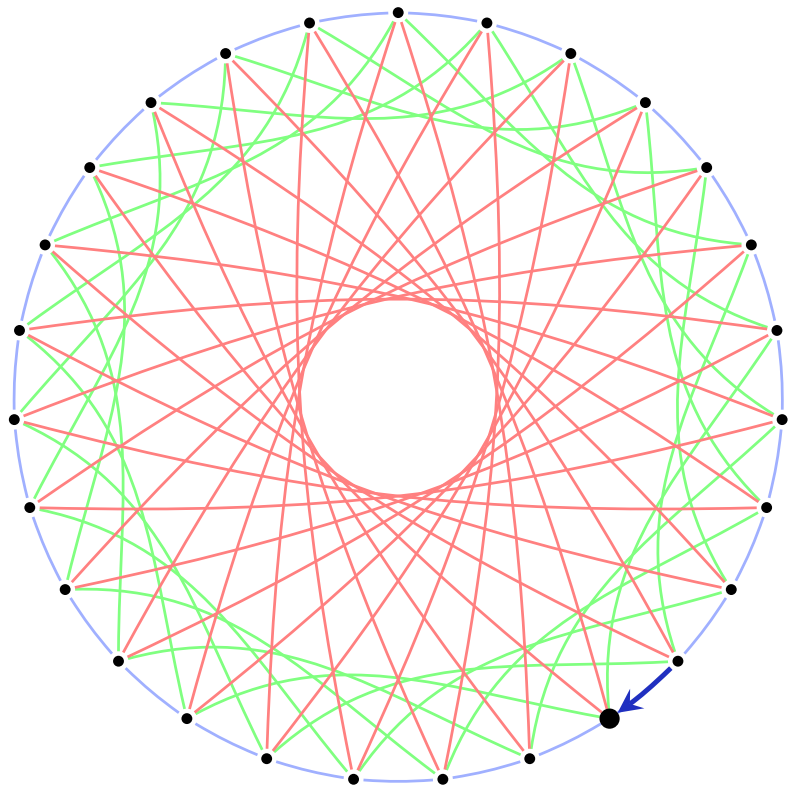
Alice

$[+, -, +, -]$



Bob

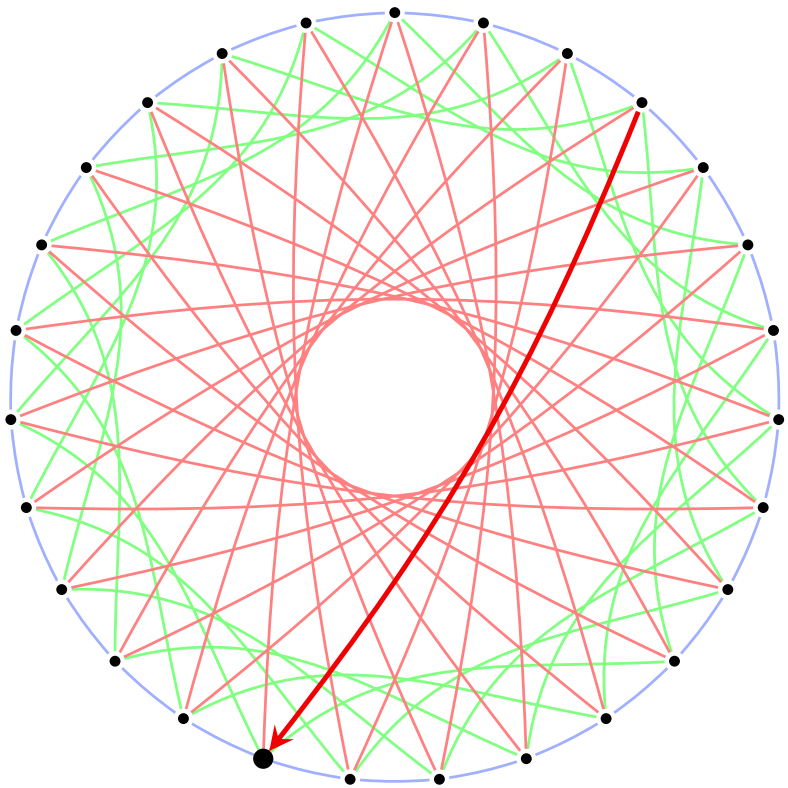
$[+, +, -, +]$



Diffie–Hellman on graphs!

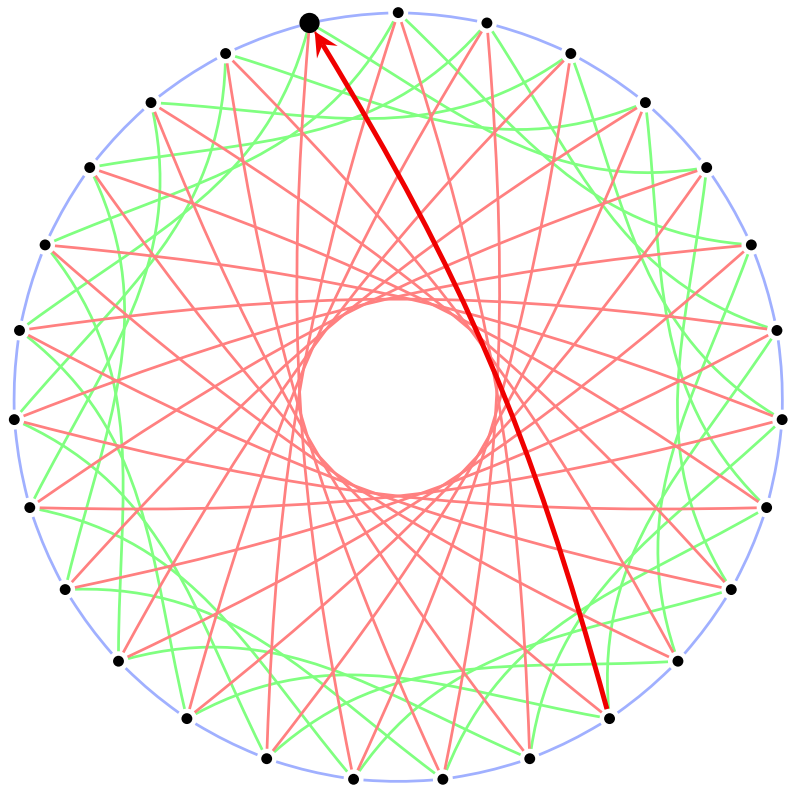
Alice

$[+, -, +, -]$



Bob

$[+, +, -, +]$



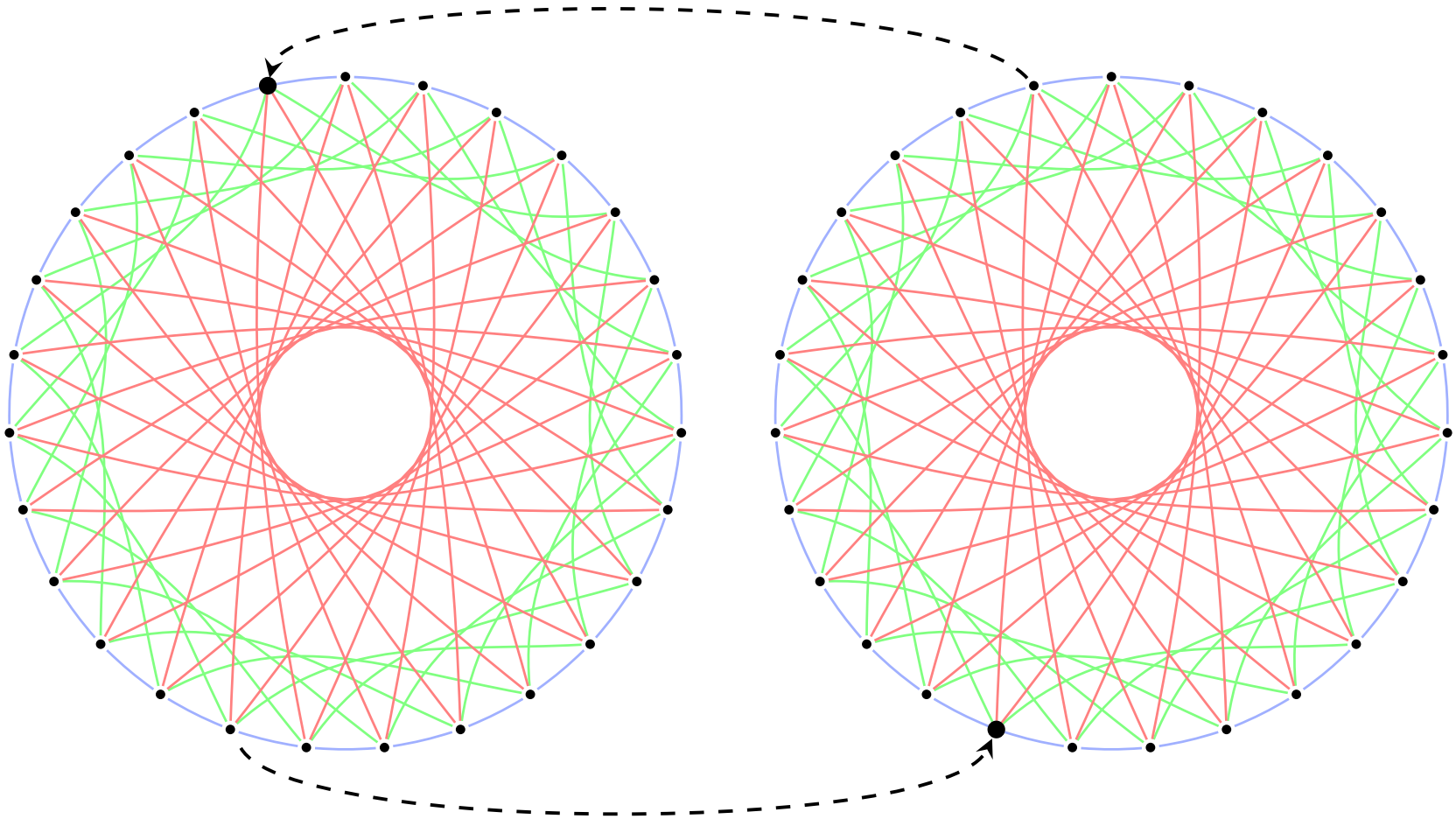
Diffie–Hellman on graphs!

Alice

$[+, -, +, -]$

Bob

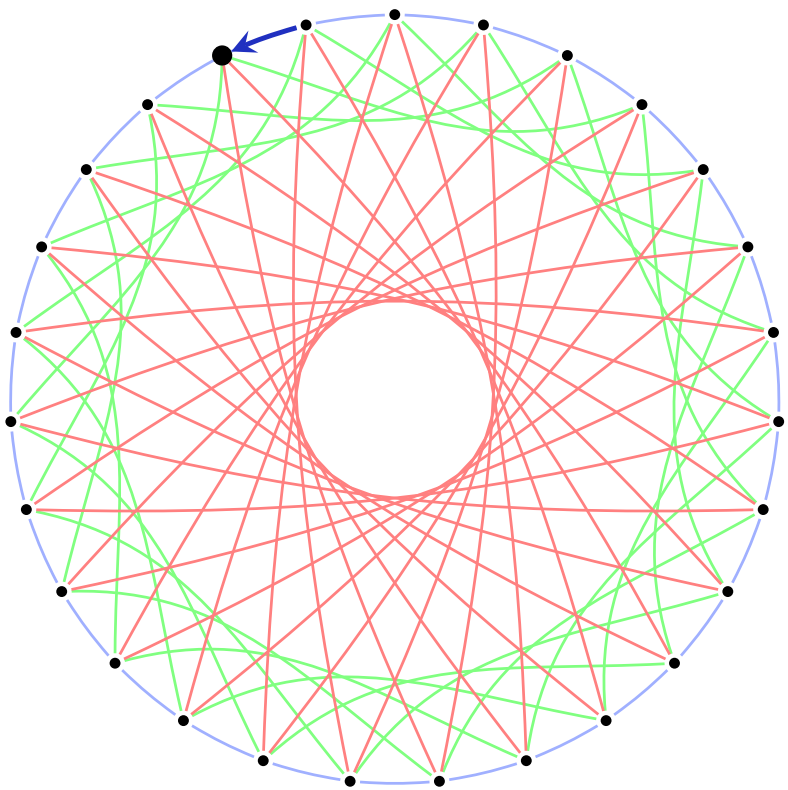
$[+, +, -, +]$



Diffie–Hellman on graphs!

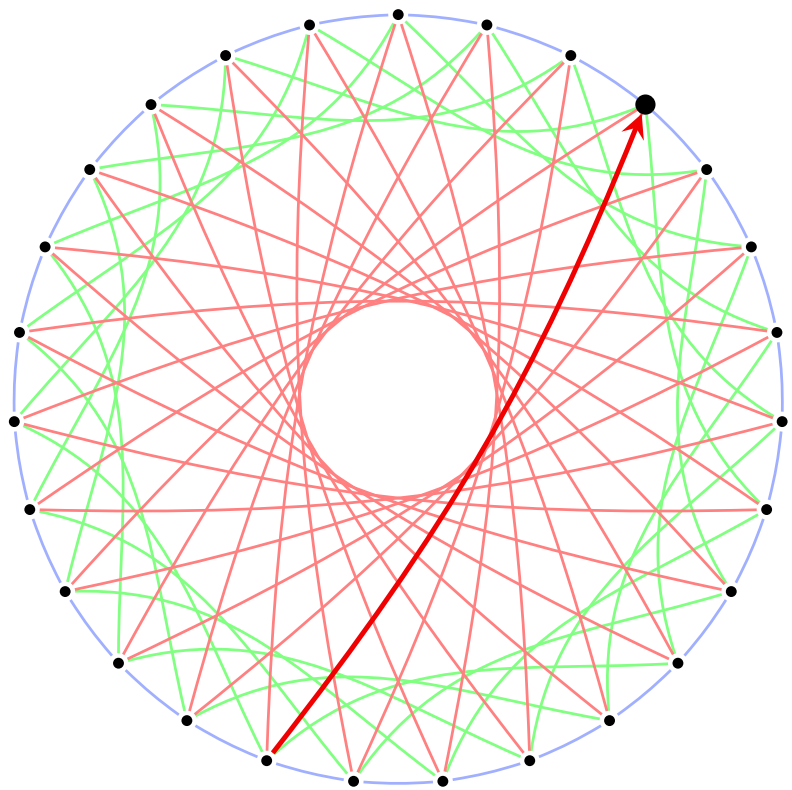
Alice

$[+, -, +, -]$



Bob

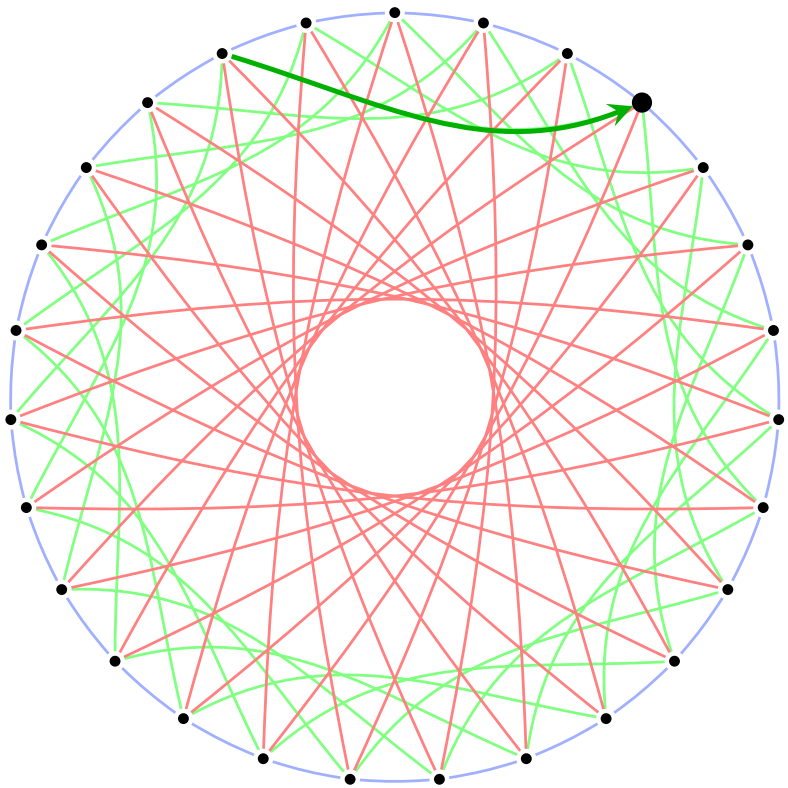
$[+, +, -, +]$



Diffie–Hellman on graphs!

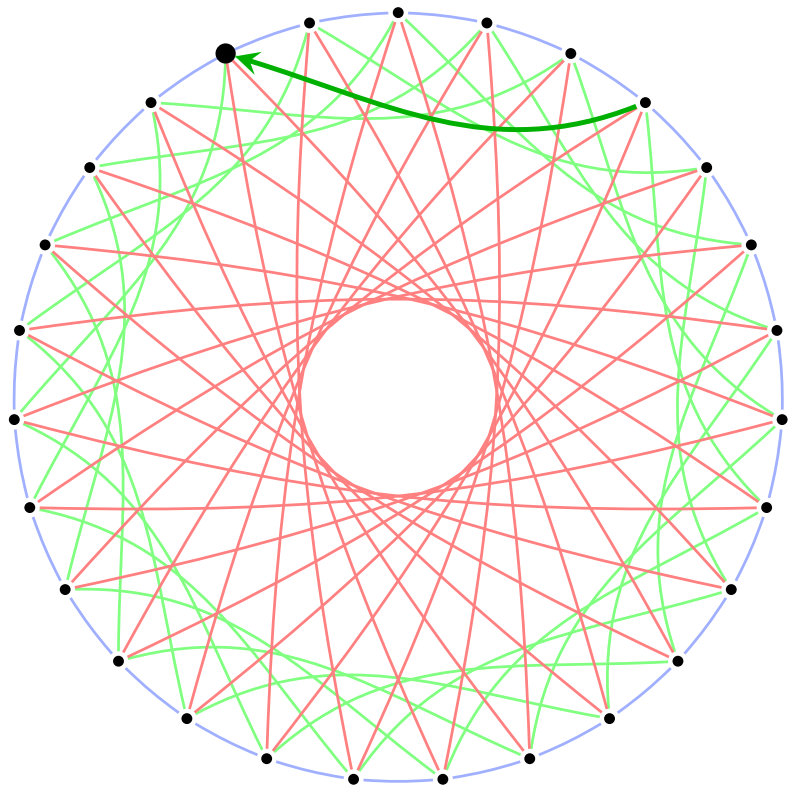
Alice

$[+, -, +, -]$



Bob

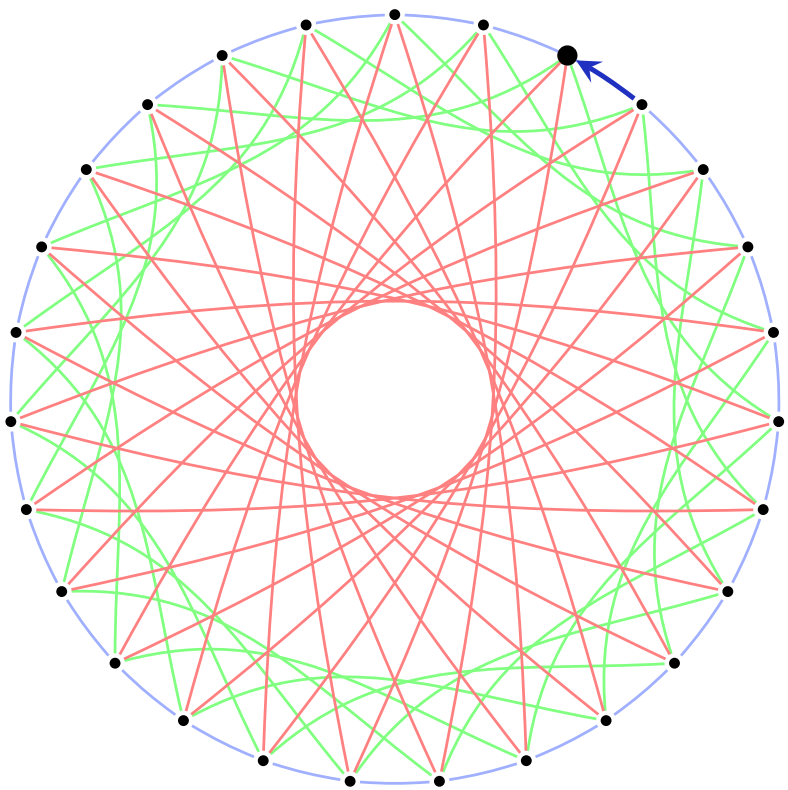
$[+, +, -, +]$



Diffie–Hellman on graphs!

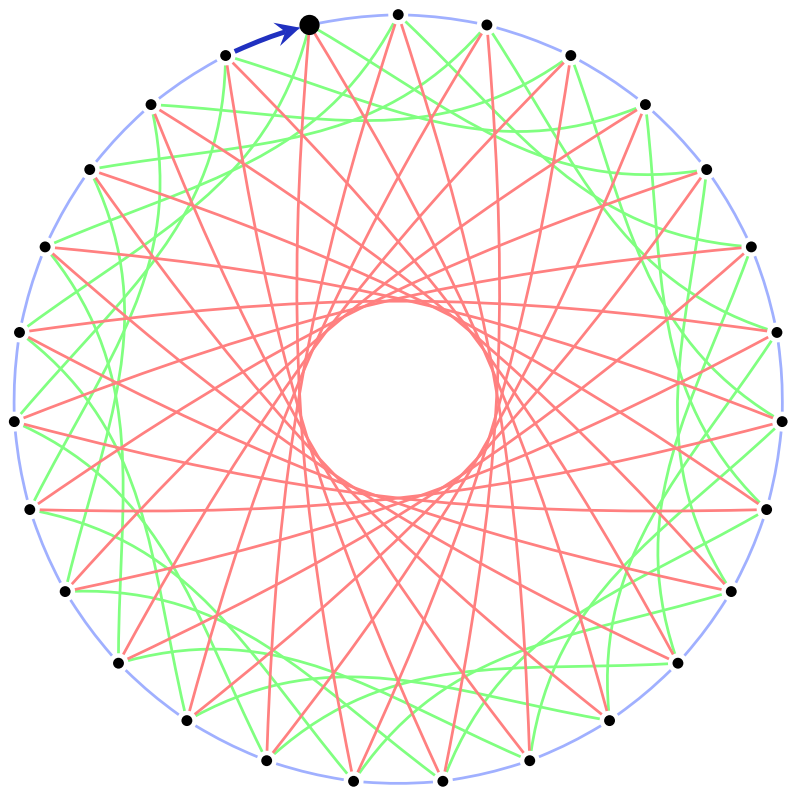
Alice

$[+, -, +, -]$



Bob

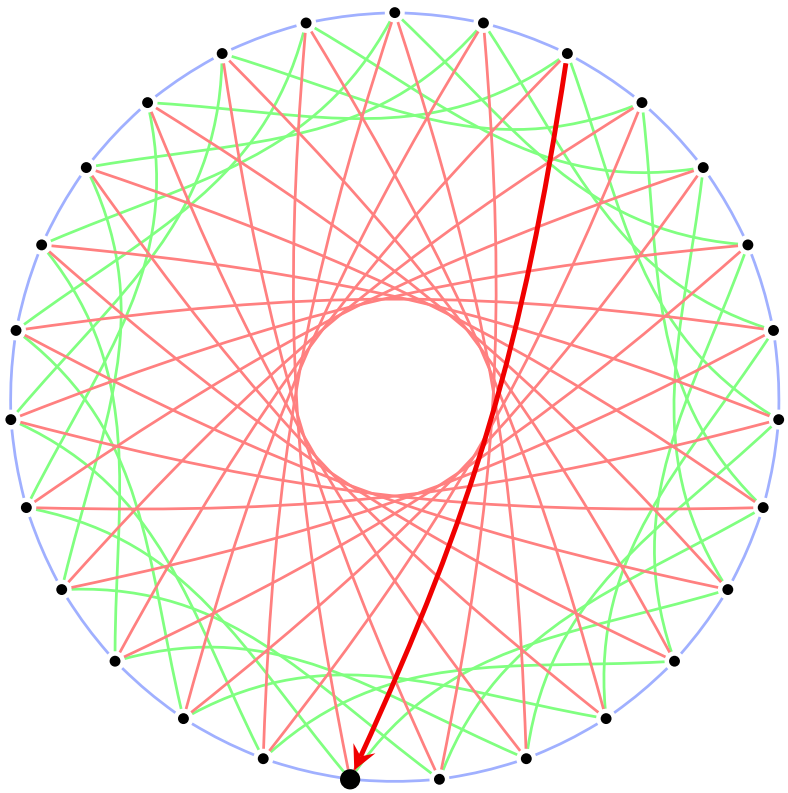
$[+, +, -, +]$



Diffie–Hellman on graphs!

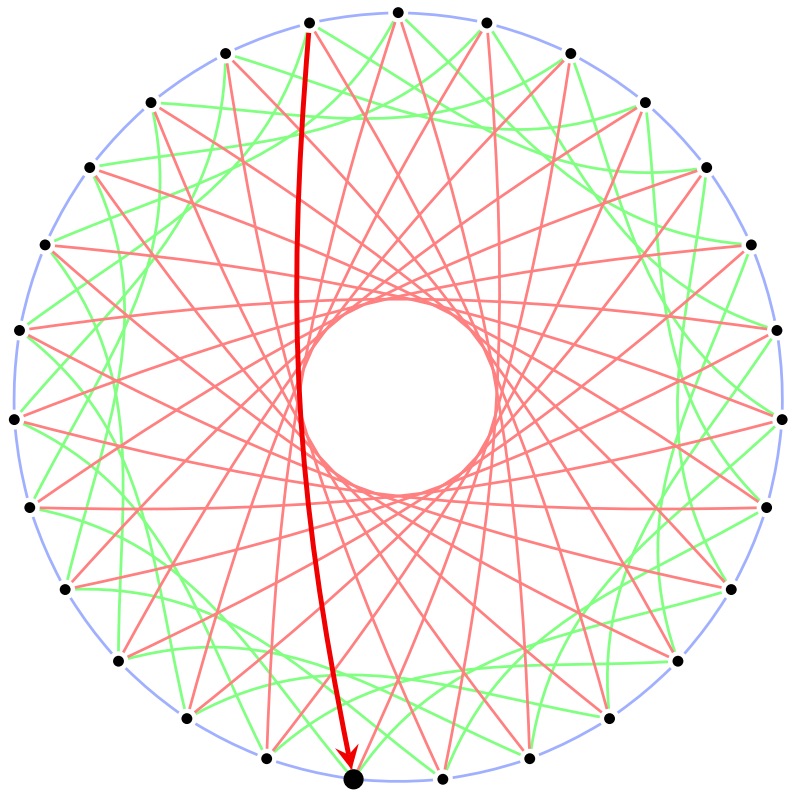
Alice

$[+, -, +, -]$



Bob

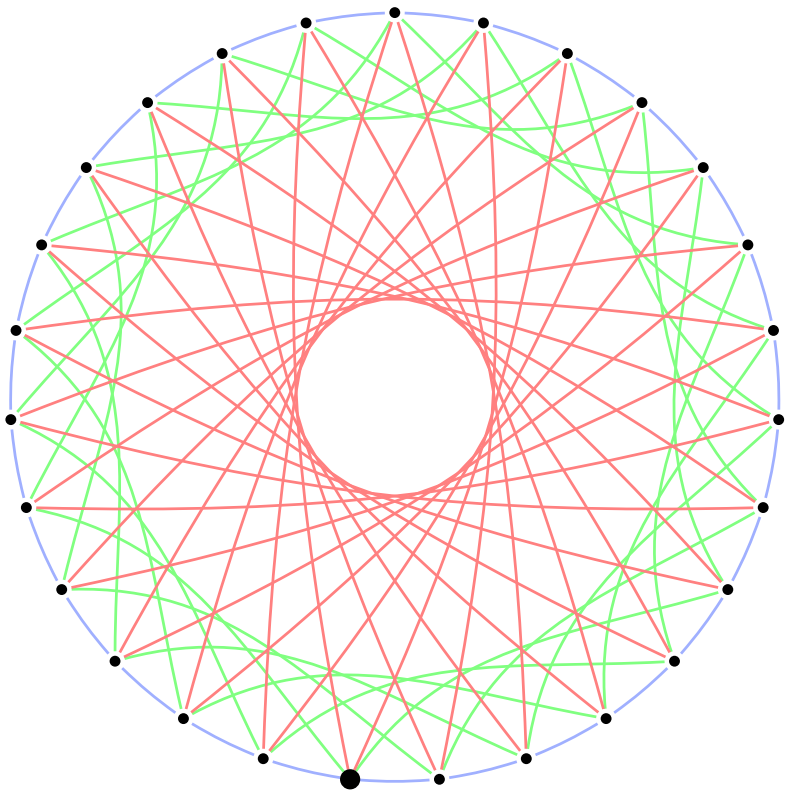
$[+, +, -, +]$



Diffie–Hellman on graphs!

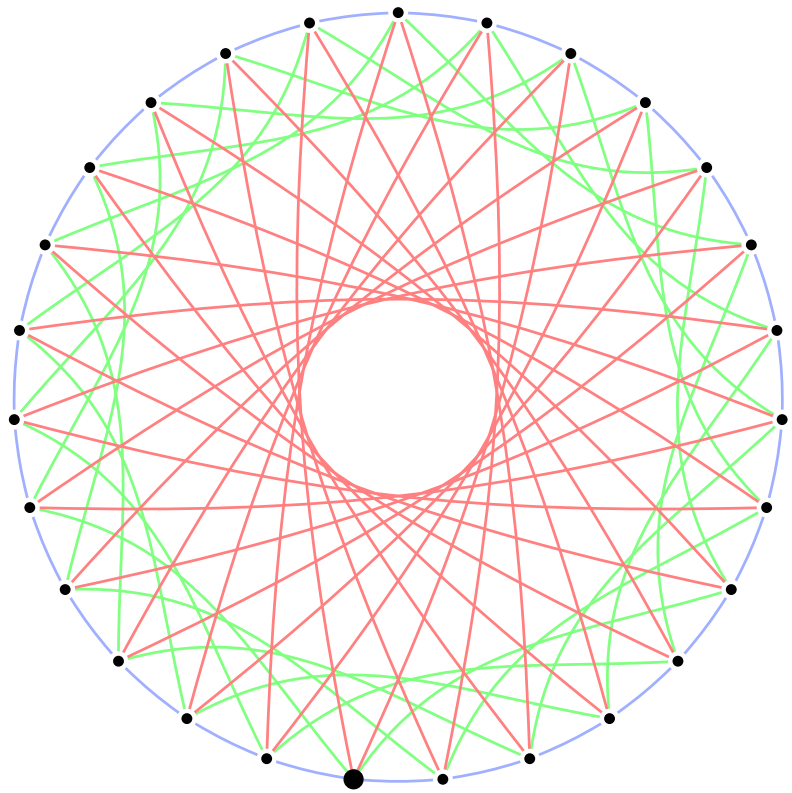
Alice

$[+, -, +, -]$



Bob

$[+, +, -, +]$



Welcome to the CSIDH!

- Our contribution:
  Huge speed-up by switching to supersingular curves.

Welcome to the CSIDH!

- Our contribution:
  Huge speed-up by switching to supersingular curves.
- 'An efficient post-quantum commutative group action'.

Welcome to the CSIDH!

- Our contribution:
  Huge speed-up by switching to supersingular curves.

- 'An efficient post-quantum commutative group action'.

$\Rightarrow$ Post-quantum non-interactive key exchange.
(Efficient public-key validation!)

- Public keys: 64 bytes (conjectured AES-128 security level).
- Full key exchange: ~ 80 ms. (Optimizations in progress!)

Welcome to the CSIDH!

- Our contribution:
  Huge speed-up by switching to supersingular curves.

- 'An efficient post-quantum commutative group action'.

$\Rightarrow$ Post-quantum non-interactive key exchange.
(Efficient public-key validation!)

- Public keys: 64 bytes (conjectured AES-128 security level).
- Full key exchange: ~ 80 ms. (Optimizations in progress!)

Thanks!