# Wallabies: classical attacks on Restricted Group Actions and high-dimensional MDLPs

Le Dévéhat Anaëlle

Inria, Institut Polytechnique de Paris

26 September, 2024

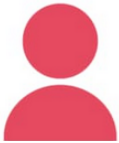
# Diffie-Hellman Key Exchange

Alice

$(a, g, p)$

$A = g^a \mod p$

$K = B^a \mod p$



Bob

$(b)$

$B = g^b \mod p$

$K = A^b \mod p$

## Security

- Diffie-Hellman's security relies on the discrete logarithm problem.
- Breaking it requires finding $x$ in $g^x \mod p$ when $g$, $p$, and $g^x \mod p$ are known.

# Diffie-Hellman Key Exchange

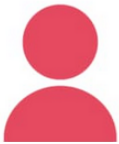
Alice

$(a, g, p)$

$A = g^a \mod p$

$K = B^a \mod p$

$\mathbf{A}, g, p$



Bob

$(b)$

$B = g^b \mod p$

$K = A^b \mod p$

## Security

- Diffie-Hellman's security relies on the discrete logarithm problem.
- Breaking it requires finding $x$ in $g^x \mod p$ when $g$, $p$, and $g^x \mod p$ are known.

# Diffie-Hellman Key Exchange

Alice

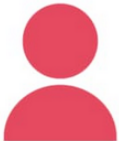
$(a, g, p)$

$A = g^a \mod p$

$K = B^a \mod p$

$\mathbf{A}, g, p$ →

← $B$



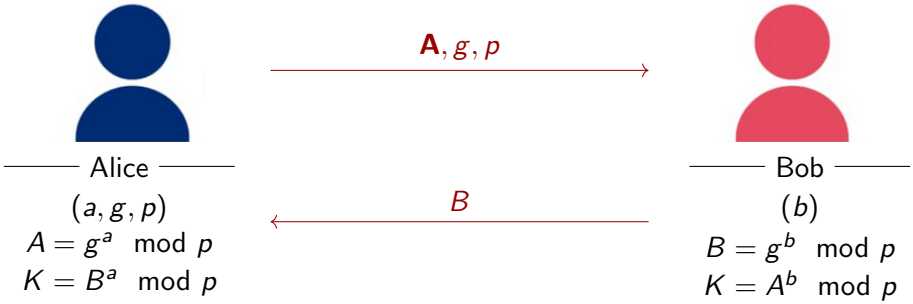
Bob

$(b)$

$B = g^b \mod p$

$K = A^b \mod p$

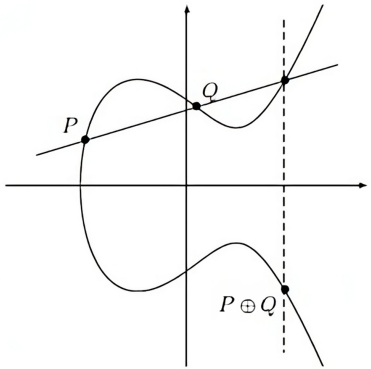
# Diffie-Hellman Key Exchange




## Security

- Diffie-Hellman's security relies on the discrete logarithm problem.
- Breaking it requires finding $x$ in $g^x \mod p$ when $g$, $p$, and $g^x \mod p$ are known.

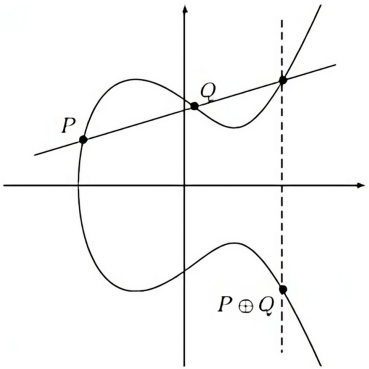
# Elliptic curves and isogenies




An *elliptic curve* can be written in Weierstrass form

$$E : y^2 = x^3 + ax + b.$$

# Elliptic curves and isogenies

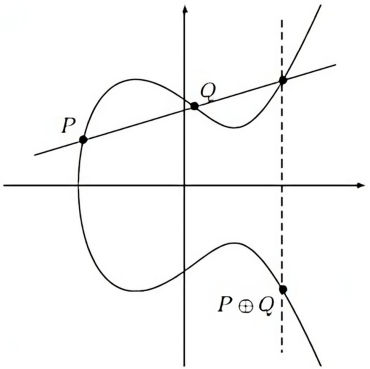



An *elliptic curve* can be written in Weierstrass form

$$E : y^2 = x^3 + ax + b.$$

Points on elliptic curves form a *group* under addition.

# Elliptic curves and isogenies




An *elliptic curve* can be written in Weierstrass form

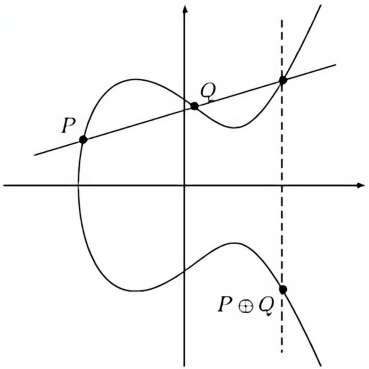
$$E : y^2 = x^3 + ax + b.$$

Points on elliptic curves form a *group* under addition.

Elliptic curves are *symmetric* about the x-axis, so

$$(x, y) \in E \implies (x, -y) \in E.$$

# Elliptic curves and isogenies




An *elliptic curve* can be written in Weierstrass form

$$E : y^2 = x^3 + ax + b.$$

Points on elliptic curves form a *group* under addition.

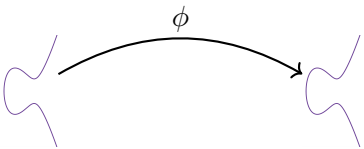
Elliptic curves are *symmetric* about the x-axis, so

$$(x, y) \in E \implies (x, -y) \in E.$$

Often we can restrict to using only x-coordinates.

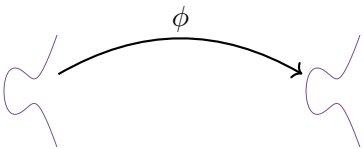
# Elliptic curves and isogenies

An *isogeny* is a rational mapping between two elliptic curves.



# Elliptic curves and isogenies

An *isogeny* is a rational mapping between two elliptic curves.



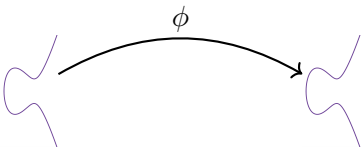

Example :

- Source curve $\mathcal{E}_1$, with equation $y^2 = x^3 + 1$
- Target curve $\mathcal{E}_2$, with equation $y^2 = x^3 + 8$

$$\phi(x, y) = \left( \frac{y^2}{x^2}, \frac{y(x^3+1)}{x^3} \right)$$

# Elliptic curves and isogenies

An *isogeny* is a rational mapping between two elliptic curves.



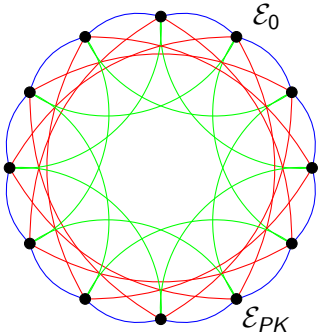

Example :

- Source curve $\mathcal{E}_1$, with equation $y^2 = x^3 + 1$
- Target curve $\mathcal{E}_2$, with equation $y^2 = x^3 + 8$

$$\phi(x, y) = \left( \frac{y^2}{x^2}, \frac{y(x^3+1)}{x^3} \right)$$

It is believed to be classically and quantumly hard to find an isogeny between two fixed elliptic curves.

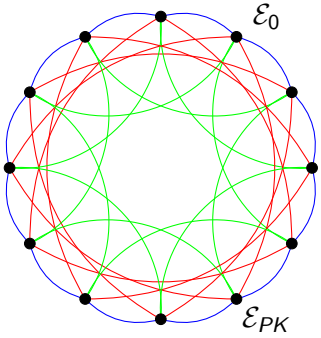
# Commuative Supersingular Isogeny Diffie–Hellman (CSIDH-2018)




## Group action

We take $G$ as being the set of $j$-invariants of supersingular elliptic curves up to $\mathbb{F}_p$-isomorphism with endomorphism ring an order $\mathcal{O}$ of an imaginary quadratic field $K$. The ideal-class group $cl(\mathcal{O})$ acts freely and transitively on $G$ through isogenies : $cl(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O})$

# Commuative Supersingular Isogeny Diffie–Hellman (CSIDH-2018)



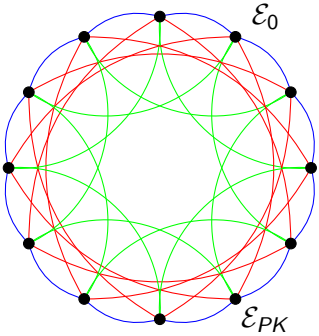

## Group action

We take $G$ as being the set of $j$-invariants of supersingular elliptic curves up to $\mathbb{F}_p$-isomorphism with endomorphism ring an order $\mathcal{O}$ of an imaginary quadratic field $K$. The ideal-class group $cl(\mathcal{O})$ acts freely and transitively on $G$ through isogenies : $cl(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O})$

A private key $(e_1, \ldots, e_n)$ corresponds to

1. $e_1$ steps in the $\ell_1$-isogeny graph, then

# Commuative Supersingular Isogeny Diffie–Hellman (CSIDH-2018)



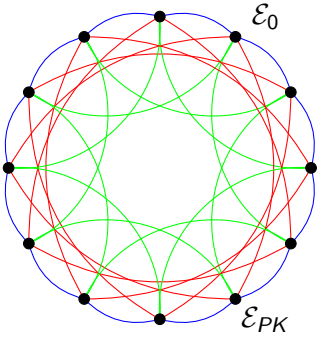

## Group action

We take $G$ as being the set of $j$-invariants of supersingular elliptic curves up to $\mathbb{F}_p$-isomorphism with endomorphism ring an order $\mathcal{O}$ of an imaginary quadratic field $K$. The ideal-class group $cl(\mathcal{O})$ acts freely and transitively on $G$ through isogenies : $cl(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O})$

A private key $(e_1, \ldots, e_n)$ corresponds to

1. $e_1$ steps in the $\ell_1$-isogeny graph, then
2. $e_2$ steps in the $\ell_2$-isogeny graph, then

# Commuative Supersingular Isogeny Diffie–Hellman (CSIDH-2018)



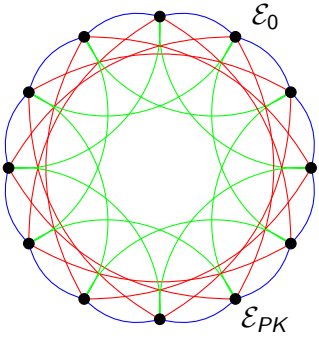

## Group action

We take $G$ as being the set of $j$-invariants of supersingular elliptic curves up to $\mathbb{F}_p$-isomorphism with endomorphism ring an order $\mathcal{O}$ of an imaginary quadratic field $K$. The ideal-class group $cl(\mathcal{O})$ acts freely and transitively on $G$ through isogenies : $cl(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O})$

A private key $(e_1, \ldots, e_n)$ corresponds to

1. $e_1$ steps in the $\ell_1$-isogeny graph, then
2. $e_2$ steps in the $\ell_2$-isogeny graph, then
3. $e_3$ steps in the $\ell_3$-isogeny graph,

# Commuative Supersingular Isogeny Diffie–Hellman (CSIDH-2018)



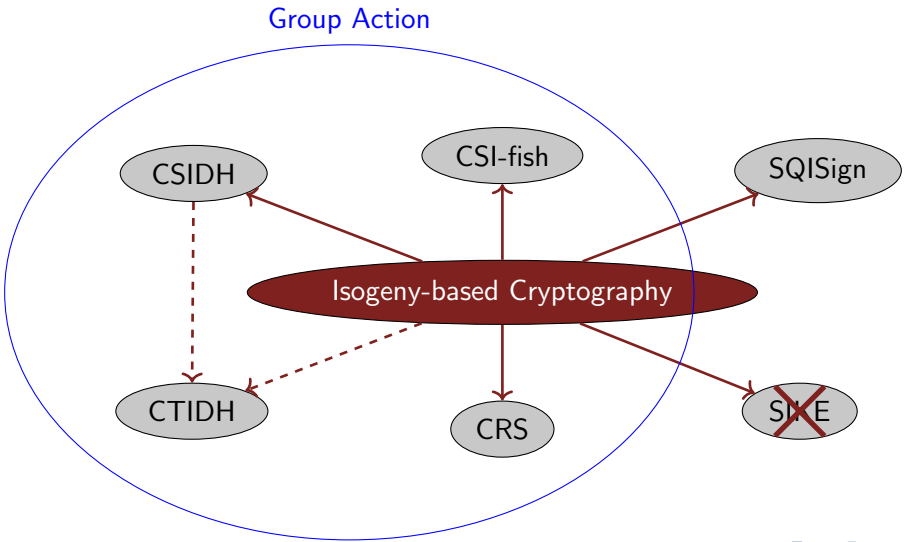

## Group action

We take $G$ as being the set of $j$-invariants of supersingular elliptic curves up to $\mathbb{F}_p$-isomorphism with endomorphism ring an order $\mathcal{O}$ of an imaginary quadratic field $K$. The ideal-class group $cl(\mathcal{O})$ acts freely and transitively on $G$ through isogenies : $cl(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O})$

A private key $(e_1, \ldots, e_n)$ corresponds to

1. $e_1$ steps in the $\ell_1$-isogeny graph, then
2. $e_2$ steps in the $\ell_2$-isogeny graph, then
3. $e_3$ steps in the $\ell_3$-isogeny graph,
4. More walks ...
5. $\mathcal{E}_{PK} = [\mathfrak{a}] * \mathcal{E}_0$ where $[\mathfrak{a}] \in cl(\mathcal{O})$

# Isogeny-based Cryptosystems Mindmap



# Key recovery attacks on post-quantum group actions

## Definition (Multidimensional Discrete Logarithm problem)

We consider $G$ an abelian group. We are given $\{P_i\}_{1\geq i\geq n}$, $Q$ points in $G$ and some bounds $\{N_i\}_{1\geq i\geq n} \in \mathbb{N}^n$.
Find an exponent vector $(k_1, k_2, \cdots, k_n) \in [\![-N_i; N_i]\!]^n$ such that

$$Q = [k_1]P_1 + [k_2]P_2 + \ldots + [k_n]P_n.$$

*Remark.* Note that in the MDLP problem as we consider it, the exponents are bounded by a fixed integer. This leads to different kind of attacks.

# Key recovery attacks on post-quantum group actions

**Definition (Multidimensional Discrete Logarithm problem)**

We consider $G$ an abelian group. We are given $\{P_i\}_{1\geq i\geq n}$, $Q$ points in $G$ and some bounds $\{N_i\}_{1\geq i\geq n} \in \mathbb{N}^n$.
Find an exponent vector $(k_1, k_2, \cdots, k_n) \in [\![-N_i; N_i]\!]^n$ such that

$$Q = [k_1]P_1 + [k_2]P_2 + \ldots + [k_n]P_n.$$

*Remark.* Note that in the MDLP problem as we consider it, the exponents are bounded by a fixed integer. This leads to different kind of attacks.

Does this look familiar? Recall that in *CSIDH* cryptosystem:

$$\mathcal{E}_{PK} = [\mathfrak{l}_1]^{k_1} \ldots [\mathfrak{l}_n]^{k_n} * \mathcal{E}_0$$

# Key recovery attacks on CSIDH

With the right definitions, this MDLP framework fits different isogeny-based cryptosystems using group actions. With CSIDH, we are looking to break the MDLP for :

- $n = 74$
- $N_i \simeq 5$ for $1 \leq i \leq n$
- $G$ is the group of supersingular Elliptic curves obtained from applying a set of isogenies on $\mathcal{E}_0$ over $F_p$

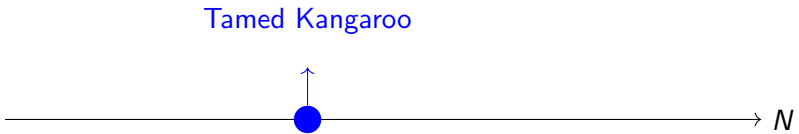
$$Q = \mathcal{E}_{PK} = [\mathfrak{l}_1^{k_1} \dots \mathfrak{l}_n^{k_n}] * E_0$$
$$\mathcal{E}_{PK} = ([\mathfrak{l}_1]^{k_1} \times \cdots \times [\mathfrak{l}_n]^{k_n}) * E_0$$

# The Kangaroo method - Dimension 1

Wild Kangaroo : $Q + [a_i]P_1 = [k_1 + a_i]P_1$
Tamed Kangaroo : $[b_i]P_1$

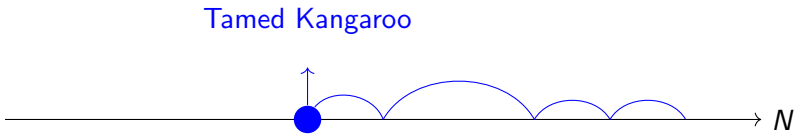
# The Kangaroo method - Dimension 1

Wild Kangaroo : $Q + [a_i]P_1 = [k_1 + a_i]P_1$
Tamed Kangaroo : $[b_i]P_1$



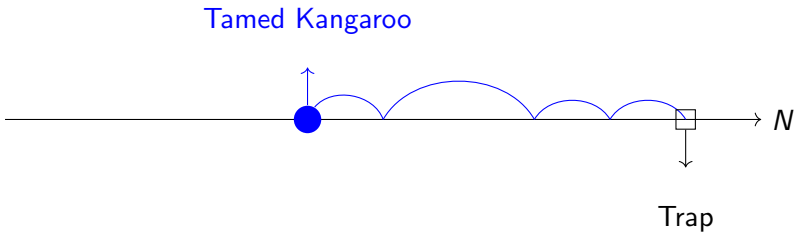
# The Kangaroo method - Dimension 1

Wild Kangaroo : $Q + [a_i]P_1 = [k_1 + a_i]P_1$
Tamed Kangaroo : $[b_i]P_1$



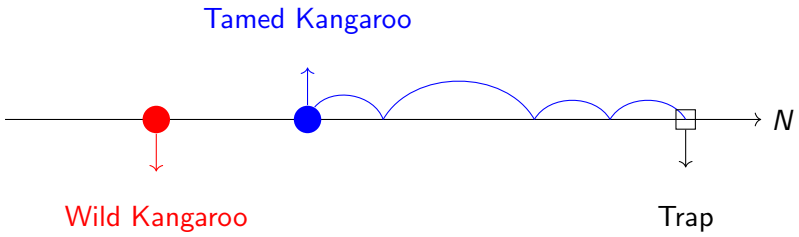
# The Kangaroo method - Dimension 1

Wild Kangaroo : $Q + [a_i]P_1 = [k_1 + a_i]P_1$
Tamed Kangaroo : $[b_i]P_1$



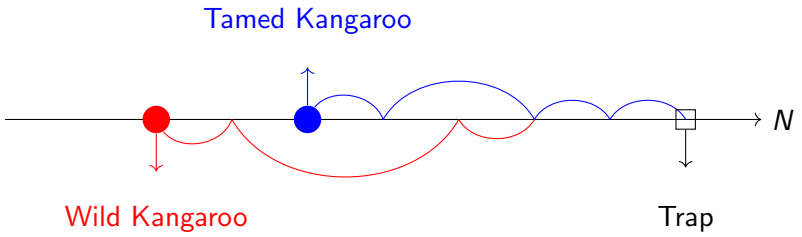
# The Kangaroo method - Dimension 1

Wild Kangaroo : $Q + [a_i]P_1 = [k_1 + a_i]P_1$
Tamed Kangaroo : $[b_i]P_1$



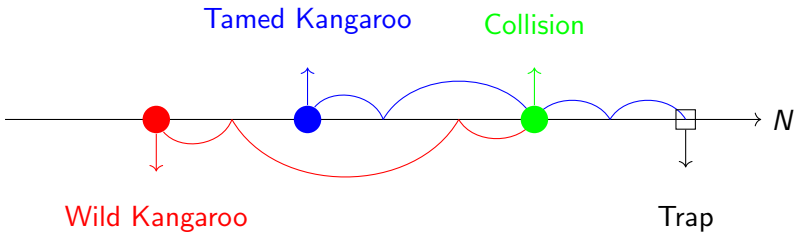
# The Kangaroo method - Dimension 1

Wild Kangaroo : $Q + [a_i]P_1 = [k_1 + a_i]P_1$
Tamed Kangaroo : $[b_i]P_1$



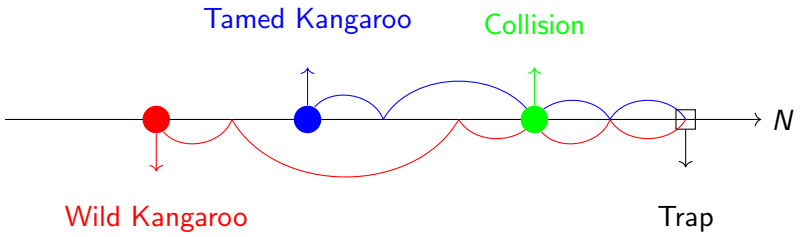
# The Kangaroo method - Dimension 1

Wild Kangaroo : $Q + [a_i]P_1 = [k_1 + a_i]P_1$
Tamed Kangaroo : $[b_i]P_1$



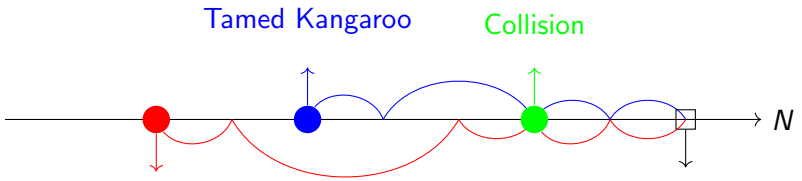
# The Kangaroo method - Dimension 1

Wild Kangaroo : $Q + [a_i]P_1 = [k_1 + a_i]P_1$
Tamed Kangaroo : $[b_i]P_1$



# The Kangaroo method - Dimension 1

Wild Kangaroo : $Q + [a_i]P_1 = [k_1 + a_i]P_1$
Tamed Kangaroo : $[b_i]P_1$



# Previous attacks on MDLP

Table: Summary of Attacks on Multidimensional Discrete Logarithm Problem

| Dimension | Attacks | Average-case Complexity |
|---|---|---|
| 1D | Van Oorschot & Wiener | $2\sqrt{N}$ |
| | Original Gaudry-Schost | $2.08\sqrt{N}$ |
| | Improved Gaudry-Schost | $2.05\sqrt{N}$ |
| | Pollard 4 kangaroo | $\leq 1.79\sqrt{N}$ |
| 2D | Original Gaudry-Schost | $2.45\sqrt{N}$ |
| | Improved Gaudry-Schost | $2.38\sqrt{N}$ |
| 3D & 4D | Original Gaudry-Schost | $2^{n/2}\sqrt{\pi N}$ |
| | Improved Gaudry-Schost | $\frac{2^n}{3^{n/2}}\sqrt{\pi N}$ |
| D $\geq 5$ | Gaudry-Schost | Ineffective |

$$N = \prod_{i=1}^{n}(2N_i + 1)$$

# Distinguished points - Low memory attack

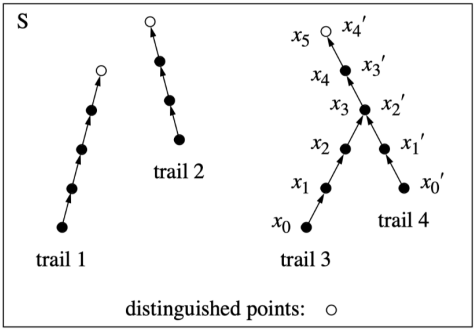
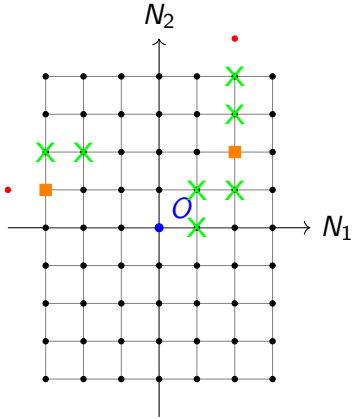



Figure: Schematic representation of distinguished points.

- $\theta$ : probability of a distinguished point
- Only distinguished points are saved into memory
- Constant storage : $\theta = \frac{c}{\sqrt{N}}$ for some constant $c$

# Limitations of GS Algorithm - Constant storage




- $(i, j)$-coordinates : $P = [i]P_1 + [j]P_2$
- A Kangaroo jump
- Distinguished point
- Unaccepted step $\rightarrow$ Restart

The mean length of a walk before hitting a DP is $\frac{\sqrt{N}}{c}$ : bigger than $\mathbb{E}(N_i)$ in most cases after $n >= 5$

Failure of Tame-Wild Birthday paradox analysis

Figure: Two-dimensional lattice centered on origin O, bounded by $N_1$ and $N_2$

# Limitations of GS Algorithm - High probability of restarting in high dimension

We consider a hypercube of bounds equal to 5. Let's take some random starting point for a walk in this space (np.random.choice(5,n)).

$n = 2$

$(2, 3)$
$(4, 1)$
$(1, 0)$
$(2, 2)$

$n = 6$

$(4, 2, 3, 3, 1, 3)$
$(4, 2, 3, 3, 1, 3)$
$(0, 0, 1, 1, 1, 0)$
$(3, 2, 0, 4, 2, 2)$

# Limitations of GS Algorithm - High probability of restarting in high dimension

We consider a hypercube of bounds equal to 5. Let's take some random starting point for a walk in this space (np.random.choice(5,d)).

$d = 2$

$(2, 3)$
$(4, 1)$
$(1, 0)$
$(2, 2)$

$d = 6$

$(4, 2, 3, 3, 1, 3)$
$(4, 2, 3, 3, 1, 3)$
$(0, 0, 1, 1, 1, 0)$
$(3, 2, 0, 4, 2, 2)$

# Wallabies - Special Points

Special Point : When a Wallaby lands on a special point, it restarts as *Wild* or *Tame* pseudo randomly ! (Probability $\beta$) : We introduce more restarts to have more random looking walks $\rightarrow$ A collision on a walk without distinguished points is still detected

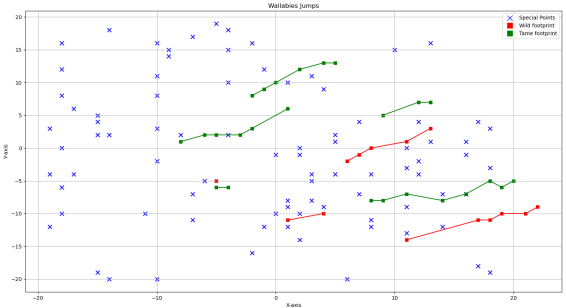



Figure: Schematic representation of a Wallaby's journey.

# A new search space

We distinguish between :

- The secret key space $[\![-N_i; N_i]\!]_{i=1}^{n}$
- The search space $\mathcal{N}$

# A new search space

We distinguish between :

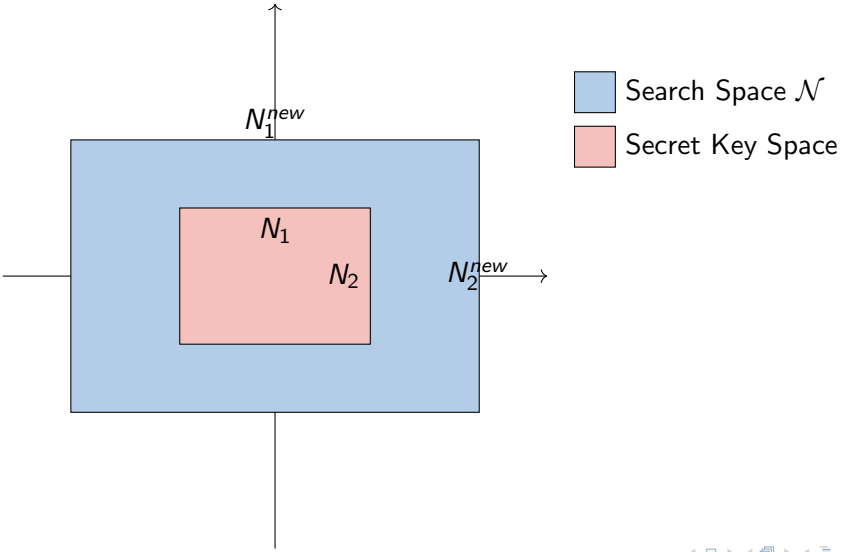
- The secret key space $[\![-N_i; N_i]\!]_{i=1}^n$
- The search space $\mathcal{N}$

**Fixed** $\beta$

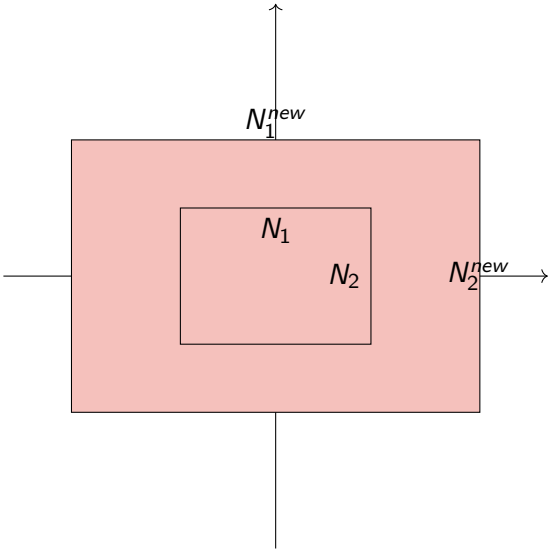
$\rightarrow$ A Wallaby reaches $-1/\sqrt{\beta}$ or $1/\sqrt{\beta}$ in $1/\beta$ steps on average.

$$\mathcal{N} = \left[-N_i - \frac{1}{\sqrt{\beta}}; N_i + \frac{1}{\sqrt{\beta}}\right]_{i=1}^{n} \cap \mathbb{Z}^n$$

# Visual representaion

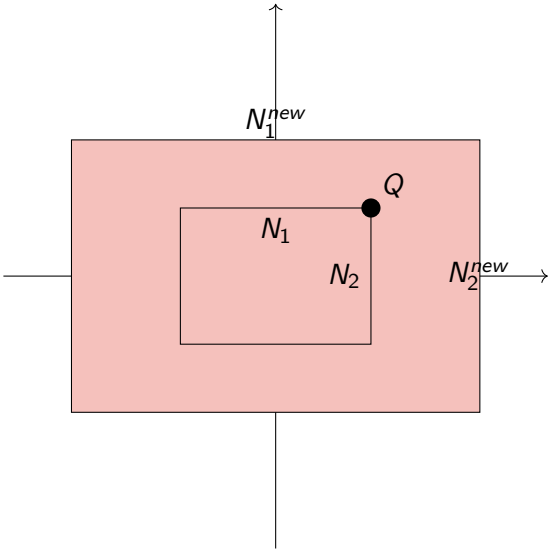


# Visual representaion



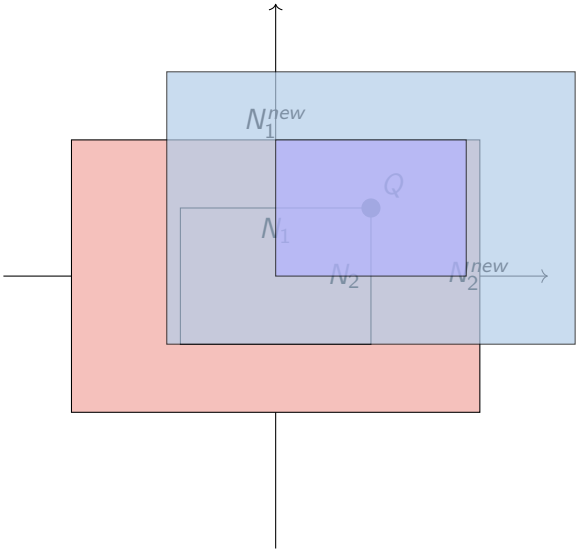

Tame Wallaby area

# Visual representaion




Tame Wallaby area

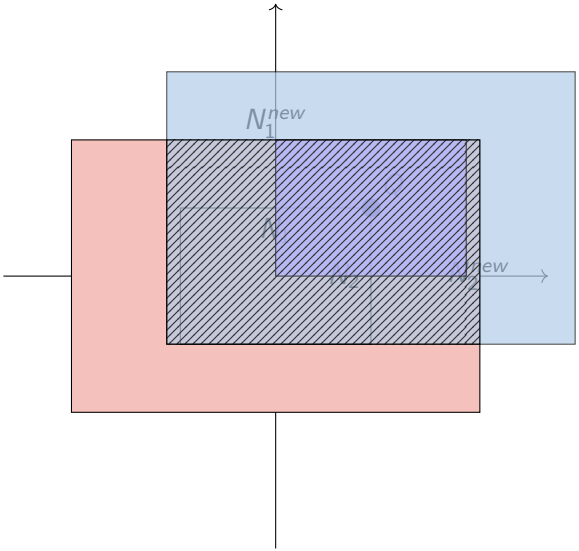
# Visual representaion




Wild Wallaby area

Tame Wallaby area

# Visual representaion




Wild Wallaby area

Tame Wallaby area

Collision zone

# An example run

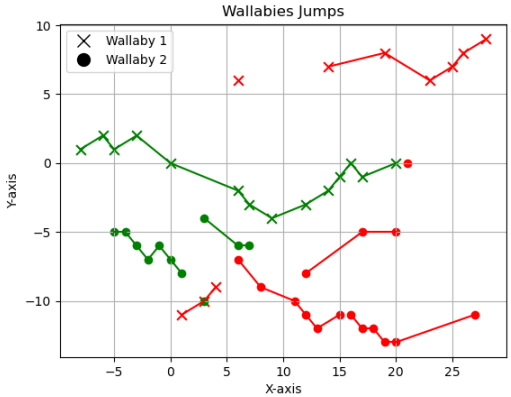



Figure: Collision between Wallabies.

# Theoretical results- Framework

$$\mathcal{N} := [-N_1 - \frac{1}{\beta}, N_1 + \frac{1}{\beta}] \times \cdots \times [-N_n - \frac{1}{\beta}, N_n + \frac{1}{\beta}] .$$

$$\mathcal{K} := [-N_1, N_1] \times \cdots \times [-N_n, N_n] .$$

We consider and pseudorandom walks in

$$\mathcal{N}_{Tame} := \mathcal{N} \qquad \text{and} \qquad \mathcal{N}_{Wild} := \vec{x} + \mathcal{N}_{Tame} ,$$

respectively. Let

$$f_i := 1 + \frac{2}{(2N_i + 1)\sqrt{\beta}} \quad \text{for each } 1 \leq i \leq n ,$$

so $f_i$ represents the "stretch factor" in the $i$-th dimension. We have

$$\#\mathcal{N}_{Tame} = \#\mathcal{N}_{Wild} = \#\mathcal{N} = N \prod_{i=1}^{n} f_i , \quad \text{where} \quad N := \#\mathcal{K} .$$

# Theoretical results - General setting

Recall that $N = \prod_{i=1}^{n} (2N_i + 1)$ and $\theta = \frac{c}{\sqrt{N}}$.

## Theorem

*With the notation above: suppose* $\vec{x} = (x_1, \ldots, x_n) \star P$ *for some* $(x_1, \ldots, x_n)$ *with each* $|x_i| \leq N_i$. *Our Algorithm outputs* $(x_1, \ldots, x_n)$ *using an expected*

- $F(\mathcal{K}, \beta, \vec{x})\sqrt{\pi N} + 3/\theta + O(1)$ *jumps,*
- $\beta F(\mathcal{K}, \beta, \vec{x})\sqrt{\pi N} + 3\beta/\theta + O(1)$ *restarts, and*
- $\theta(1 + \beta)F(\mathcal{K}, \beta, \vec{x})\sqrt{\pi N} + O(1)$ *units of memory,*

*where* $F = F_{worst}$ *in the worst case, and* $F_{av}$ *in the average case (over all* $(x_1, \ldots, x_n)$*).*

# Theoretical results - General setting

We define,

$$\alpha(\beta) = \frac{\beta}{(1+\beta)(\beta+1)\left[(1-\beta)\sqrt{1/3}\right]^{1/\beta}}$$

We have the approximate upperbound

$$F(\mathcal{K}, \beta, \vec{x}) \lessapprox (\alpha(\beta))^n \sqrt{\frac{N}{\prod_{i=1}^{n}(2N_i - |x_i| + 1)}}.$$

- $\alpha\left(\frac{1}{5}\right) = 6.6$
- $\alpha\left(\frac{1}{8}\right) \approx 23.3$
- $\alpha\left(\frac{1}{10}\right) \approx 57.6$

# Theoretical results - The always restart algorithm $\beta = 1$

Take the case $\mathcal{K} = [-B, B]^n$. We have an easier analysis for our algorithm when it always restarts. We can simply apply the birthday paradox on the intersection while multiplying by the probability that a step is indeed in the intersection. We get the complexity T of our algorithm to be :

$$\begin{aligned} T &= \sqrt{\frac{\#Tame \times \#Wild}{\#Tame \cap Wild}} \\ &= (1.15)^n \sqrt{\#\mathcal{K}} \\ &= (2B+1)^{n(1/2+log_{2B+1}(1.15))} \end{aligned}$$

# Experimental results - An overview

Parameters choices for Experiments :

- $B = 4$
- $\beta = 1/4$
- $\#\mathcal{K} = 9^n$ and $\#\mathcal{N} = 17^n$
- $\theta = 7/\sqrt{\#\mathcal{K}}$
- footprints : hash (SHA3 256) of associated integer $n$-tuple

Given by our theory, we expect the average number of steps to follow the following formula :

$$total\_jumps = (1+\beta)c^n\sqrt{\pi N} + 3/\theta$$

# Experimental results - An overview

$total\_jumps = (1+\beta)c^n\sqrt{\pi N}$ and $N = \prod_{i=1}^{n}(2N_i+1)$

Table: Comparison of Theoretical and Experimental Results for our Algorithm with precision $10^2$

| | | Total steps | | Constant $c$ | | |
|---|---|---|---|---|---|---|
| $n$ | $N$ | $\mathbb{E}$ | $\sigma$ | *formula* | $\mathbb{E}$ | $\sigma$ |
| 2 | 81 | 34.64 | 27.38 | 3.94 | 1.15 | 0.49 |
| 3 | 729 | 156.93 | 127.88 | 3.94 | 1.25 | 0.37 |
| 4 | 6 561 | 756.76 | 590.03 | 3.94 | 1.33 | 0.28 |
| 5 | 59 049 | 3 634.71 | 2 786.22 | 3.94 | 1.39 | 0.22 |
| 6 | 531 441 | 17 207.54 | 12 879.46 | 3.94 | 1.43 | 0.18 |
| 7 | 4 782 969 | 87 825.55 | 62 529.71 | 3.94 | 1.47 | 0.15 |
| 8 | 43 046 721 | 466 732.56 | 326 683.40 | 3.94 | 1.50 | 0.13 |
| 9 | 387 420 489 | 2 545 298.96 | 1 745 770.89 | 3.94 | 1.54 | 0.12 |
| 10 | $9^{10}$ | 14 743 638.36 | 9 923 252.22 | 3.94 | 1.57 | 0.11 |

# Experimental results - Fixed search space size with different dimensions

$total\_jumps = (1+\beta)c^n\sqrt{\pi N}$ and $N = \prod_{i=1}^{n}(2N_i+1)$

Table: Comparison of Theoretical and Experimental Results for our Algorithm with precision $10^2$ with fixed search space size

| | | Total steps | | Constant $c$ | | |
|---|---|---|---|---|---|---|
| $n$ | $N$ | $\mathbb{E}$ | $\sigma$ | *formula* | $\mathbb{E}$ | $\sigma$ |
| 2 | $64^2 = 2^{12}$ | 814.31 | very big | 3.94 | 1.69 | 0.65 |
| 3 | $16^3 = 2^{12}$ | 1 503.02 | 1 290.02 | 3.94 | 1.47 | 0.42 |
| 4 | $8^4 = 2^{12}$ | 3 255.97 | 2 630.25 | 3.94 | 1.44 | 0.30 |
| 6 | $4^6 = 2^{12}$ | 17 207.54 | 12 879.46 | 3.94 | 1.43 | 0.18 |

# Comparisons - Parallel collision search

The van Oorschot and Wiener's method for parallel collision over a group $\mathcal{G}$ search gives a time complexity in $O(\sqrt{\#\mathcal{G}})$.
We consider two implementation of CSIDH :

- CSIDH-512 : 128-bit security, $log_2(p) = 512$, $B = 5$, $n = 74$
- CSIDH-1024 : 256-bit security, $log_2(p) = 1024$, $B = 2$, $n = 130$

Table: Comparison of Time complexity

| | Time complexity | |
|---|---|---|
| | This work | van Oorschot and Wiener. |
| CSIDH-512 | $2^{134}$ | $2^{128}$ |
| CSIDH-1024 | $2^{179}$ | $2^{256}$ |

# Comparisons - a REGA oriented attack

We compare our results to a paper published in 2023 by Chi-Dom'inguez et al.. The resulting time complexity of their algorithm is $(2B+1)^{nc}$ for some constant c depending on the bound $B$.

Table: Comparison of Time complexity

| $B$ | Time complexity | |
|---|---|---|
| | This work | Chi-Dom'inguez et al. |
| 1 | $3^{0.627}$ | $3^{0.750}$ |
| 2 | $5^{0.587}$ | $5^{0.629}$ |
| 3 | $7^{0.572}$ | $7^{0.618}$ |

What is our advantage ? $\rightarrow$ *Memory*

# Thank you for your attention!