FourQ

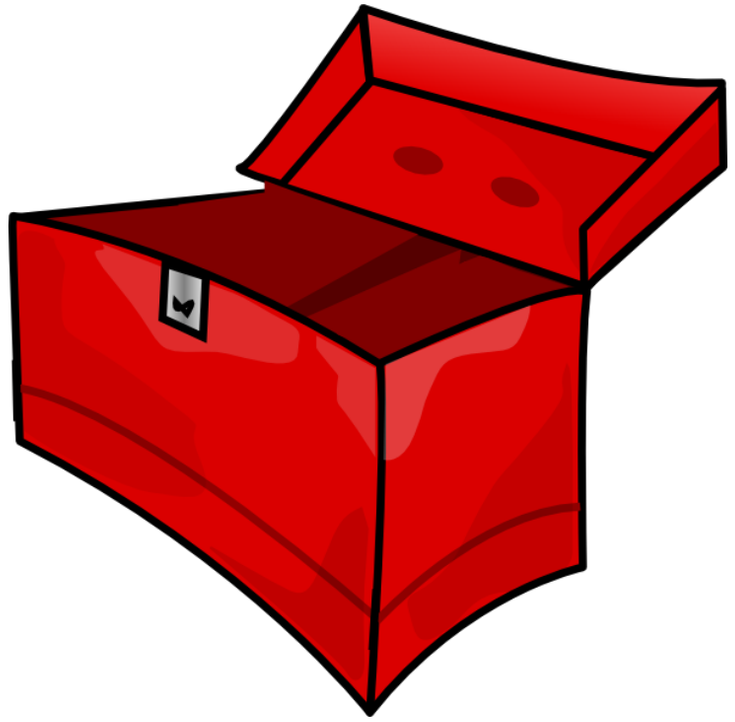
Four-dimensional decompositions on a $\mathbb{Q}$-curve over the Mersenne prime

Craig Costello and Patrick Longa

Full version: http://eprint.iacr.org/2015/565.pdf
FourQlib: http://research.microsoft.com/en-us/projects/fourqlib/

Microsoft Research

Elliptic curves: the best of both worlds

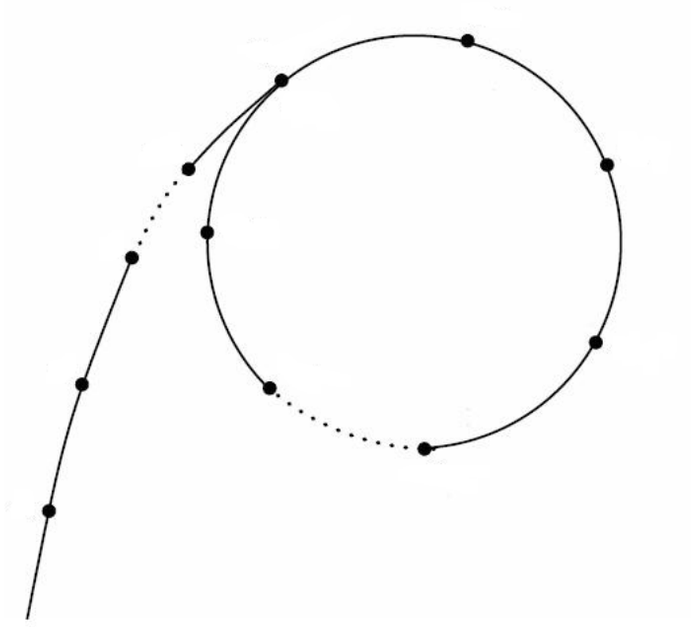


attacker's toolbox

vs.

our toolbox

Elliptic curves: the best of both worlds



$\mathbb{P}^1 \times \mathbb{P}^1$

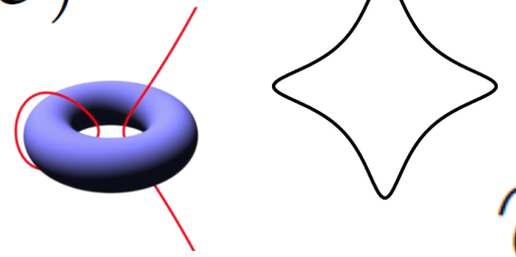
$\mathcal{L} \in \mathbb{R}^n$

$p := 2^{127} - 1$

$\mathbb{F}_{p^2} := \mathbb{F}_p(i)$

$\mathrm{Hom}(\mathcal{E}, \hat{\mathcal{E}})$

ϕ



ψ

$\pi_p : \mathcal{E}_{\mathrm{W}}^{\sigma} \to \mathcal{E}_{\mathrm{W}}$

$\mathbb{Q}(\sqrt{D})$

$\tau : \mathcal{E} \to \hat{\mathcal{E}}$

$(m, 0, 0, 0) + \mathcal{L}$

attacker: generic vs. us: not generic

This work: **FourQ**

Speed: how fast can we possibly go while still giving the attacker a ($\approx 2^{128}$) black-box group?

In other words, how fast can we compute

$$k, P \mapsto [k]P$$

while ensuring Pollard rho (is the best attack and) computes

$$P, [k]P \mapsto k$$

in $\approx 2^{128}$ steps?

Side-channel resistance: don't give real-world (timing) attackers *any* advantage over theoretical ECDLP attacker

All the tricks in the book…

1. **Fast(est) field:** $\mathbb{F}_{p^2} := \mathbb{F}_p(i)$, with $p := 2^{127} - 1$

2. **Fast(est) formulas:** extended twisted Edwards coords.

3. **Largest(est) possible (secure) decomposition:** 4-way multi-exponentiation

Prior (H)ECC speed-records with endomorphisms did two of the above, e.g.,

(1) and (2), but not (3):	[GLS'09], [OLAR'13], [CHS'14] (2-dimensional)
(2) and (3), but not (1):	[LS'12], [BCHL'13] (suboptimal fields)

Endomorphisms in (large char) ECC

- A useful endomorphism $\phi : E \to E$ is a cheap way to compute a large, fixed scalar multiple $\phi(P) = [\lambda]P$
- Random curves won't have a useful ϕ, but two ways to construct them

- **Option 1 [GLV01]:** special curves with (low-degree) CM
  ⇨ only a handful over any particular prime
  ⇨ no hope of finding secure instance over favourite field
- **Option 2 [GLS09,Smi13,GI13]:** use p-power Frobenius over (quad.) extension
  ⇨ not so special
  ⇨ can now find secure instances over favourite (quad.) ext.

- **Combine 1+2 [GLS09,LS12,GI13,Smi14]:** for 4-dimensional decomposition…
  ….but again lose hope of best fields!

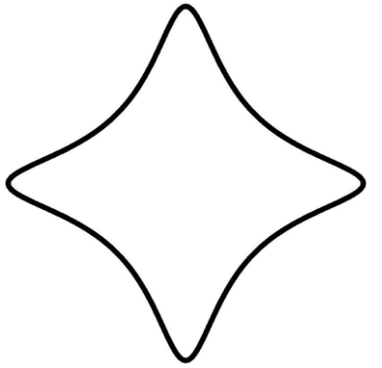
Lose hope, you say?

- [GI13,Smi14]: examples of $\mathbb{Q}$-curves (generalized GLS) with CM

- [Smi14, Thm 6]: Only a handful ($\ll \mathbf{40}$) of low-degree (d=2,3,5,7) $\mathbb{Q}$-curves with CM over a fixed field. Pretty small chance of finding strong group order

- Nevertheless, we still checked Smith's tables over *the* Mersenne prime $p = \mathbf{2^{127} - 1}$ just to be sure…

One curve to rule them all...



$$p := 2^{127} - 1$$

$$\mathcal{E}/\mathbb{F}_{p^2} : \; -x^2 + y^2 = 1 + dx^2y^2$$

$$d := 125317048443780598345676279555970305165 \cdot i + 42058576488057777768770.$$

- Group order is $\mathbf{2^3 \cdot 7^2 \cdot N}$, where $\boldsymbol{N}$ is a 246-bit prime!
- Fastest formulas [HCWD08] "*complete*" $(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_1 + x_2y_2}{y_1y_2 - x_1x_2}, \frac{x_1y_1 - x_2y_2}{x_1y_2 - y_1x_2} \right)$
- Degree-2 $\mathbb{Q}$-curve, meaning degree $2p$ endomorphism $\boldsymbol{\psi}$
- CM by ring of integers in $\mathbb{Q}(\sqrt{-40})$, meaning degree 5 endomorphism $\boldsymbol{\phi}$

Reviewer 3: *The elliptic curve itself has no real right to exist. [They searched] for something that wasn't expected to be there, and were lucky enough to find it...*

How to use endomorphisms

- Recall our task: given integer k and point P, compute $[k]P$
- For any P, we can now quickly get the three points $\phi(P)$, $\psi(P)$ and $\psi(\phi(P))$, where

$$\phi(P) = [\lambda_\phi]P,$$
$$\psi(P) = [\lambda_\psi]P, \text{ and}$$
$$\psi(\phi(P)) = [\lambda_\phi \lambda_\psi]P$$

λ_ϕ = 120989397220997583929700361544554473854860353375346945340423143194252719 08

λ_ψ = 437602317558070402762848557709110782525363684226353183763107140773198670 16

N = 738469956870639001425835363575815738847980758598000974612940963335964295 43

$$k \mapsto (a_1, a_2, a_3, a_4)$$

$$[k]P = [a_1]P + [a_2]\phi(P) + [a_3]\psi(P) + [a_4]\psi(\phi(P))$$

$$k \equiv a_1 + a_2\lambda_\phi + a_3\lambda_\psi + a_4\lambda_\phi\lambda_\psi \bmod N$$

Scalar decompositions

$$k \equiv a_1 + a_2\lambda_\phi + a_3\lambda_\psi + a_4\lambda_\phi\lambda_\psi \bmod N$$

Define the "lattice of decompositions of zero"

$$\mathcal{L} := \langle (z_1, z_2, z_3, z_4) \in \mathbb{Z}^4 \mid z_1 + z_2\lambda_\phi + z_3\lambda_\psi + z_4\lambda_\phi\lambda_\psi \equiv 0 \pmod N \rangle$$

[think: four-dimensional version of 0]

In one dimension: if $k \gg N$, we would reduce $k \mapsto k \bmod N$ (before $[k]P$)

In four dimensions: reduce (e.g., Babai) to get $(a_1, a_2, a_3, a_4) \equiv (k, 0, 0, 0) \bmod \mathcal{L}$

In one dimension, there is only one basis. In higher dimensions, there are (infinitely) many bases!

Scalar decompositions and Babai-optimal bases

$$\mathbf{B}_{\text{bad}} = \begin{pmatrix} N & 0 & 0 & 0 \\ -\lambda_\phi & 1 & 0 & 0 \\ -\lambda_\psi & 0 & 1 & 0 \\ -\lambda_\phi\lambda_\psi & 0 & 0 & 1 \end{pmatrix} \longrightarrow ||(a_1, a_2, a_3, a_4)||_\infty < 2^{246}$$

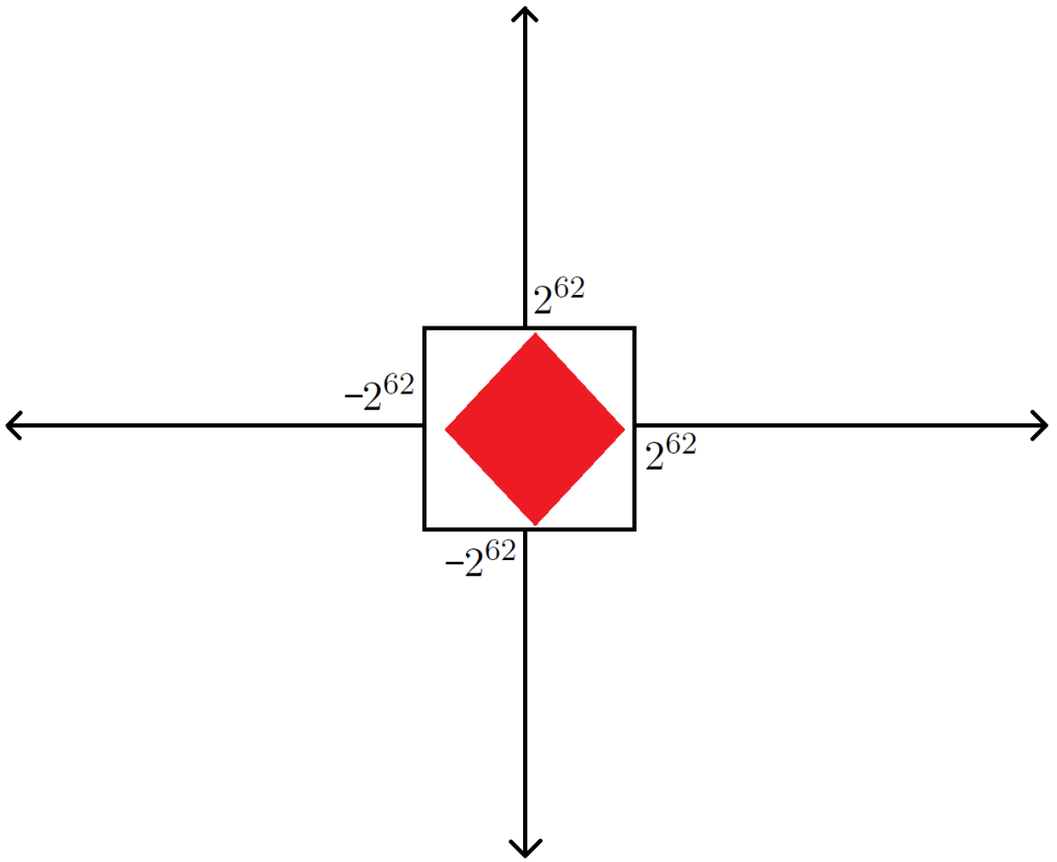
$(k, 0, 0, 0)$

$$|\mathbf{B}_{\text{opt}}[i, j]| = \begin{pmatrix} < 2^{60} & < 2^{61} & < 2^{59} & < 2^{60} \\ < 2^{61} & < 2^{1} & < 2^{1} & < 2^{62} \\ < 2^{61} & < 2^{58} & < 2^{62} & < 2^{60} \\ < 2^{61} & < 2^{62} & < 2^{59} & < 2^{61} \end{pmatrix} \longrightarrow ||(a_1, a_2, a_3, a_4)||_\infty < 2^{62}$$

- $\mathbf{B}_{\text{opt}}$ is not necessarily LLL-reduced, etc. It minimizes the max possible $|\cdot|_\infty$ across all k
- Computing $\mathbf{B}_{\text{opt}}$ is the harder part, but this is offline/precomputation/once-and-for-all!
- Computing (a_1, a_2, a_3, a_4) from $(k, 0, 0, 0)$ is online, but Babai rounding is easy/fast

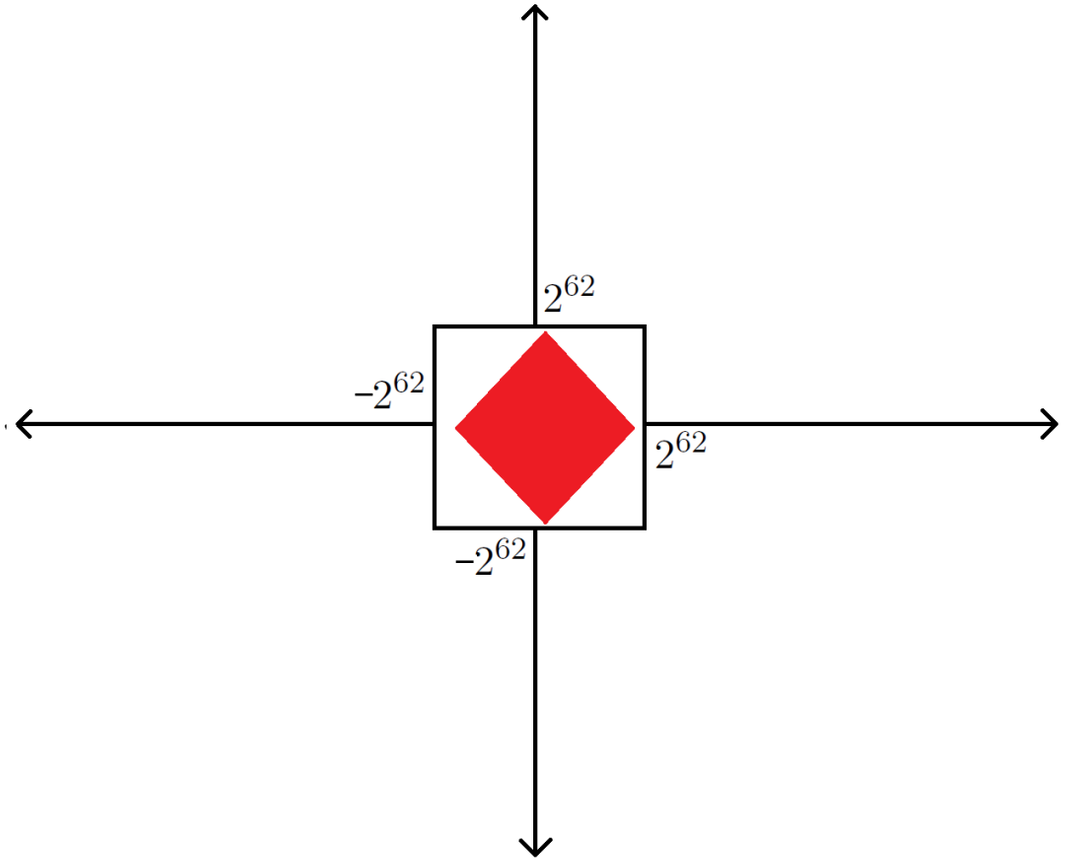
Simplified multi-exponentiations

$$||(a_1, a_2, a_3, a_4)||_\infty < 2^{62}$$



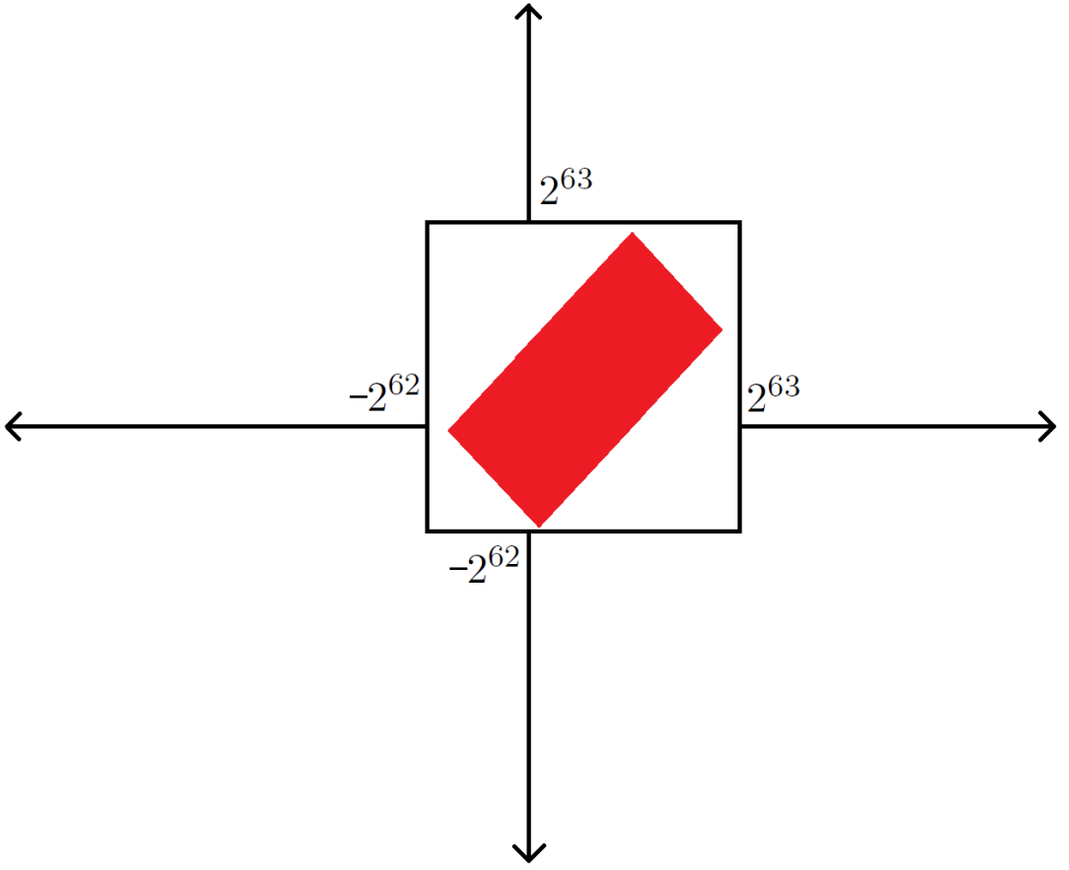
Handling round-off errors

Problem 1: To compute (a_1, a_2, a_3, a_4) from $(k, 0, 0, 0)$ Babai requires four roundings $\left\lfloor \frac{\alpha_i \cdot k}{N} \right\rceil$ but there is no efficient way to compute them (correctly, constant-time, fixed precision)



Handling round-off errors

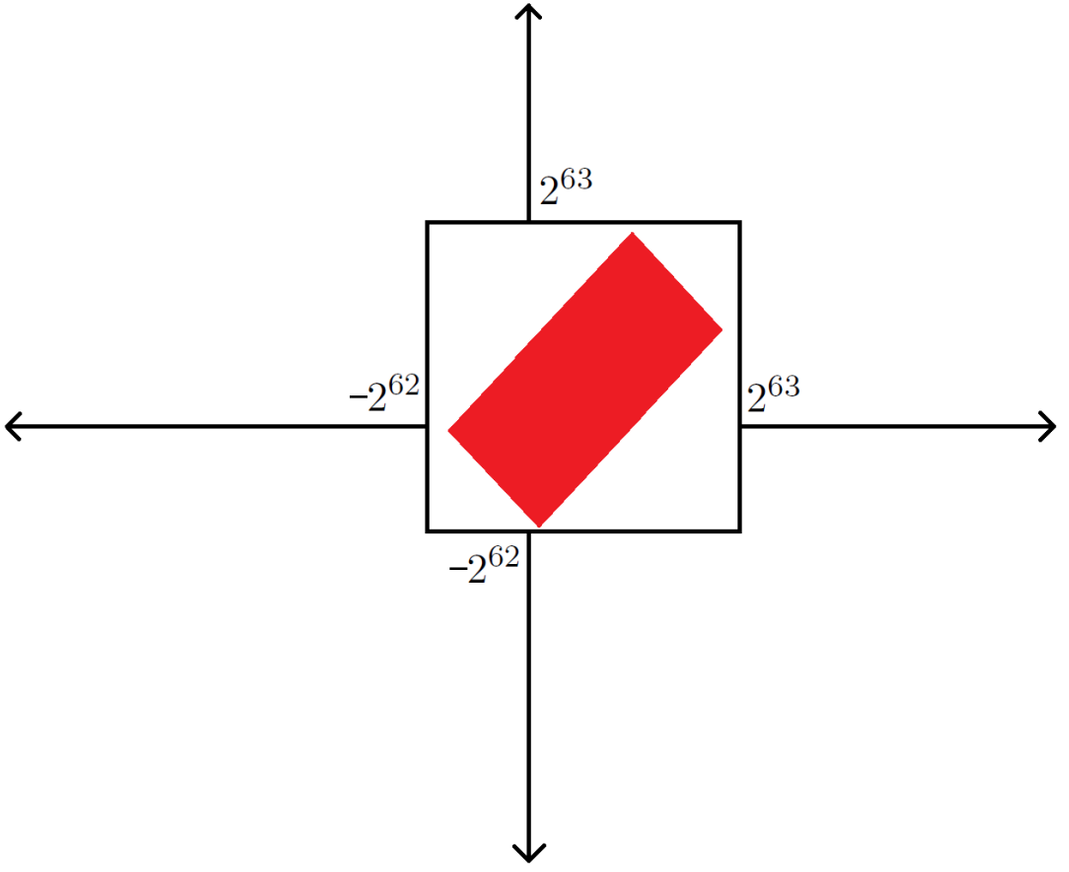
Problem 1: To compute (a_1, a_2, a_3, a_4) from $(k, 0, 0, 0)$ Babai requires four roundings $\left\lfloor \frac{\alpha_i \cdot k}{N} \right\rceil$ but there is no efficient way to compute them (correctly, constant-time, fixed precision)



Solution 1: Compute fast, constant-time approximations to the Babai round-offs

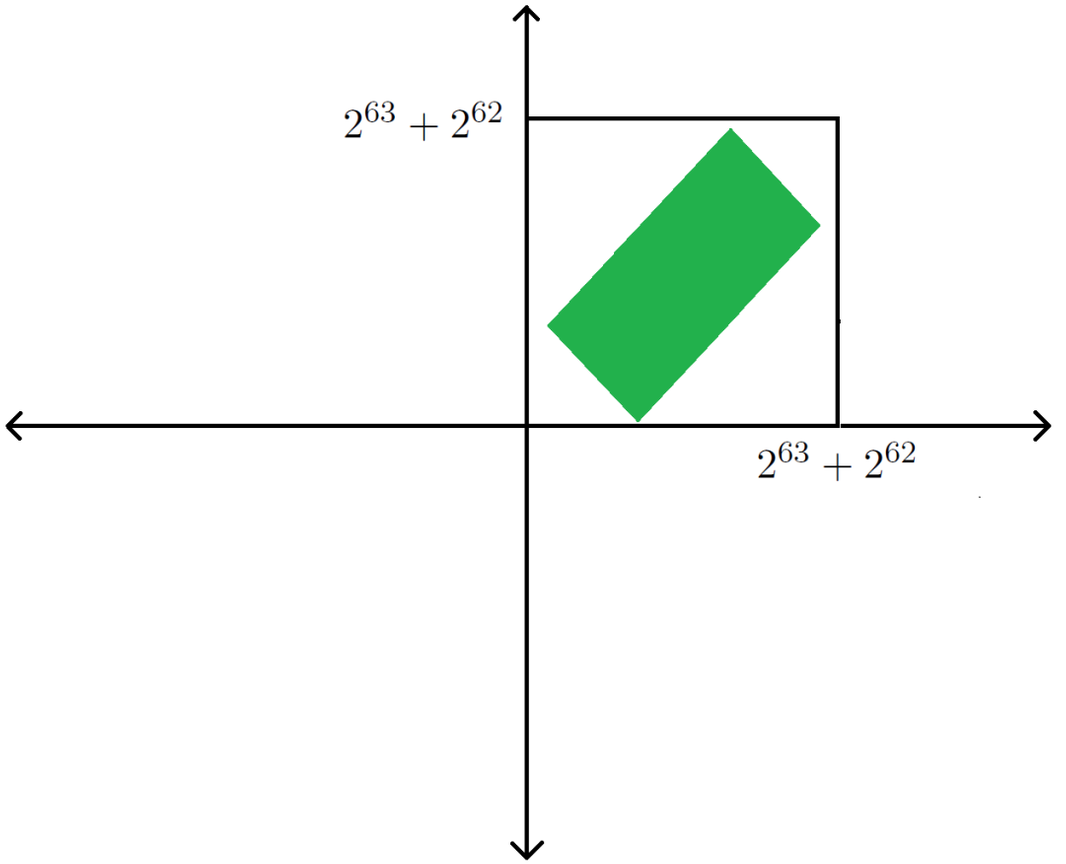
All positive multiscalars via offsets in $\mathcal{L}$

- **Problem 2:** The a_i can be either sign, which is annoying



All positive multiscalars via offsets in $\mathcal{L}$

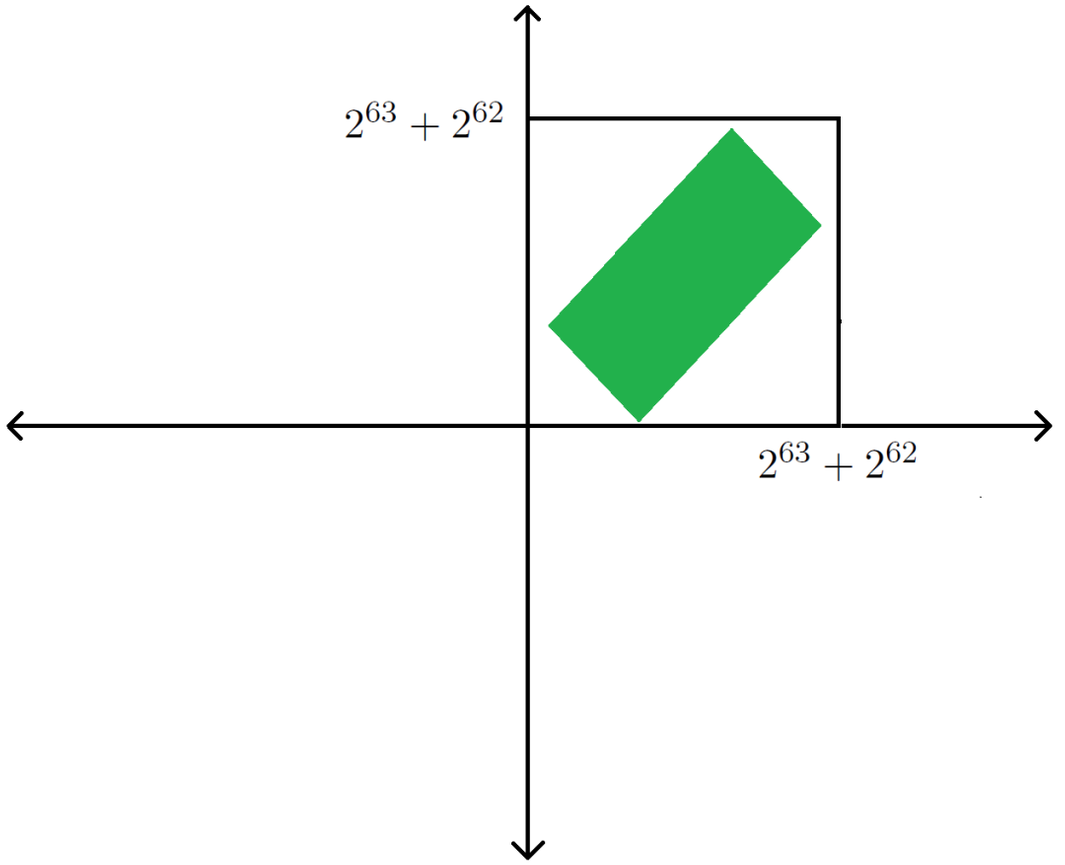
- **Problem 2:** The a_i can be either sign, which is annoying



Solution 2: Use offset vectors in zero lattice $\mathcal{L}$ to move to all-positive region
[think: 4-dimensional analogue of adding multiples of the group order]

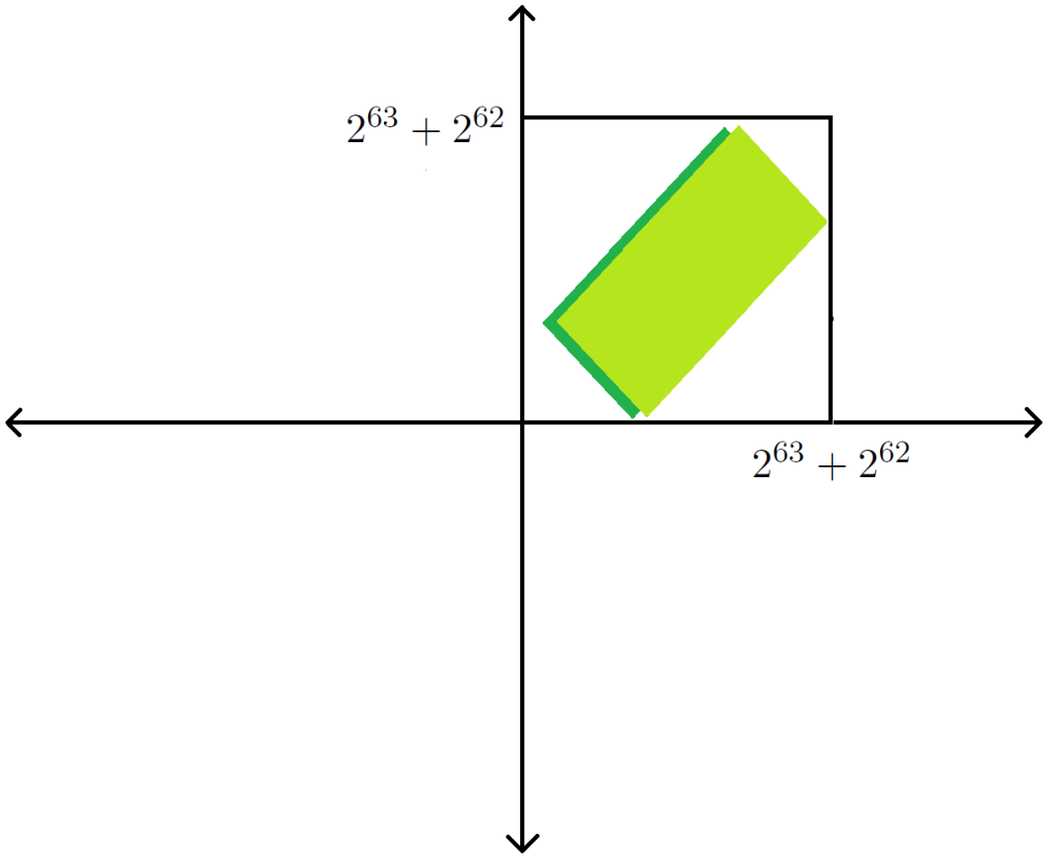
First multiscalar is always odd

Problem 3: a_1 being odd allows a signed, non-zero recoding of k



First multiscalar is always odd

Problem 3: a_1 being odd allows a signed, non-zero recoding of k



Solution 3: Find two translates inside all-positive box, i.e., find short vector in $\mathcal{L}$ whose first component is odd, and make this the difference of the two translates.

The multiscalar multiplication

- Computed $\phi(P)$, $\psi(P)$, $\psi(\phi(P))$, and $k \mapsto (a_1, a_2, a_3, a_4)$, now what?

k = 64840569332679984426672436340494668739430332089137885001096300239355695153788

a_1 = 14445124749170047041
a_2 = 11638376461179115075
a_3 = 5032911711680286358
a_4 = 8810925828828842431

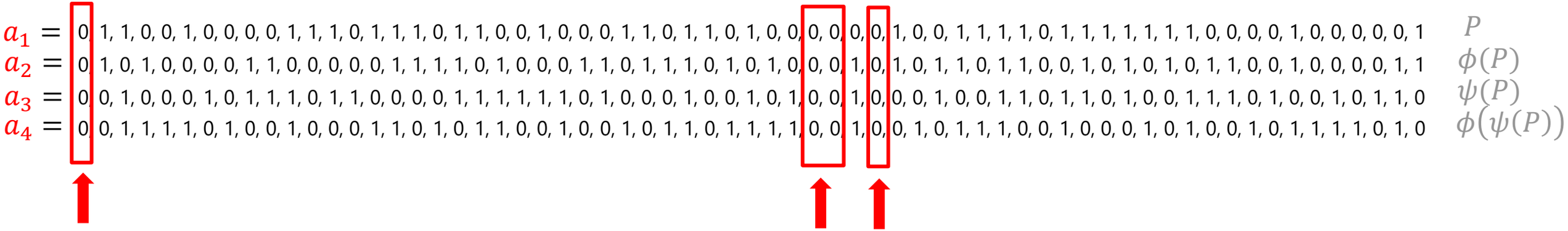
a_1 = 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1 P
a_2 = 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1 $\phi(P)$
a_3 = 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0 $\psi(P)$
a_4 = 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0 $\phi(\psi(P))$

The multiscalar multiplication

- Computed $\phi(P)$, $\psi(P)$, $\psi(\phi(P))$, and $k \mapsto (a_1, a_2, a_3, a_4)$, now what?

k = 64840569332679984426672436340494668739430332089137885001096300239355695153788

a_1 = 14445124749170047041
a_2 = 11638376461179115075
a_3 = 5032911711680286358
a_4 = 8810925828828431



do nothings can leak info!
So let's get rid of them

The multiscalar multiplication

- Computed $\phi(P)$, $\psi(P)$, $\psi(\phi(P))$, and $k \mapsto (a_1, a_2, a_3, a_4)$, now what?

k = 64840569332679984426672436340494668739430332089137885001096300239355695153788

a_1 = 14445124749170047041
a_2 = 11638376461179115075
a_3 = 5032911711680286358
a_4 = 8810925828828431

This is why a_1 is odd

a_1 = 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1 P

a_2 = 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1 $\phi(P)$

a_3 = 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0 $\psi(P)$

a_4 = 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0 $\phi(\psi(P))$

$0, 1$ ⟺ $1, -1$

The multiscalar multiplication

- Computed $\phi(P)$, $\psi(P)$, $\psi(\phi(P))$, and $k \mapsto (a_1, a_2, a_3, a_4)$, now what?

k = 64840569332679984426672436340494668739430332089137885001096300239355695153788

a_1 = 14445124749170047041
a_2 = 11638376461179115075
a_3 = 5032911711680286358
a_4 = 8810925828828842431

a_1 = 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, $\bar{1}$ P
a_2 = 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1 $\phi(P)$
a_3 = 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0 $\psi(P)$
a_4 = 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0 $\phi(\psi(P))$

$0, 1$ ⟺ $1, -1$

The multiscalar multiplication

- Computed $\phi(P)$, $\psi(P)$, $\psi(\phi(P))$, and $k \mapsto (a_1, a_2, a_3, a_4)$, now what?

k = 648405693326799844266724363404946687394303320891378850010963002393556951537 88

a_1 = 14445124749170047041
a_2 = 11638376461179115075
a_3 = 5032911711680286358
a_4 = 8810925828828842431

a_1 = 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, $\bar{1}$, $\bar{1}$ P
a_2 = 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1 $\phi(P)$
a_3 = 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0 $\psi(P)$
a_4 = 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0 $\phi(\psi(P))$

$0, 1$ ⟺ $1, -1$

The multiscalar multiplication

- Computed $\phi(P)$, $\psi(P)$, $\psi(\phi(P))$, and $k \mapsto (a_1, a_2, a_3, a_4)$, now what?

k = 648405693326799844266724363404946687394303320891378850010963002393556951537 88

a_1 = 14445124749170047041
a_2 = 11638376461179115075
a_3 = 5032911711680286358
a_4 = 8810925828828842431

a_1 = 1, $\bar{1}$, 1, 1, $\bar{1}$, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, 1, 1, 1, $\bar{1}$, 1, 1, 1, $\bar{1}$, 1, 1, $\bar{1}$, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, 1, 1, $\bar{1}$, 1, 1, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, 1, 1, 1, 1, $\bar{1}$, 1, 1, 1, 1, 1, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$,$\bar{1}$ P

a_2 = 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1 $\phi(P)$

a_3 = 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0 $\psi(P)$

a_4 = 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0 $\phi(\psi(P))$

$0, 1 \iff 1, -1$

The multiscalar multiplication

- Computed $\phi(P)$, $\psi(P)$, $\psi(\phi(P))$, and $k \mapsto (a_1, a_2, a_3, a_4)$, now what?

k = 64840569332679984426672436340494668739430332089137885001096300239355695153788

a_1 = 14445124749170047041
a_2 = 11638376461179115075
a_3 = 5032911711680286358
a_4 = 8810925828828431

$a_1 =$ 1, $\bar{1}$, 1, 1, $\bar{1}$, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, 1, 1, 1, $\bar{1}$, 1, 1, 1, $\bar{1}$, 1, 1, $\bar{1}$, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, 1, 1, $\bar{1}$, 1, 1, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, 1, 1, 1, 1, $\bar{1}$, 1, 1, 1, 1, 1, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$, 1, $\bar{1}$, $\bar{1}$, $\bar{1}$, $\bar{1}$,$\bar{1}$ P

$a_2 =$ 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1 $\phi(P)$

$a_3 =$ 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0 $\psi(P)$

$a_4 =$ 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0 $\phi(\psi(P))$

Lookup table still has 16 elements
So let's halve it!

Make the signs "align" vertically

$X, 1$ ⟺ $X + 1, -1$

The multiscalar multiplication

k = 648405693326799844266724363404946687394303320891378850010963002393556951537 88

s_i = -1,-1,-1,-1,-1,-1, 1,-1,-1,-1,-1, 1, 1, 1, 1, 1, 1, 1,-1, 1, 1, 1, 1,-1,-1, 1,-1,-1,-1,-1,-1,-1, 1,-1, 1, 1,-1, 1, 1,-1,-1, 1,-1, 1, 1,-1, 1, 1, 1,-1, 1, 1, 1,-1,-1,-1,-1, 1,-1,-1, 1, 1,-1, 1

d_i = 2, 7, 6, 4, 2, 4, 2, 8, 4, 5, 5, 6, 4, 5, 4, 1, 4, 7, 7, 4, 8, 5, 6, 7, 4, 6, 6, 3, 8, 8, 2, 3, 4, 3, 6, 8, 3, 5, 6, 7, 7, 2, 5, 5, 2, 1, 6, 3, 5, 5, 6, 8, 3, 1, 7, 6, 5, 4, 6, 7, 3, 4, 3, 6, 6

The full routine

Steps 1-2: On input of P, compute endomorphisms and build lookup table

$$T[1] = P$$
$$T[2] = P + \phi(P)$$
$$\dots$$
$$T[8] = P + \phi(P) + \psi(P) + \psi(\phi(P))$$

Steps 3-4: On input of k, decompose and recode

k = 64840569332679984426672436340494668739430332089137885001096300239355695153788

s_i = -1,-1,-1,-1,-1, 1,-1,-1,-1,-1, 1, 1, 1, 1, 1, 1, 1,-1, 1, 1, 1, 1,-1,-1, 1,-1,-1,-1,-1,-1,-1, 1,-1, 1, 1,-1, 1, 1,-1,-1,-1, 1,-1,-1, 1, 1,-1, 1, 1, 1,-1, 1, 1, 1,-1,-1,-1,-1, 1,-1,-1, 1, 1,-1, 1

d_i = 2, 7, 6, 4, 2, 4, 2, 8, 4, 5, 5, 6, 4, 5, 4, 1, 4, 7, 7, 4, 8, 5, 6, 7, 4, 6, 6, 3, 8, 8, 2, 3, 4, 3, 6, 8, 3, 5, 6, 7, 7, 2, 5, 5, 2, 1, 6, 3, 5, 5, 6, 8, 3, 1, 7, 6, 5, 4, 6, 7, 3, 4, 3, 6, 6

Step 5: Main loop

64 DBL + 64 ADD

Speed

Bottom line: At the 128-bit level, $\mathbf{FourQ}$ is significantly faster than all other known curve primitives.

e.g.,

Platform	$\mathbf{FourQ}$ w. endos	$\mathbf{FourQ}$ w/o. endos	**Curve25519** [Cho15, eBACS]	**NIST p-256** [GK15]
Atom Pineview	442	803	1,109	-
Intel Sandy	74	138	157	400
Intel Ivy	71	131	159	-
Intel Haswell	59	109	162	312
AMD Kaveri	122	226	301	-

Speed (in thousands of cycles) of $k, P \mapsto [k]P$ on some popular curves & 64-bit platforms.

Security

- Unlike some GLV and GLS endomorphisms, no speedup in rho from ψ and ϕ
- Pollard rho (with negation map) best attack on ECDLP: $2^{122.5}$ point additions in $E[N]$
- $\mathbb{F}_{p^2}$ safe against index calculus/Weil descent
- Large MOV degree and trace of Frobenius
- Yes, small discriminant ($D = -40$), just like other standardized curves $secp192k1$, $secp224k1$, $secp256k1$ (Bitcoin's curve).

Koblitz-Koblitz-Menezs **[KKM11, Ch. 11]**: it could be that special (small D) curves turn out to be more secure than random (large D) ones!

Further comparison to other popular curves

criteria	FourQ	Curve25519	NIST p-256
ECDLP security (best attack)	$2^{122.5}$ (rho with "−")	$2^{125.8}$ (rho with "−")	$2^{127.8}$ (rho with "−")
performance	Fast	Slow	Very slow
rigidity (performance based)	There is **no other** known curve with close perf.	There are **exponentially*** **many** curves with equal or better perf.	There are **exponentially+** **many** curves with equal or better perf

*exp. many other constants over $p = 2^{255} - 19$, exp. many other constants over better fields (e.g., $2^{251} - 9$), exp. many faster fields/curve/scalar-mult choices, etc.

+exp. faster fields and faster curve/scalar-mult choices, etc.

References

[B06] D. J. Bernstein. Curve25519: New Diffie-Hellman speed Records. PKC 2006: 207-228.

[BCHL13] J. W. Bos, C.Costello, H. Hisil, K. Lauter. Fast cryptography in genus 2. EUROCRYPT 2013: 194-210.

[Cho15] T. Cho. Sandy2x: New Curve25519 speed records. SAC 2015.

[CHS14] C. Costello, H. Hisil, B. Smith. Faster compact Diffie-Hellman: endomorphisms on the x-line. EUROCRYPT 2014: 183-200.

[eBACS] D. Bernstein, T. Lange. eBACS: ECRYPT Benchmarking of Cryptographic Systems. http://bench.cr.yp.to

[GI13] A. Guillevic, S. Ionica. Four dimensional GLV via the Weil restriction. ASIACRYPT 2013: 79-96.

[GK15] S. Gueron, V. Krasnov. Fast prime field elliptic curve cryptography with 256-bit primes. Journal of Cryptography Engineering, 5, 2015: 141-151.

[GLS09] S. D. Galbraith, X. Lin, M. Scott. Endomorphisms for faster elliptic curve cryptography on a large class of curves. EUROCRYPT 2009: 518-535.

[GLV01] R. P. Gallant, R. J. Lambert, S. A. Vanstone. Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms. CRYPTO 2001: 190-200.

[HCWD08] H. Hisil, G. Carter, K. K. Wong, E. Dawson. Twisted Edwards curves revisited. ASIACRYPT 2008: 326-343.

[KKM11] Koblitz, Koblitz, Menezes. Elliptic curve cryptography: the serpentine course of a paradigm shift. Journal of Number Theory, 131.5, 2011: 781-814.

[LS12] P. Longa, F. Sica. Four-dimensional Gallant-Lambert-Vanstone Scalar Multiplication. ASIACRYPT 2012: 718-739.

[OLAR13] T Oliveira, J. Lopez, D. F. Aranha, F. Rodriguez-Henriquez. Lambda coordinates for binary elliptic curves. CHES 2013: 311-330.

[Smi13] B. Smith. Fast families of elliptic curves from Q-curves. ASIACRYPT 2013: 61-78

[Smi14] B. Smith. The Q-curve construction for endomorphism-accelerated elliptic curves. Journal of Cryptology, to appear.

FourQ

Four-dimensional decompositions on a $\mathbb{Q}$-curve over the Mersenne prime

Craig Costello and Patrick Longa

Full version: http://eprint.iacr.org/2015/565.pdf
FourQlib: http://research.microsoft.com/en-us/projects/fourqlib/

Microsoft Research