Halving differential addition on Kummer lines

Nicolas Sarkis
Advisors: Razvan Barbulescu and Damien Robert

Institut de Mathématiques de Bordeaux, CANARI team

December 10th, 2024 – CARAMBA seminar

Motivation

Figure: Biometric passport

- ECDSA and ECDH rely on the scalar product of an elliptic curve, we'd like to improve that.
- SIDH computes chains of 2-isogenies $\varphi_1 \circ \cdots \circ \varphi_n$, we are interested in finding 2-isogenies formulas.

Outline

Kummer lines and 2-isogenies

Elliptic curves (char $k \neq 2, 3$)

- Short Weierstrass (general case):

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

- Montgomery curves:

$$E : By^2 = x(x^2 + Ax + 1)$$

- How to compute efficiently $n \cdot P = P + \cdots + P$?

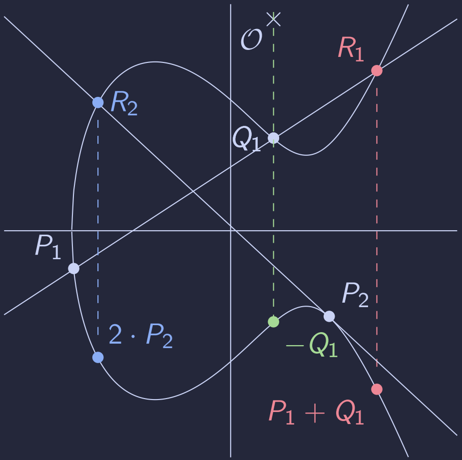


Figure: An elliptic curve

Elliptic curves (char $k \neq 2, 3$)

- Short Weierstrass (general case):

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

- Montgomery curves:

$$E : By^2 = x(x^2 + Ax + 1)$$

- How to compute efficiently $n \cdot P = P + \cdots + P$?

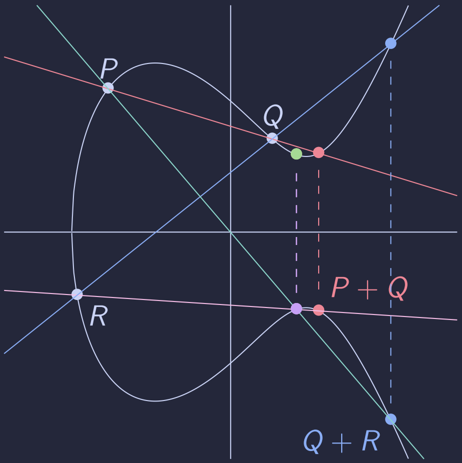


Figure: Trust me it's associative

Kummer line of a Montgomery curve

$$E : By^2 = x(x^2 + \mathcal{A}x + 1)$$

If $P = (X : Y : Z)$, then $-P = (X : -Y : Z)$.

Montgomery XZ-coordinates

$$\pi : E \to \mathbb{P}^1$$

$$(X : Y : Z) \mapsto \begin{cases} \infty := (1 : 0) & \text{if } (X : Y : Z) = (0 : 1 : 0) = \mathcal{O} \\ \frac{X}{Z} := (X : Z) & \text{otherwise} \end{cases}$$

We have $\pi^{-1}(X : Z) = \{(X : \pm Y : Z)\}$.
It is a degree 2 covering: $\#\pi^{-1}(X : Z) = 2$, except when $Y = 0$ or $(X : Z) = \infty$.

Kummer line: general definition

Kummer line

A Kummer line of an elliptic curve E is:

- A degree 2 covering $\pi : E \to \mathbb{P}^1$:

$$\pi^{-1}(\pi(P)) = \{-P, P\}.$$

- 4 ramification points, which correspond to the 2-torsion:

$$\pi^{-1}(\pi(T)) = \{T\} \text{ for } T \in E[2].$$

Kummer line: general definition

Kummer line

A Kummer line of an elliptic curve E is:

- A degree 2 covering $\pi : E \to \mathbb{P}^1$:

$$\pi^{-1}(\pi(P)) = \{-P, P\}.$$

- 4 ramification points, which correspond to the 2-torsion:

$$\pi^{-1}(\pi(T)) = \{T\} \text{ for } T \in E[2].$$

A map between Kummer lines $\varphi : \mathbb{P}^1 \to \mathbb{P}^1$ has to be compatible with this ramification.

Examples

Legendre curve $y^2 = x(x-1)(x-\lambda)$

$$\pi : P \mapsto \begin{cases} (1:0) & \text{if } P = \mathcal{O}, \\ (x:1) & \text{if } P = (x,y). \end{cases}$$

$$\mathcal{O} = (1:0)^*, \quad T_1 = (0:1), \quad T_2 = (1:1), \quad T_3 = (\lambda:1).$$

Examples

Legendre curve $y^2 = x(x-1)(x-\lambda)$

$$\pi : P \mapsto \begin{cases} (1:0) & \text{if } P = \mathcal{O}, \\ (x:1) & \text{if } P = (x,y). \end{cases}$$

$$\mathcal{O} = (1:0)^*, \quad T_1 = (0:1), \quad T_2 = (1:1), \quad T_3 = (\lambda:1).$$

Montgomery curve with rational 2-torsion: $y^2 = x(x-a/b)(x-b/a)$

$$\pi : P \mapsto \begin{cases} (a:b) & \text{if } P = \mathcal{O}, \\ (aX-bZ : bX-aZ) & \text{if } P = (X:Y:Z). \end{cases}$$

$$\mathcal{O} = (a:b)^*, \quad T_1 = (b:a), \quad T_2 = (1:0), \quad T_3 = (0:1).$$

Models we are interested in

- Montgomery Kummer lines (whether $a/b \in k$ or not):

$$\mathcal{O} = (1:0)^*, \quad T_1 = (0:1), \quad T_2 = (a:b), \quad T_3 = (b:a).$$

- Theta model $\theta(a:b)$:

$$\mathcal{O} = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (b:a), \quad T_3 = (-b:a).$$

- Theta squared model $\theta_s(a:b)$:

$$\mathcal{O} = (a:b)^*, \quad T_1 = (b:a), \quad T_2 = (1:0), \quad T_3 = (0:1).$$

$$S : \theta(a:b) \to \theta_s(a^2:b^2), (X:Z) \mapsto (X^2:Z^2)$$

- Theta twisted model $\theta_t(a:b)$:

$$\mathcal{O} = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (1:1), \quad T_3 = (-1:1).$$

$$C : \theta(a:b) \to \theta_t(a^2:b^2), (X:Z) \mapsto (aX:bZ)$$

$$H : \theta_s(a:b) \xrightarrow{\sim} \theta_t(a+b:a-b), (X:Z) \mapsto (X+Z:X-Z)$$

What about the group law?

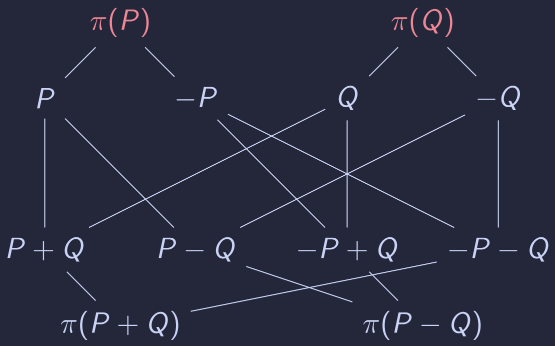


Figure: Two possible choices

What about the group law?

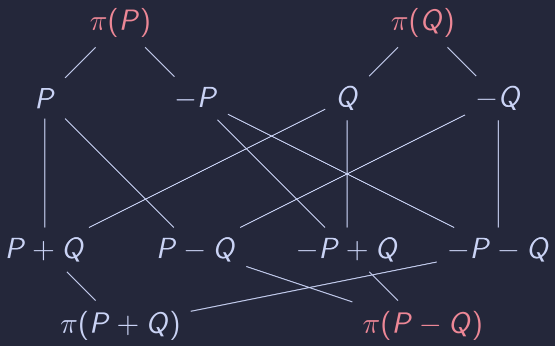


Figure: Two possible choices

However, if we know $\pi(P)$, $\pi(Q)$, $\pi(P-Q)$, we can compute $\pi(P+Q)$.

Arithmetic on $y^2 = x(x^2 + \mathcal{A}x + 1)$[1]

Differential addition $(3M + 2S)$

$u := (X_P + Z_P)(X_Q - Z_Q)$, $v := (X_P - Z_P)(X_Q + Z_Q)$.

$$X_{P+Q} = (u+v)^2, \quad Z_{P+Q} = \frac{X_{P-Q}}{Z_{P-Q}}(u-v)^2.$$

Doubling $(2M + 2S + 1m_0,\ d = \frac{\mathcal{A}+2}{4})$

$u := (X_P + Z_P)^2$, $v := (X_P - Z_P)^2$, $t := u - v$.

$$X_{2.P} = uv, \quad Z_{2.P} = t(v + dt).$$

[1] *P. L. Montgomery*, Speeding the Pollard and elliptic curve methods of factorization, 1987

Arithmetic on $\theta_s(a:b)^2$

Differential addition $(3M + 2S + 1m_0)$

$u := (X_P + Z_P)(X_Q + Z_Q)$, $v := \frac{a+b}{a-b}(X_P - Z_P)(X_Q - Z_Q)$.

$$X_{P+Q} = (u+v)^2, \quad Z_{P+Q} = \frac{X_{P-Q}}{Z_{P-Q}}(u-v)^2.$$

Doubling $(4S + 2m_0)$

$u := (X_P + Z_P)^2$, $v := \frac{a+b}{a-b}(X_P - Z_P)^2$.

$$X_{2.P} = (u+v)^2, \quad Z_{2.P} = \frac{a}{b}(u-v)^2.$$

[2] *P. Gaudry* and *D. Lubicz*, The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines, 2009

$n = 9 = \overline{1001}^2$

$\mathcal{O}$ ——— P

Algorithm 1: Montgomery ladder step

Input: $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit
Output: $(2 \cdot R, R + S)$ if $b = 0$ $(R + S, 2 \cdot S)$ if $b = 1$
Data: The point P

```
Function xDBLADD(R, S, b):
    if b = 0 then
        S ← DiffAdd(R, S, P);
        R ← Doubling(R);
    else if b = 1 then
        R ← DiffAdd(R, S, P);
        S ← Doubling(S);
    end
    return (R, S);
```

Figure: Montgomery ladder

Algorithm 1: Montgomery ladder step

Input: $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit
Output: $(2 \cdot R, R + S)$ if $b = 0$ $(R + S, 2 \cdot S)$ if $b = 1$
Data: The point P

Function xDBLADD(R, S, b):
  if $b = 0$ **then**
    $S \leftarrow$ DiffAdd(R, S, P);
    $R \leftarrow$ Doubling(R);
  else if $b = 1$ **then**
    $R \leftarrow$ DiffAdd(R, S, P);
    $S \leftarrow$ Doubling(S);
  end
  return (R, S);

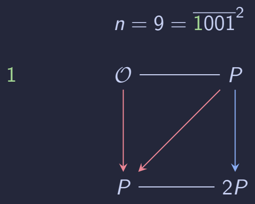


Figure: Montgomery ladder

Algorithm 1: Montgomery ladder step

Input: $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit
Output: $(2 \cdot R, R + S)$ if $b = 0$ $(R + S, 2 \cdot S)$ if $b = 1$
Data: The point P

```
Function xDBLADD(R, S, b):
    if b = 0 then
        S ← DiffAdd(R, S, P);
        R ← Doubling(R);
    else if b = 1 then
        R ← DiffAdd(R, S, P);
        S ← Doubling(S);
    end
    return (R, S);
```

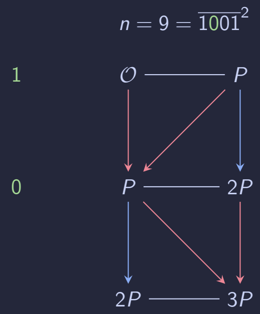


Figure: Montgomery ladder

Algorithm 1: Montgomery ladder step

Input: $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit
Output: $(2 \cdot R, R + S)$ if $b = 0$ $(R + S, 2 \cdot S)$ if $b = 1$
Data: The point P

```
Function xDBLADD(R, S, b):
    if b = 0 then
        S ← DiffAdd(R, S, P);
        R ← Doubling(R);
    else if b = 1 then
        R ← DiffAdd(R, S, P);
        S ← Doubling(S);
    end
    return (R, S);
```

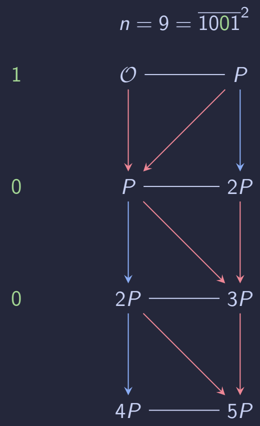


Figure: Montgomery ladder

Algorithm 1: Montgomery ladder step

Input: $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit
Output: $(2 \cdot R, R + S)$ if $b = 0$ $(R + S, 2 \cdot S)$ if $b = 1$
Data: The point P

Function $\texttt{xDBLADD}(R, S, b)$:
  if $b = 0$ **then**
    $S \leftarrow \texttt{DiffAdd}(R, S, P)$;
    $R \leftarrow \texttt{Doubling}(R)$;
  else if $b = 1$ **then**
    $R \leftarrow \texttt{DiffAdd}(R, S, P)$;
    $S \leftarrow \texttt{Doubling}(S)$;
  end
  return (R, S);

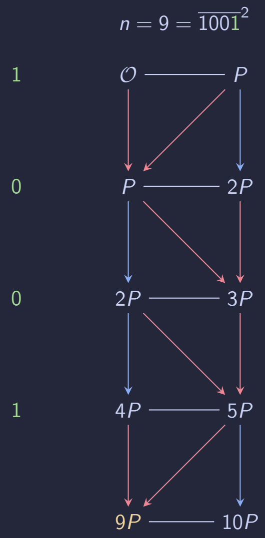


Figure: Montgomery ladder

(Separable) isogenies

- Isogeny: surjective morphism $\varphi : E \to E'$ with finite kernel.
- $\deg \varphi := \# \ker \varphi$, it is multiplicative.
- It always comes with a dual $\widetilde{\varphi} : E' \to E$ such that:

$$\widetilde{\varphi} \circ \varphi = [\deg \varphi]_E \quad \text{and} \quad \varphi \circ \widetilde{\varphi} = [\deg \varphi]_{E'}.$$

2-isogenies on Kummer lines[3]

$\varphi : E \to E'$ a 2-isogeny, $\ker \varphi = \{\mathcal{O}, T\}$, $T \in E[2]$:

$$\widetilde{\varphi} \circ \varphi = [2]_E.$$

For all $P \in E$, $\varphi(P + T) = \varphi(P)$.

[3] *D. Robert* and *N. S.*, Computing 2-isogenies between Kummer lines, 2024

2-isogenies on Kummer lines[3]

$\varphi : E \to E'$ a 2-isogeny, $\ker \varphi = \{\mathcal{O}, T\}$, $T \in E[2]$:

$$\widetilde{\varphi} \circ \varphi = [2]_E.$$

For all $P \in E$, $\varphi(P + T) = \varphi(P)$.

Kummer line with coordinates $(X : Z)$

We want $\overline{\varphi} : \mathbb{P}^1 \to \mathbb{P}^1$:

- $\deg 2$: Expressed in terms of X^2, Z^2, XZ;

[3] *D. Robert* and *N. S.*, Computing 2-isogenies between Kummer lines, 2024

2-isogenies on Kummer lines[3]

$\varphi : E \to E'$ a 2-isogeny, $\ker\varphi = \{\mathcal{O}, T\}$, $T \in E[2]$:

$$\widetilde{\varphi} \circ \varphi = [2]_E.$$

For all $P \in E$, $\varphi(P+T) = \varphi(P)$.

Kummer line with coordinates $(X : Z)$

We want $\overline{\varphi} : \mathbb{P}^1 \to \mathbb{P}^1$:

- deg 2: Expressed in terms of X^2, Z^2, XZ;
- Kummer lines: respecting ramification;

[3] *D. Robert* and *N. S.*, Computing 2-isogenies between Kummer lines, 2024

2-isogenies on Kummer lines[3]

$\varphi : E \to E'$ a 2-isogeny, $\ker \varphi = \{\mathcal{O}, T\}$, $T \in E[2]$:

$$\widetilde{\varphi} \circ \varphi = [2]_E.$$

For all $P \in E$, $\varphi(P + T) = \varphi(P)$.

Kummer line with coordinates $(X : Z)$

We want $\overline{\varphi} : \mathbb{P}^1 \to \mathbb{P}^1$:

- deg 2: Expressed in terms of X^2, Z^2, XZ;
- Kummer lines: respecting ramification;
- Isogeny: invariant by $t_T : P \mapsto P + T$.

[3] *D. Robert* and *N. S.*, Computing 2-isogenies between Kummer lines, 2024

Half differential addition

The differential addition isogeny

Differential addition isogeny

$$F : E \times E \to E \times E$$
$$(P, Q) \mapsto (P + Q, P - Q)$$

It is a $(2, 2)$-isogeny (between abelian surfaces), with kernel:

$$\ker F = \{(T, T) \mid T \in E[2]\} = \langle (T_1, T_1), (T_2, T_2) \rangle, \text{ where } E[2] = \langle T_1, T_2 \rangle.$$

The differential addition isogeny

Differential addition isogeny

$$F : E \times E \to E \times E$$
$$(P, Q) \mapsto (P + Q, P - Q)$$

It is a $(2, 2)$-isogeny (between abelian surfaces), with kernel:

$$\ker F = \{(T, T) \mid T \in E[2]\} = \langle (T_1, T_1), (T_2, T_2) \rangle, \text{ where } E[2] = \langle T_1, T_2 \rangle.$$

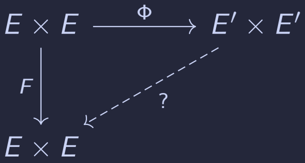
Diagonal isogeny ($\varphi : E \to E'$ a 2-isogeny with kernel $\langle T_1 \rangle$)

$$\Phi : E \times E \to E' \times E'$$
$$(P, Q) \mapsto (\varphi(P), \varphi(Q))$$

Φ is a $(2, 2)$-isogeny, with kernel $\langle T_1 \rangle \times \langle T_1 \rangle = \langle (\mathcal{O}, T_1), (T_1, \mathcal{O}) \rangle$.

Our idea: can we factor F?

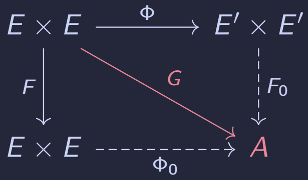
$$F : (P, Q) \mapsto (P + Q, P - Q), \qquad \Phi : (P, Q) \mapsto (\varphi(P), \varphi(Q)).$$



Our idea: can we factor F?

$$F : (P, Q) \mapsto (P + Q, P - Q), \qquad \Phi : (P, Q) \mapsto (\varphi(P), \varphi(Q)).$$

We can't factor F or Φ because $\ker F \not\subseteq \ker \Phi$ or $\ker \Phi \not\subseteq \ker F$.
$\rightarrow$ We consider a third one G with $\ker G := \ker F + \ker \Phi$.



Half differential addition

$$F : (P, Q) \mapsto (P + Q, P - Q), \quad \Phi : (P, Q) \mapsto (\varphi(P), \varphi(Q)).$$

Definition

Half differential addition formulas relative to φ are formulas such that given $\varphi(P)$, $\varphi(Q)$ and $P - Q$, can compute $P + Q$ on the Kummer line.

Notation: $P + Q = \mathtt{HalfDiffAdd}_{\varphi}(\varphi(P), \varphi(Q), P - Q)$.

Half differential addition

$$F : (P, Q) \mapsto (P + Q, P - Q), \quad \Phi : (P, Q) \mapsto (\varphi(P), \varphi(Q)).$$

Definition

Half differential addition formulas relative to φ are formulas such that given $\varphi(P)$, $\varphi(Q)$ and $P - Q$, can compute $P + Q$ on the Kummer line.

Notation: $P + Q = \mathtt{HalfDiffAdd}_{\varphi}(\varphi(P), \varphi(Q), P - Q)$.
For consistency, $2 \cdot P = \mathtt{HalfDouble}_{\varphi}(\varphi(P))$ $(= \widetilde{\varphi}(\varphi(P)))$.

An example

On the theta model $\theta(a:b)$ with ramification

$$\mathcal{O} = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (b:a), \quad T_3 = (-b:a).$$

Set $(A^2 : B^2) := (a^2 + b^2 : a^2 - b^2)$, $\ker \varphi = \langle T_1 \rangle$:

$$\varphi : (X:Z) \in \theta(a:b) \mapsto (B(X^2+Z^2) : A(X^2-Z^2)) \in \theta(A:B)$$

`HalfDiffAdd`$_\varphi(\varphi(P), \varphi(Q), P-Q)$ $(4M)$

$$(X_{P+Q}X_{P-Q} : Z_{P+Q}Z_{P-Q}) = \begin{pmatrix} X_{\varphi(P)}X_{\varphi(Q)} + Z_{\varphi(P)}Z_{\varphi(Q)} \\ X_{\varphi(P)}X_{\varphi(Q)} - Z_{\varphi(P)}Z_{\varphi(Q)} \end{pmatrix}$$

In comparison, a full differential addition in $\theta(a:b)$ is $3M + 4S + 1m_0$ (or $3M + 2S + 1m_0$ with squared coordinates).

Half ladder

Motivation

In the usual Montgomery ladder, we perform one differential addition and one doubling per bit: we compute the images by φ and immediately get the results back on the original curve.

Motivation

In the usual Montgomery ladder, we perform one differential addition and one doubling per bit: we compute the images by φ and immediately get the results back on the original curve.

Instead, we will pre-compute the pre-required images, and then perform the ladder backwards with `HalfDiffAdd` and `HalfDouble`.

Context and pre-computation

We want to compute $n \cdot P$, where $P \in \mathcal{K}$.

- $n = (b_{\ell-1}, b_{\ell-2}, \dots, b_0)$ has ℓ bits.
- $P_0 := P$ and $\mathcal{K}_0 := \mathcal{K}$.
- We have $\mathcal{K}_1, \dots, \mathcal{K}_\ell$ Kummer lines and $\varphi_i : \mathcal{K}_{i-1} \to \mathcal{K}_i$ 2-isogenies.
- $P_i := \varphi_i(P_{i-1})$.

$$\begin{array}{ccccccccc} \mathcal{K}_0 & \xrightarrow{\varphi_1} & \mathcal{K}_1 & \xrightarrow{\varphi_2} & \mathcal{K}_2 & \xrightarrow{\varphi_3} & \cdots & \xrightarrow{\varphi_\ell} & \mathcal{K}_\ell \\ P_0 & \longmapsto & P_1 & \longmapsto & P_2 & \longmapsto & \cdots & \longmapsto & P_\ell \end{array}$$

Figure: Successive images

In practice, $\varphi_{2i+1} = \varphi$ and $\varphi_{2i} = \widetilde{\varphi}$.

Main loop

$$\begin{array}{ccccccccccccc} \mathcal{K}_0 & \longrightarrow & \mathcal{K}_1 & \longrightarrow & \cdots & \longrightarrow & \mathcal{K}_{i-1} & \xrightarrow{\varphi_i} & \mathcal{K}_i & \longrightarrow & \cdots & \longrightarrow & \mathcal{K}_\ell \\ P_0 & \longmapsto & P_1 & \longmapsto & \cdots & \longmapsto & P_{i-1} & \longmapsto & P_i & \longmapsto & \cdots & \longmapsto & P_\ell \\ & & & & & & & & (R_i, S_i) & & & & \end{array}$$

If we know $R_i = m_i \cdot P_i$ and $S_i = (m_i + 1) \cdot P_i$ on $\mathcal{K}_i$, then:

- $R_i = \varphi_i(m_i \cdot P_{i-1})$ and $S_i = \varphi_i((m_i + 1) \cdot P_{i-1})$,

Main loop

$$\mathcal{K}_0 \longrightarrow \mathcal{K}_1 \longrightarrow \cdots \longrightarrow \mathcal{K}_{i-1} \xrightarrow{\varphi_i} \mathcal{K}_i \longrightarrow \cdots \longrightarrow \mathcal{K}_\ell$$

$$P_0 \longmapsto P_1 \longmapsto \cdots \longmapsto P_{i-1} \longmapsto P_i \longmapsto \cdots \longmapsto P_\ell$$

$$(R_i, S_i)$$

If we know $R_i = m_i \cdot P_i$ and $S_i = (m_i + 1) \cdot P_i$ on $\mathcal{K}_i$, then:

- $R_i = \varphi_i(m_i \cdot P_{i-1})$ and $S_i = \varphi_i((m_i + 1) \cdot P_{i-1})$,
- Hence $(2m_i + 1) \cdot P_{i-1} = \texttt{HalfDiffAdd}_{\varphi_i}(R_i, S_i, P_{i-1})$,

Main loop

$$\begin{array}{ccccccccccccc} \mathcal{K}_0 & \longrightarrow & \mathcal{K}_1 & \longrightarrow & \cdots & \longrightarrow & \mathcal{K}_{i-1} & \xrightarrow{\varphi_i} & \mathcal{K}_i & \longrightarrow & \cdots & \longrightarrow & \mathcal{K}_\ell \\ P_0 & \longmapsto & P_1 & \longmapsto & \cdots & \longmapsto & P_{i-1} & \longmapsto & P_i & \longmapsto & \cdots & \longmapsto & P_\ell \\ & & & & & & & & (R_i, S_i) & & & & \end{array}$$

If we know $R_i = m_i \cdot P_i$ and $S_i = (m_i + 1) \cdot P_i$ on $\mathcal{K}_i$, then:

- $R_i = \varphi_i(m_i \cdot P_{i-1})$ and $S_i = \varphi_i((m_i + 1) \cdot P_{i-1})$,
- Hence $(2m_i + 1) \cdot P_{i-1} = \mathtt{HalfDiffAdd}_{\varphi_i}(R_i, S_i, P_{i-1})$,
- Moreover, $2m_i \cdot P_{i-1} = \mathtt{HalfDouble}_{\varphi_i}(R_i)$ and $(2m_i + 2) \cdot P_{i-1} = \mathtt{HalfDouble}_{\varphi_i}(S_i)$.

Main loop

$$\mathcal{K}_0 \longrightarrow \mathcal{K}_1 \longrightarrow \cdots \longrightarrow \mathcal{K}_{i-1} \xrightarrow{\varphi_i} \mathcal{K}_i \longrightarrow \cdots \longrightarrow \mathcal{K}_\ell$$

$$P_0 \longmapsto P_1 \longmapsto \cdots \longmapsto P_{i-1} \longmapsto P_i \longmapsto \cdots \longmapsto P_\ell$$

$$(R_{i-1}, S_{i-1}) \dashleftarrow (R_i, S_i)$$

If we know $R_i = m_i \cdot P_i$ and $S_i = (m_i + 1) \cdot P_i$ on $\mathcal{K}_i$, then:

- $R_i = \varphi_i(m_i \cdot P_{i-1})$ and $S_i = \varphi_i((m_i + 1) \cdot P_{i-1})$,
- Hence $(2m_i + 1) \cdot P_{i-1} = \mathtt{HalfDiffAdd}_{\varphi_i}(R_i, S_i, P_{i-1})$,
- Moreover, $2m_i \cdot P_{i-1} = \mathtt{HalfDouble}_{\varphi_i}(R_i)$ and $(2m_i + 2) \cdot P_{i-1} = \mathtt{HalfDouble}_{\varphi_i}(S_i)$.

We can compute $R_{i-1} = m_{i-1} \cdot P_{i-1}$ and $S_{i-1} = (m_{i-1} + 1) \cdot P_{i-1}$.

Algorithm 1: Montgomery ladder step

Input: $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit
Output: $(2 \cdot R, R + S)$ if $b = 0$ $(R + S, 2 \cdot S)$ if $b = 1$
Data: The point P

```
Function xDBLADD(R, S, b):
    if b = 0 then
        S ← DiffAdd(R, S, P);
        R ← Doubling(R);
    else if b = 1 then
        R ← DiffAdd(R, S, P);
        S ← Doubling(S);
    end
    return (R, S);
```

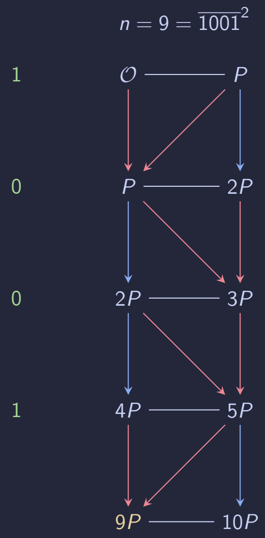


Figure: Montgomery ladder

Algorithm 2: Half ladder step for a 2-isogeny φ

Input: $\varphi(R)$, $\varphi(S)$ where $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit

Output: $(2 \cdot R, R + S)$ if $b = 0$
$(R + S, 2 \cdot S)$ if $b = 1$

Data: The point P

Function $\texttt{HalfxDBLADD}_\varphi(\varphi(R), \varphi(S), b)$:
  if $b = 0$ **then**
    $S \leftarrow \texttt{HalfDiffAdd}_\varphi(\varphi(R), \varphi(S), P)$;
    $R \leftarrow \texttt{HalfDouble}_\varphi(\varphi(R))$;
  else if $b = 1$ **then**
    $R \leftarrow \texttt{HalfDiffAdd}_\varphi(\varphi(R), \varphi(S), P)$;
    $S \leftarrow \texttt{HalfDouble}_\varphi(\varphi(S))$;
  end
  return (R, S);

$n = 9 = \overline{1001}^2$

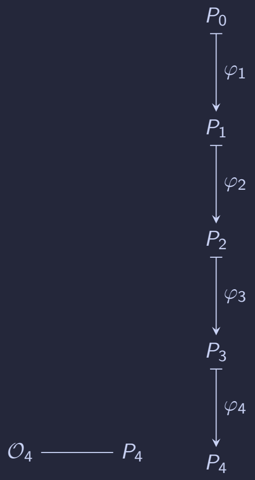


Figure: Half ladder

Algorithm 2: Half ladder step for a 2-isogeny φ

Input: $\varphi(R)$, $\varphi(S)$ where $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit

Output: $(2 \cdot R, R+S)$ if $b = 0$; $(R+S, 2 \cdot S)$ if $b = 1$

Data: The point P

```
Function HalfxDBLADD_φ(φ(R), φ(S), b):
    if b = 0 then
        S ← HalfDiffAdd_φ(φ(R), φ(S), P);
        R ← HalfDouble_φ(φ(R));
    else if b = 1 then
        R ← HalfDiffAdd_φ(φ(R), φ(S), P);
        S ← HalfDouble_φ(φ(S));
    end
    return (R, S);
```

$n = 9 = \overline{1001}^2$

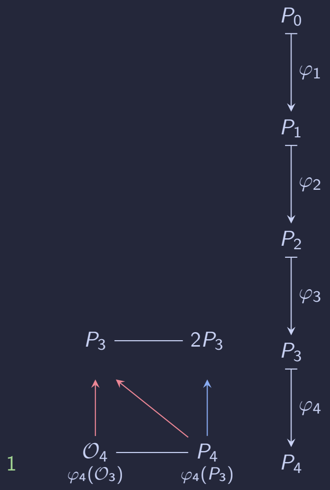


Figure: Half ladder

Algorithm 2: Half ladder step for a 2-isogeny φ

Input: $\varphi(R)$, $\varphi(S)$ where $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit

Output: $(2 \cdot R, R + S)$ if $b = 0$; $(R + S, 2 \cdot S)$ if $b = 1$

Data: The point P

Function $\texttt{HalfxDBLADD}_\varphi(\varphi(R), \varphi(S), b)$:

  if $b = 0$ **then**

    $S \leftarrow \texttt{HalfDiffAdd}_\varphi(\varphi(R), \varphi(S), P)$;

    $R \leftarrow \texttt{HalfDouble}_\varphi(\varphi(R))$;

  else if $b = 1$ **then**

    $R \leftarrow \texttt{HalfDiffAdd}_\varphi(\varphi(R), \varphi(S), P)$;

    $S \leftarrow \texttt{HalfDouble}_\varphi(\varphi(S))$;

  end

  return (R, S);

$n = 9 = \overline{1001}^2$

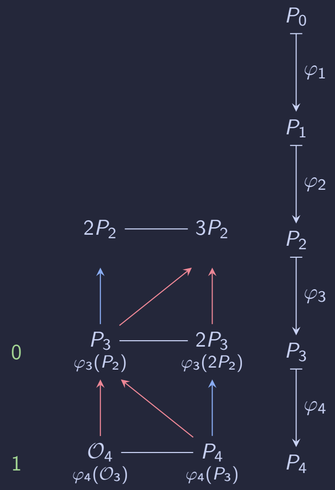


Figure: Half ladder

Algorithm 2: Half ladder step for a 2-isogeny φ

Input: $\varphi(R)$, $\varphi(S)$ where $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit

Output: $(2 \cdot R, R+S)$ if $b = 0$
$(R+S, 2 \cdot S)$ if $b = 1$

Data: The point P

```
Function HalfxDBLADD_φ(φ(R), φ(S), b):
    if b = 0 then
        S ← HalfDiffAdd_φ(φ(R), φ(S), P);
        R ← HalfDouble_φ(φ(R));
    else if b = 1 then
        R ← HalfDiffAdd_φ(φ(R), φ(S), P);
        S ← HalfDouble_φ(φ(S));
    end
    return (R, S);
```

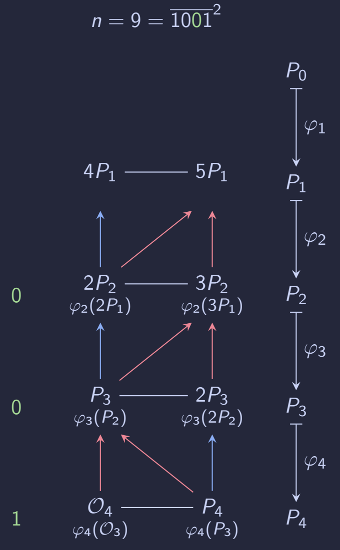


Figure: Half ladder

Algorithm 2: Half ladder step for a 2-isogeny φ

Input: $\varphi(R)$, $\varphi(S)$ where $R = m \cdot P$, $S = (m+1) \cdot P$, b a bit

Output: $(2 \cdot R, R + S)$ if $b = 0$; $(R + S, 2 \cdot S)$ if $b = 1$

Data: The point P

Function $\texttt{HalfxDBLADD}_\varphi(\varphi(R), \varphi(S), b)$:
- **if** $b = 0$ **then**
  - $S \leftarrow \texttt{HalfDiffAdd}_\varphi(\varphi(R), \varphi(S), P)$;
  - $R \leftarrow \texttt{HalfDouble}_\varphi(\varphi(R))$;
- **else if** $b = 1$ **then**
  - $R \leftarrow \texttt{HalfDiffAdd}_\varphi(\varphi(R), \varphi(S), P)$;
  - $S \leftarrow \texttt{HalfDouble}_\varphi(\varphi(S))$;
- **end**
- **return** (R, S);

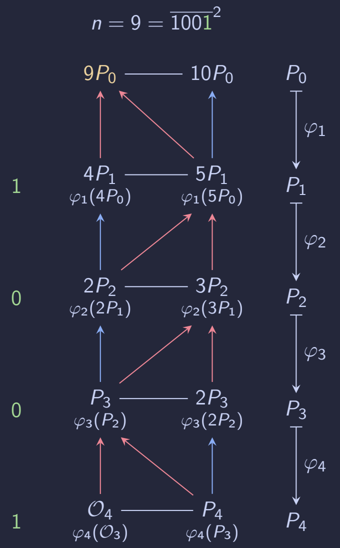


Figure: Half ladder

Computational cost

On our theta model $\theta(a:b)$ previously studied, with $\varphi_{2i+1} = \varphi$ and $\varphi_{2i} = \widetilde{\varphi}$:

- $\varphi : \theta(a:b) \to \theta(A:B)$ and $\widetilde{\varphi}$: $2S + 1m_0$.
- $\texttt{HalfDiffAdd}_{\varphi}$ and $\texttt{HalfDiffAdd}_{\widetilde{\varphi}}$: $4M$.
- $\texttt{HalfDouble}_{\varphi}$ and $\texttt{HalfDouble}_{\widetilde{\varphi}}$: $2S + 1m_0$.

Computational cost

On our theta model $\theta(a:b)$ previously studied, with $\varphi_{2i+1} = \varphi$ and $\varphi_{2i} = \widetilde{\varphi}$:

- $\varphi : \theta(a:b) \to \theta(A:B)$ and $\widetilde{\varphi}$: $2S + 1m_0$.
- $\texttt{HalfDiffAdd}_{\varphi}$ and $\texttt{HalfDiffAdd}_{\widetilde{\varphi}}$: $4M$.
- $\texttt{HalfDouble}_{\varphi}$ and $\texttt{HalfDouble}_{\widetilde{\varphi}}$: $2S + 1m_0$.

	Montgomery ladder	Half ladder, our contribution
Non-normalized base point	$6M + 4S + 1m_0$	$4M + 4S + 2m_0$
Normalized base point	$5M + 4S + 1m_0$ (or $4M + 4S + 2m_0$)	

Table: Ladder costs per bit with no pre-computation

Computational cost

On our theta model $\theta(a : b)$ previously studied, with $\varphi_{2i+1} = \varphi$ and $\varphi_{2i} = \widetilde{\varphi}$:

- $\varphi : \theta(a : b) \to \theta(A : B)$ and $\widetilde{\varphi}$: $2S + 1m_0$.
- $\texttt{HalfDiffAdd}_{\varphi}$ and $\texttt{HalfDiffAdd}_{\widetilde{\varphi}}$: $4M$.
- $\texttt{HalfDouble}_{\varphi}$ and $\texttt{HalfDouble}_{\widetilde{\varphi}}$: $2S + 1m_0$.

Algorithm	Pre-computation	Step
Montgomery ladder LtR	—	$5M + 4S + 1m_0$
Montgomery ladder RtL[4]	$2M + 2S + 1m_0$	$4M + 2S$
Half ladder, our contribution	$2S + 1m_0$	$4M + 2S + 1m_0$

Table: Ladder costs per bit with a pre-computation but no normalization

[4] *T. Oliveira, J. C. López-Hernández, H. Hisil, A. Faz-Hernández* and *F. Rodríguez-Henríquez*, How to (Pre-)Compute a Ladder - Improving the Performance of X25519 and X448, 2017

Computational cost

Still holds on a theta twisted model $\theta_t(a:b)$ with a few tweaks (equiv. to Montgomery with rational 2-torsion):

- $\varphi : \theta_t(a:b) \to \theta_t(a':b')$ and $\widetilde{\varphi}$: $2S + 1m_0$.
- $\texttt{HalfDiffAdd}_{\varphi}$ and $\texttt{HalfDiffAdd}_{\widetilde{\varphi}}$: $4M + 2m_0$ $\to$ can be adjusted to $4M$.
- $\texttt{HalfDouble}_{\varphi}$ and $\texttt{HalfDouble}_{\widetilde{\varphi}}$: $2S + 1m_0$.

Algorithm	Pre-computation	Step
Montgomery ladder LtR	—	$5M + 4S + 1m_0$
Montgomery ladder RtL[4]	$2M + 2S + 1m_0$	$4M + 2S$
Half ladder, our contribution	$2S + 1m_0$	$4M + 2S + 1m_0$

Table: Ladder costs per bit with a pre-computation but no normalization

[4] *T. Oliveira, J. C. López-Hernández, H. Hisil, A. Faz-Hernández* and *F. Rodríguez-Henríquez*, How to (Pre-)Compute a Ladder - Improving the Performance of X25519 and X448, 2017

Curve25519

Curve25519[5] does not have rational 2-torsion, but it has a 8-torsion point on $\mathbb{F}_p$ with $p = 2^{255} - 19$:

$$y^2 = x(x^2 + 486662x + 1) \;\rightarrow\; M(A:B) \text{ Kummer line}$$

[5] *D. J. Bernstein*, Curve25519: New Diffie-Hellman Speed Records, 2006

Curve25519

Curve25519[5] does not have rational 2-torsion, but it has a 8-torsion point on $\mathbb{F}_p$ with $p = 2^{255} - 19$:

$$y^2 = x(x^2 + 486662x + 1) \;\rightarrow\; M(A : B) \text{ Kummer line}$$

Because of the 8-torsion, it is 2-isogenous to a Montgomery curve with rational 2-torsion:

$$\psi : M(A : B) \rightarrow \theta_t(a : b)$$

We can compute $\texttt{HalfDiffAdd}_\psi$ formulas.

[5] *D. J. Bernstein*, Curve25519: New Diffie-Hellman Speed Records, 2006

Curve25519

Curve25519[5] does not have rational 2-torsion, but it has a 8-torsion point on $\mathbb{F}_p$ with $p = 2^{255} - 19$:

$$y^2 = x(x^2 + 486662x + 1) \;\rightarrow\; M(A : B) \text{ Kummer line}$$

Because of the 8-torsion, it is 2-isogenous to a Montgomery curve with rational 2-torsion:

$$\psi : M(A : B) \rightarrow \theta_t(a : b)$$

We can compute $\texttt{HalfDiffAdd}_\psi$ formulas.
We can then perform our half ladder with the following chain:

$$\begin{array}{ccccccccc} M(A:B) & \xrightarrow{\psi} & \theta_t(a:b) & \xrightarrow{\varphi} & \theta_t(a':b') & \xrightarrow{\widetilde{\varphi}} & \cdots & \xrightarrow{\varphi \text{ or } \widetilde{\varphi}} & \theta_t(?) \\ P_0 & \longmapsto & P_1 & \longmapsto & P_2 & \longmapsto & \cdots & \longmapsto & P_\ell \end{array}$$

[5] *D. J. Bernstein*, Curve25519: New Diffie-Hellman Speed Records, 2006

Finding formulas

Acting on coordinates

With two coordinates $(X : Z)$ on E and $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(k)$:

$$M \cdot X := aX + bZ, \qquad M \cdot Z := cX + dZ.$$

Acting on coordinates

With two coordinates $(X : Z)$ on E and $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(k)$:

$$M \cdot X := aX + bZ, \qquad M \cdot Z := cX + dZ.$$

On $E \times E$, we are interested in products $X_1X_2, X_1Z_2, Z_1X_2, Z_1Z_2$:

$$M \otimes M \cdot X_1X_2 := (M \cdot X_1)(M \cdot X_2), \quad M \otimes M \cdot X_1Z_2 := (M \cdot X_1)(M \cdot Z_2),$$
$$M \otimes M \cdot Z_1X_2 := (M \cdot Z_1)(M \cdot X_2), \quad M \otimes M \cdot Z_1Z_2 := (M \cdot Z_1)(M \cdot Z_2).$$

Acting on coordinates

With two coordinates $(X : Z)$ on E and $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(k)$:

$$M \cdot X := aX + bZ, \qquad M \cdot Z := cX + dZ.$$

On $E \times E$, we are interested in products $X_1X_2, X_1Z_2, Z_1X_2, Z_1Z_2$:

$$M \otimes M \cdot X_1X_2 := (M \cdot X_1)(M \cdot X_2), \quad M \otimes M \cdot X_1Z_2 := (M \cdot X_1)(M \cdot Z_2),$$
$$M \otimes M \cdot Z_1X_2 := (M \cdot Z_1)(M \cdot X_2), \quad M \otimes M \cdot Z_1Z_2 := (M \cdot Z_1)(M \cdot Z_2).$$

If M is of order 2, we can derive easily invariants with a trace:

$$M \otimes M \cdot (X_1X_2 + M \otimes M \cdot X_1X_2) = X_1X_2 + M \otimes M \cdot X_1X_2.$$

Methodology

$\varphi : E \to E'$ a 2-isogeny with kernel $\langle T \rangle$, consider $T' \in E'[2]$.

1. Compute the homography $t_T : P \mapsto P + T$ on $\mathbb{P}^1$.
2. Take an affine lift of t_T: $[t_T]^2 = [\mathrm{id}]$ so $t_T^2 = \lambda\,\mathrm{id}$. Set $\tau_T = (t_T \otimes t_T)/\lambda$.

Methodology

$\varphi : E \to E'$ a 2-isogeny with kernel $\langle T \rangle$, consider $T' \in E'[2]$.

1. Compute the homography $t_T : P \mapsto P + T$ on $\mathbb{P}^1$.
2. Take an affine lift of t_T: $[t_T]^2 = [\text{id}]$ so $t_T^2 = \lambda\,\text{id}$. Set $\tau_T = (t_T \otimes t_T)/\lambda$.
3. Compute the action of τ_T on

$$X_{P+Q}X_{P-Q},\ X_{P+Q}Z_{P-Q},\ Z_{P+Q}X_{P-Q},\ Z_{P+Q}Z_{P-Q}.$$

4. Find invariants for this action u_1, u_2.

Methodology

$\varphi : E \to E'$ a 2-isogeny with kernel $\langle T \rangle$, consider $T' \in E'[2]$.

1. Compute the homography $t_T : P \mapsto P + T$ on $\mathbb{P}^1$.
2. Take an affine lift of t_T: $[t_T]^2 = [\mathrm{id}]$ so $t_T^2 = \lambda\,\mathrm{id}$. Set $\tau_T = (t_T \otimes t_T)/\lambda$.
3. Compute the action of τ_T on
$$X_{P+Q}X_{P-Q},\ X_{P+Q}Z_{P-Q},\ Z_{P+Q}X_{P-Q},\ Z_{P+Q}Z_{P-Q}.$$
4. Find invariants for this action u_1, u_2.
5. Similarly, compute $t_{T'}$ and $\tau_{T'}$.
6. Compute the action of $\tau_{T'}$ on
$$X_{\varphi(P)}X_{\varphi(Q)},\ X_{\varphi(P)}Z_{\varphi(Q)},\ Z_{\varphi(P)}X_{\varphi(Q)},\ Z_{\varphi(P)}Z_{\varphi(Q)}.$$
7. Find invariants for this action v_1, v_2.

Methodology

$\varphi : E \to E'$ a 2-isogeny with kernel $\langle T \rangle$, consider $T' \in E'[2]$.

1. Compute the homography $t_T : P \mapsto P + T$ on $\mathbb{P}^1$.
2. Take an affine lift of t_T: $[t_T]^2 = [\mathrm{id}]$ so $t_T^2 = \lambda\, \mathrm{id}$. Set $\tau_T = (t_T \otimes t_T)/\lambda$.
3. Compute the action of τ_T on

$$X_{P+Q}X_{P-Q},\ X_{P+Q}Z_{P-Q},\ Z_{P+Q}X_{P-Q},\ Z_{P+Q}Z_{P-Q}.$$

4. Find invariants for this action u_1, u_2.
5. Similarly, compute $t_{T'}$ and $\tau_{T'}$.
6. Compute the action of $\tau_{T'}$ on

$$X_{\varphi(P)}X_{\varphi(Q)},\ X_{\varphi(P)}Z_{\varphi(Q)},\ Z_{\varphi(P)}X_{\varphi(Q)},\ Z_{\varphi(P)}Z_{\varphi(Q)}.$$

7. Find invariants for this action v_1, v_2.
8. Use relations between points to find coefficients such that:

$$\begin{cases} u_1(P+Q, P-Q) = \alpha_1 v_1(\varphi(P), \varphi(Q)) + \alpha_2 v_2(\varphi(P), \varphi(Q)), \\ u_2(P+Q, P-Q) = \beta_1 v_1(\varphi(P), \varphi(Q)) + \beta_2 v_2(\varphi(P), \varphi(Q)). \end{cases}$$

An example: finding the translation

We will work on the theta model $\theta(a:b)$ with ramification:

$$\mathcal{O} = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (b:a), \quad T_3 = (-b:a).$$

Set $(A^2 : B^2) = (a^2 + b^2 : a^2 - b^2)$, we have the following 2-isogeny:

$$\varphi : (X:Z) \in \theta(a:b) \mapsto (B(X^2+Z^2) : A(X^2-Z^2)) \in \theta(A:B)$$

An example: finding the translation

We will work on the theta model $\theta(a : b)$ with ramification:

$$\mathcal{O} = (a : b)^*, \quad T_1 = (-a : b), \quad T_2 = (b : a), \quad T_3 = (-b : a).$$

Set $(A^2 : B^2) = (a^2 + b^2 : a^2 - b^2)$, we have the following 2-isogeny:

$$\varphi : (X : Z) \in \theta(a : b) \mapsto (B(X^2 + Z^2) : A(X^2 - Z^2)) \in \theta(A : B)$$

Given the ramification, the homography t_{T_1} is simply $t_{T_1} : (X : Z) \mapsto (-X : Z)$.

An example: finding the translation

We will work on the theta model $\theta(a:b)$ with ramification:

$$\mathcal{O} = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (b:a), \quad T_3 = (-b:a).$$

Set $(A^2 : B^2) = (a^2 + b^2 : a^2 - b^2)$, we have the following 2-isogeny:

$$\varphi : (X:Z) \in \theta(a:b) \mapsto (B(X^2+Z^2) : A(X^2-Z^2)) \in \theta(A:B)$$

Given the ramification, the homography t_{T_1} is simply $t_{T_1} : (X:Z) \mapsto (-X:Z)$. The affine lift is also given by $(X, Z) \mapsto (-X, Z)$, which is already involutive.

An example: computing invariants

- Theta model $\theta(a:b)$, with $(A^2:B^2) = (a^2+b^2:a^2-b^2)$:

$$\mathcal{O} = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (b:a), \quad T_3 = (-b:a).$$

- 2-isogeny with kernel $\langle T_1 \rangle$: $\varphi : (X:Z) \mapsto (B(X^2+Z^2) : A(X^2-Z^2))$.
- Translation $t_{T_1} : (X, Z) \mapsto (-X, Z)$, $\tau_{T_1} := t_{T_1} \otimes t_{T_1}$ acts on

$$X_1X_2,\ X_1Z_2,\ Z_1X_2,\ Z_1Z_2.$$

- $\tau_{T_1} \cdot X_1X_2 = (-X_1)(-X_2) = X_1X_2$

An example: computing invariants

- Theta model $\theta(a:b)$, with $(A^2:B^2)=(a^2+b^2:a^2-b^2)$:

$$\mathcal{O}=(a:b)^*,\quad T_1=(-a:b),\quad T_2=(b:a),\quad T_3=(-b:a).$$

- 2-isogeny with kernel $\langle T_1\rangle$: $\varphi:(X:Z)\mapsto(B(X^2+Z^2):A(X^2-Z^2))$.
- Translation $t_{T_1}:(X,Z)\mapsto(-X,Z)$, $\tau_{T_1}:=t_{T_1}\otimes t_{T_1}$ acts on

$$X_1X_2,\ X_1Z_2,\ Z_1X_2,\ Z_1Z_2.$$

- $\tau_{T_1}\cdot X_1X_2=(-X_1)(-X_2)=X_1X_2$
- $\tau_{T_1}\cdot X_1Z_2=-X_1Z_2$
- $\tau_{T_1}\cdot Z_1X_2=-Z_1X_2$
- $\tau_{T_1}\cdot Z_1Z_2=Z_1Z_2$

An example: computing invariants

- Theta model $\theta(a:b)$, with $(A^2:B^2) = (a^2+b^2:a^2-b^2)$:

$$\mathcal{O} = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (b:a), \quad T_3 = (-b:a).$$

- 2-isogeny with kernel $\langle T_1 \rangle$: $\varphi : (X:Z) \mapsto (B(X^2+Z^2) : A(X^2-Z^2))$.
- Translation $t_{T_1} : (X, Z) \mapsto (-X, Z)$, $\tau_{T_1} := t_{T_1} \otimes t_{T_1}$ acts on

$$X_1X_2,\ X_1Z_2,\ Z_1X_2,\ Z_1Z_2.$$

- $\tau_{T_1} \cdot X_1X_2 = (-X_1)(-X_2) = X_1X_2$
- $\tau_{T_1} \cdot X_1Z_2 = -X_1Z_2$
- $\tau_{T_1} \cdot Z_1X_2 = -Z_1X_2$
- $\tau_{T_1} \cdot Z_1Z_2 = Z_1Z_2$

Two invariants

$$X_1X_2,\ Z_1Z_2$$

An example: relations

$$u_1(P+Q, P-Q) = X_{P+Q}X_{P-Q}, \quad u_2(P+Q, P-Q) = Z_{P+Q}Z_{P-Q}.$$

An example: relations

$$u_1(P+Q, P-Q) = X_{P+Q}X_{P-Q}, \quad u_2(P+Q, P-Q) = Z_{P+Q}Z_{P-Q}.$$

Similarly:

$$v_1(\varphi(P), \varphi(Q)) = X_{\varphi(P)}X_{\varphi(Q)}, \quad v_2(\varphi(P), \varphi(Q)) = Z_{\varphi(P)}Z_{\varphi(Q)}.$$

The theory gives the existence of coefficients $\alpha_1, \alpha_2, \beta_1, \beta_2 \in k$ such that:

$$\begin{cases} X_{P+Q}X_{P-Q} = \alpha_1 X_{\varphi(P)}X_{\varphi(Q)} + \alpha_2 Z_{\varphi(P)}Z_{\varphi(Q)}, \\ Z_{P+Q}Z_{P-Q} = \beta_1 X_{\varphi(P)}X_{\varphi(Q)} + \beta_2 Z_{\varphi(P)}Z_{\varphi(Q)}. \end{cases}$$

An example: relations

$$u_1(P+Q, P-Q) = X_{P+Q}X_{P-Q}, \quad u_2(P+Q, P-Q) = Z_{P+Q}Z_{P-Q}.$$

Similarly:

$$v_1(\varphi(P), \varphi(Q)) = X_{\varphi(P)}X_{\varphi(Q)}, \quad v_2(\varphi(P), \varphi(Q)) = Z_{\varphi(P)}Z_{\varphi(Q)}.$$

The theory gives the existence of coefficients $\alpha_1, \alpha_2, \beta_1, \beta_2 \in k$ such that:

$$\begin{cases} X_{P+Q}X_{P-Q} = \alpha_1 X_{\varphi(P)}X_{\varphi(Q)} + \alpha_2 Z_{\varphi(P)}Z_{\varphi(Q)}, \\ Z_{P+Q}Z_{P-Q} = \beta_1 X_{\varphi(P)}X_{\varphi(Q)} + \beta_2 Z_{\varphi(P)}Z_{\varphi(Q)}. \end{cases}$$

For instance, with $P = Q = \mathcal{O}$:

- $P + Q = P - Q = \mathcal{O} = (a : b)$,
- $\varphi(P) = \varphi(Q) = \mathcal{O}' = (A : B)$.

$$\begin{cases} a^2 = \alpha_1 A^2 + \alpha_2 B^2, \\ b^2 = \beta_1 A^2 + \beta_2 B^2. \end{cases}$$

An example: relations

$$u_1(P+Q,P-Q)=X_{P+Q}X_{P-Q},\quad u_2(P+Q,P-Q)=Z_{P+Q}Z_{P-Q}.$$

Similarly:

$$v_1(\varphi(P),\varphi(Q))=X_{\varphi(P)}X_{\varphi(Q)},\quad v_2(\varphi(P),\varphi(Q))=Z_{\varphi(P)}Z_{\varphi(Q)}.$$

The theory gives the existence of coefficients $\alpha_1,\alpha_2,\beta_1,\beta_2\in k$ such that:

$$\begin{cases} X_{P+Q}X_{P-Q}=\alpha_1X_{\varphi(P)}X_{\varphi(Q)}+\alpha_2Z_{\varphi(P)}Z_{\varphi(Q)},\\ Z_{P+Q}Z_{P-Q}=\beta_1X_{\varphi(P)}X_{\varphi(Q)}+\beta_2Z_{\varphi(P)}Z_{\varphi(Q)}.\end{cases}$$

For instance, with $P=Q=\mathcal{O}$:

- $P+Q=P-Q=\mathcal{O}=(a:b)$,
- $\varphi(P)=\varphi(Q)=\mathcal{O}'=(A:B)$.

$$\begin{cases} a^2=\alpha_1A^2+\alpha_2B^2,\\ b^2=\beta_1A^2+\beta_2B^2.\end{cases}$$

By using various combinations of (P,Q), we end up finding $\alpha_1=\alpha_2=\beta_1=-\beta_2$.

Future work and research direction

What's new?

- Isogeny in dimension 2 to gain new formulas in dimension 1: `HalfDiffAdd`.
- Half ladder: enhanced pre-computation cost, close to Montgomery ladder in best case scenario.

Work in progress

Generalizing half ladder to dimension 2 to improve arithmetic.

Code available here:
`https://gitlab.inria.fr/nsarkis/half-diff-add`.

Figure: eprint 2024/1582