Halving differential additions on Kummer lines

Damien Robert, *Nicolas Sarkis*

Institut de Mathématiques de Bordeaux, CANARI INRIA team

May 5th, 2025 – Eurocrypt, Madrid

Motivation

Figure: An RFID tag

- 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

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

- Short Weierstrass (general case):

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

- Montgomery curves:

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

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

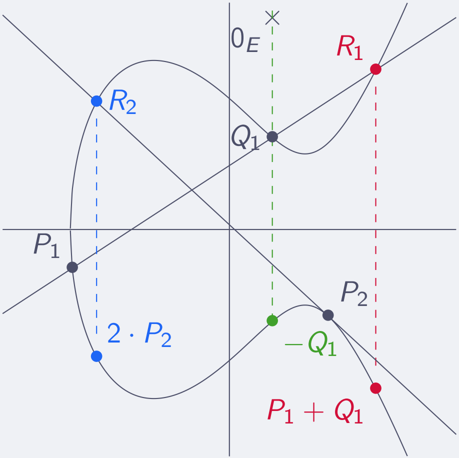

Figure: An elliptic curve

Kummer line of a Montgomery curve

$$E : Y^2 Z = X(X^2 + \mathcal{A}XZ + Z^2)$$

If $P = (X : Y : Z)$, then $-P = (X : -Y : Z)$; $0_E = (0 : 1 : 0)$.

Montgomery XZ-coordinates

$$x : E \to \mathbb{P}^1, \begin{cases} 0_E & \mapsto \infty := (1 : 0), \\ (X : Y : Z) & \mapsto \frac{X}{Z} := (X : Z). \end{cases}$$

We have $x(P) = x(-P)$:

- $\#x^{-1}(X : Z) = 1$ when $P = -P$ (i. e. $2 \cdot P = 0_E$, 4 points);
- $\#x^{-1}(X : Z) = 2$ otherwise.

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 $\varphi : \mathbb{P}^1 \to \mathbb{P}^1$ maps the ramification points to ramification points.

Examples

Montgomery curve $y^2 = x(x^2 + \mathcal{A}x + 1)$, $\alpha^2 + \mathcal{A}\alpha + 1 = 0$, $\alpha \in \overline{k}$

$$\pi : \begin{cases} 0_E & \mapsto (1:0), \\ (x,y) & \mapsto (x:1). \end{cases}$$

$$0_E = (1:0)^*, \quad T_1 = (0:1), \quad T_2 = (\alpha:1), \quad T_3 = (1:\alpha).$$

Examples

Montgomery curve $y^2 = x(x^2 + \mathcal{A}x + 1)$, $\alpha^2 + \mathcal{A}\alpha + 1 = 0$, $\alpha \in \overline{k}$

$$\pi : \begin{cases} 0_E & \mapsto (1:0), \\ (x,y) & \mapsto (x:1). \end{cases}$$

$$0_E = (1:0)^*, \quad T_1 = (0:1), \quad T_2 = (\alpha:1), \quad T_3 = (1:\alpha).$$

Theta model $\theta(a:b)$: $y^2 = x(x - A^2/B^2)(x - B^2/A^2)$, $A/B \in k$

$$(a^2 : b^2) = (A^2 + B^2 : A^2 - B^2), \ a/b \in k.$$

$$\pi : \begin{cases} 0_E & \mapsto (a:b), \\ (X:Y:Z) & \mapsto (a(X-Z) : b(X+Z)). \end{cases}$$

$$0_E = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (b:a), \quad T_3 = (-b:a).$$

What about the group law?

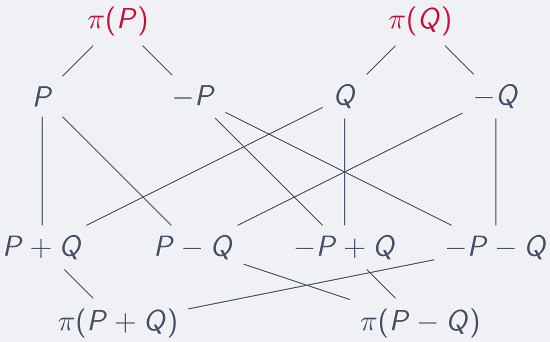

Figure: Two possible choices

What about the group law?

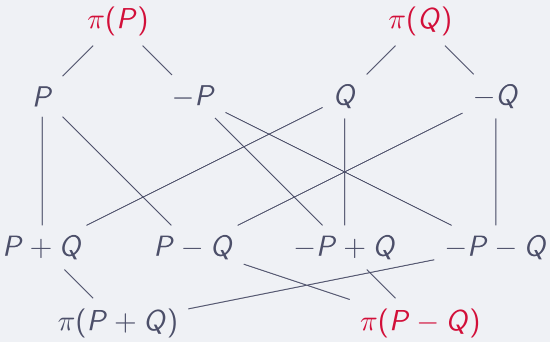

Figure: Two possible choices

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". In: *Mathematics of Computation* 48 (1987), pp. 243–264.

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

Differential addition $(3M + 4S + 1m_0)$

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

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

Doubling $(4S + 2m_0)$

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

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

[2] P. Gaudry and D. Lubicz. "The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines". In: *Finite Fields Their Appl.* 15.2 (2009), pp. 246–260.

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);
```

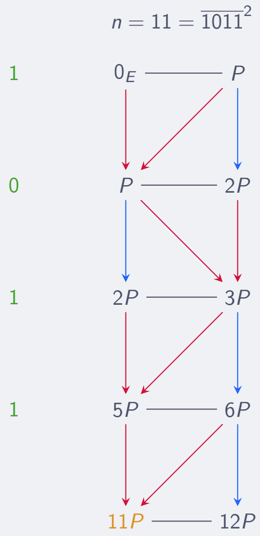

Figure: Montgomery ladder

2-isogenies

Definition

- Isogeny: surjective morphism $\varphi : E \to E'$ with finite kernel.
- 2-isogeny: $\ker \varphi = \{0_E, T\}$, where $2 \cdot T = 0_E$.
- It always comes with a dual $\widetilde{\varphi} : E' \to E$ such that:

$$\widetilde{\varphi} \circ \varphi = [2]_E \text{ and } \varphi \circ \widetilde{\varphi} = [2]_{E'}.$$

2-isogenies

Definition

- Isogeny: surjective morphism $\varphi : E \to E'$ with finite kernel.
- 2-isogeny: $\ker \varphi = \{0_E, T\}$, where $2 \cdot T = 0_E$.
- It always comes with a dual $\widetilde{\varphi} : E' \to E$ such that:

$$\widetilde{\varphi} \circ \varphi = [2]_E \text{ and } \varphi \circ \widetilde{\varphi} = [2]_{E'}.$$

Doubling formulas

- Computing $2 \cdot P$: with 2-isogenies.
- We know how to find 2-isogenies formulas on Kummer lines[3].

[3] D. Robert and N. S. "Computing 2-isogenies between Kummer lines". In: *Communications in Cryptology* 1.1 (2024), p. 26

Half ladder

The differential addition isogeny

$$\begin{aligned} F : E \times E &\to E \times E \\ (P, Q) &\mapsto (P + Q, P - Q) \end{aligned}$$

The differential addition isogeny

$$
\begin{aligned}
F : E \times E &\to E \times E \\
(P, Q) &\mapsto (P + Q, P - Q)
\end{aligned}
$$

$$
\begin{tikzcd}
E \arrow[dd, "{[2]_E}"'] \arrow[dr, "\varphi"] & \\
& E' \arrow[dl, "\widetilde{\varphi}"] \\
E &
\end{tikzcd}
$$

Figure: Factoring doubling

The differential addition isogeny

$$\begin{aligned} F : E \times E &\to E \times E \\ (P, Q) &\mapsto (P + Q, P - Q) \end{aligned}$$

$$\begin{aligned} \Phi : E \times E &\to E' \times E' \\ (P, Q) &\mapsto (\varphi(P), \varphi(Q)) \end{aligned}$$

$$\begin{tikzcd} E \arrow[dd, "{[2]_E}"'] \arrow[dr, "\varphi"] & \\ & E' \arrow[dl, "\widetilde{\varphi}"] \\ E & \end{tikzcd}$$

Figure: Factoring doubling

The differential addition isogeny

$$\begin{aligned} F : E \times E &\to E \times E \\ (P, Q) &\mapsto (P + Q, P - Q) \end{aligned}$$

$$\begin{aligned} \Phi : E \times E &\to E' \times E' \\ (P, Q) &\mapsto (\varphi(P), \varphi(Q)) \end{aligned}$$

$$\begin{tikzcd} E \times E \arrow[r, "\Phi"] \arrow[d, "F"'] & E' \times E' \arrow[dl, dashed, "?"] \\ E \times E & \end{tikzcd}$$

Figure: Factoring diff. add. ?

$$\begin{tikzcd} E \arrow[dr, "\varphi"] \arrow[dd, "{[2]_E}"'] & \\ & E' \arrow[dl, "\widetilde{\varphi}"] \\ E & \end{tikzcd}$$

Figure: Factoring doubling

The differential addition isogeny

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

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

$$\begin{tikzcd} E \times E \arrow[r, "\Phi"] \arrow[d, "F"'] & E' \times E' \arrow[d, dashed] \\ E \times E \arrow[r, dashed] & A \end{tikzcd}$$

Figure: Factoring diff. add. ? Not as easy

$$\begin{tikzcd} E \arrow[dr, "\varphi"] \arrow[dd, "{[2]_E}"'] & \\ & E' \arrow[dl, "\widetilde{\varphi}"] \\ E & \end{tikzcd}$$

Figure: Factoring doubling

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, $2 \cdot P = \mathtt{HalfDouble}_{\varphi}(\varphi(P))$ $(= \widetilde{\varphi}(\varphi(P)))$.

Example on $\theta(a:b)$

$$0_E = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (b:a), \quad T_3 = (-b:a).$$

Set $(A^2 : B^2) := (a^2+b^2 : a^2-b^2)$, assume $A/B \in k$. The 2-isogeny considered is:

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

Example on $\theta(a:b)$

$$0_E = (a:b)^*, \quad T_1 = (-a:b), \quad T_2 = (b:a), \quad T_3 = (-b:a).$$

Set $(A^2 : B^2) := (a^2+b^2 : a^2-b^2)$, assume $A/B \in k$. The 2-isogeny considered is:

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

$\texttt{HalfDiffAdd}_\varphi(\varphi(P), \varphi(Q), P-Q)$ $(4M)$

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

A full differential addition in $\theta(a:b)$ is $3M + 4S + 1m_0$.

Main idea

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.

Main idea

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 $P \in \mathcal{K}$.

- $n = (b_{\ell-1}, b_{\ell-2}, \ldots, b_0)_2$ has ℓ bits;
- $P_0 := P$ and $\mathcal{K}_0 := \mathcal{K}$;
- We have $\mathcal{K}_1, \ldots, \mathcal{K}_\ell$ Kummer lines and $\varphi_i : \mathcal{K}_{i-1} \to \mathcal{K}_i$ 2-isogenies;
- $P_i := \varphi_i(P_{i-1})$.

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

Figure: Successive images

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

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);
```

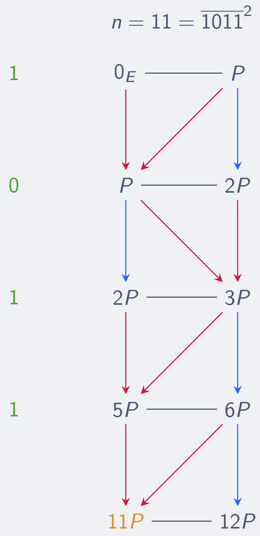

Figure: Montgomery ladder

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);

$n = 11 = \overline{1011}^2$

$$\begin{tikzcd}
 & P_0 \arrow[d, mapsto, "\varphi_1"] \\
 & P_1 \arrow[d, mapsto, "\varphi_2"] \\
 & P_2 \arrow[d, mapsto, "\varphi_3"] \\
 & P_3 \arrow[d, mapsto, "\varphi_4"] \\
0_{E_4} \arrow[r, dash] & P_4 \quad P_4
\end{tikzcd}$$

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);

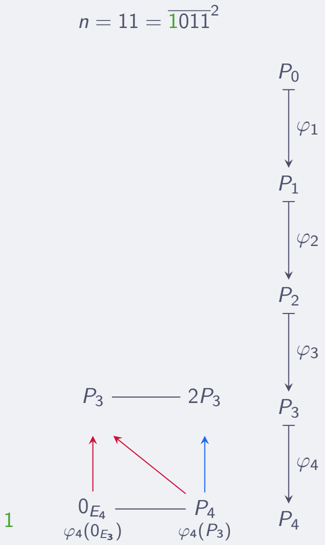

Figure: Half ladder

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);
```

$n = 11 = \overline{1011}^2$

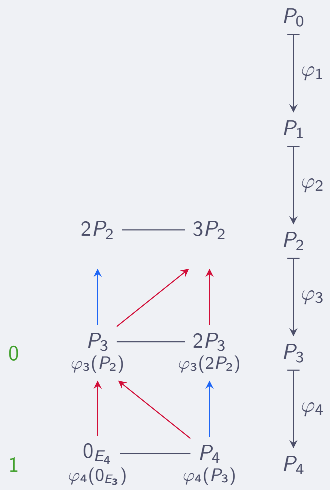

Figure: Half ladder

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);

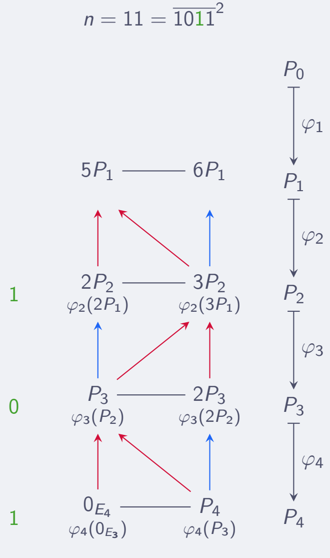

Figure: Half ladder

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);

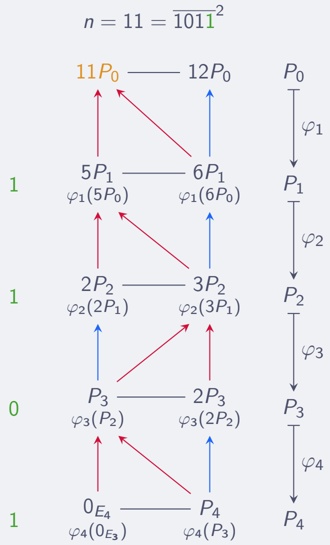

Figure: Half ladder

Computational cost: without pre-computation

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: without pre-computation

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 cost per bit with no pre-computation

Computational cost: with pre-computation

Algorithm	Pre-computation	Step
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 et al. "How to (Pre-)Compute a Ladder - Improving the Performance of X25519 and X448". In: *SAC 2017*. Vol. 10719. Lecture Notes in Computer Science. Aug. 2017, pp. 172–191

Computational cost: with pre-computation

Algorithm	Pre-computation	Step
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

Still holds on a Montgomery curve with full 2-torsion (with a few tweaks)

$$\begin{cases} y^2 = x(x - A^2/B^2)(x - B^2/A^2), \\ \sqrt{\frac{A^2+B^2}{A^2-B^2}} \in k, \end{cases} \rightsquigarrow y^2 = x(x - a/b)(x - b/a).$$

[4] T. Oliveira et al. "How to (Pre-)Compute a Ladder - Improving the Performance of X25519 and X448". In: *SAC 2017*. Vol. 10719. Lecture Notes in Computer Science. Aug. 2017, pp. 172–191

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`.