Curve25519

Inwiefern tragen die mathematischen Eigenschaften von Curve25519 zu ihrer Sicherheit und Effizienz bei?

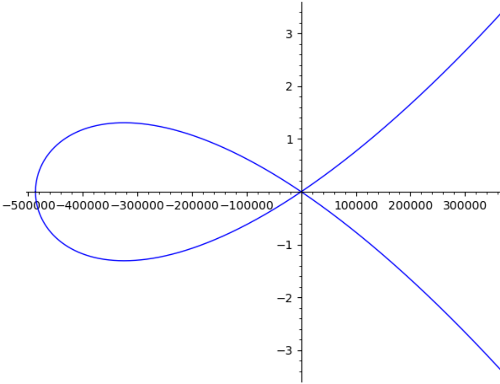
Tom Pilgram

April 2025

Gruppenstruktur

$$\mathcal{E} : y^2 = x^3 + 486662x^2 + x$$

$$\begin{aligned}
\mathcal{E}(\mathbb{F}_{p^2}) &= \{\mathcal{O}\} \cup \{(x, y) \in \mathbb{F}_{p^2} : y^2 = x^3 + Ax^2 + x\} \\
A &= 486662 \\
p &= 2^{255} - 19 \\
\mathcal{E}'(\mathbb{F}_{p^2}) &= \{\mathcal{O}\} \cup (\mathcal{E}(\mathbb{F}_{p^2}) \cap (\mathbb{F}_p \times \mathbb{F}_p)) \\
\#\mathcal{E}'(\mathbb{F}_{p^2}) &= 8l \\
l &= \#\langle P \rangle \\
&= 2^{252} + \mathtt{0x14def9dea2f79cd65812631a5cf5d3ed} \\
X(P) &= 9
\end{aligned}$$



Gruppenstruktur

Gruppenoperationen.
Seien $P, Q \in \mathcal{E}(\mathbb{F}_{p^2})$:

$$
\begin{aligned}
\text{Neutrales Element} :&\ \mathcal{O}\,\text{(Punkt im Unendlichen)}\\
\text{Punktaddition} :&\ P \oplus Q\\
\text{Inverses Element} :&\ \ominus(x,y) = (x,-y)\\
&\ P \oplus (\ominus P) = P \ominus P = \mathcal{O}\\
\text{Skalarprodukt} :&\ [k]P = \underbrace{P \oplus \cdots \oplus P}_{k-Mal}
\end{aligned}
$$

X25519

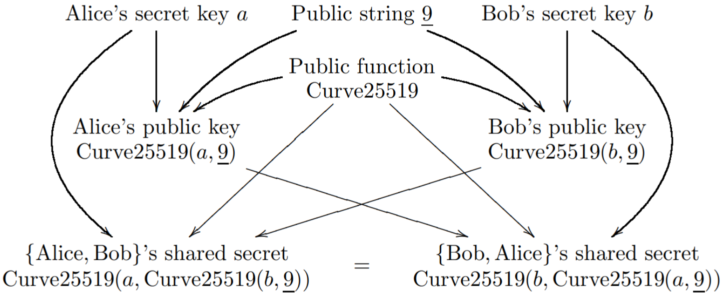
Definition: X25519.

$$\mathcal{K}_{pr} = \{0, 8, 16, 24, \ldots, 248\} \times \{0, 1, \ldots, 255\}^{30} \times \{64, 65, 66, \ldots, 127\}$$

$$\mathcal{K}_{pub} = \{0, 1, \ldots, 255\}^{32}$$

$$X25519 : \mathcal{K}_{pr} \times \mathcal{K}_{pub} \longrightarrow \mathcal{K}_{pub}$$

$$(n, q) \longmapsto X([n]Q) \text{ mit } X(Q) = q$$



$$By^2 = x^3 + Ax^2 + x$$

Definition: Montgomery Curve.

$$\mathcal{E}_{(A,B)} : By^2 = x^3 + Ax^2 + x, \quad B(A^2 - 4) \neq 0$$

Punktoperationen $P = (x_P, y_P),\ Q = (x_Q, y_Q)$

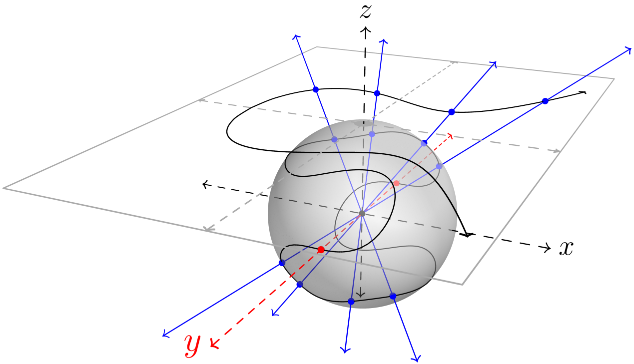
Punktverdopplung

$$x_{[2]P} = B\lambda^2 - 2x_P - A$$
$$y_{[2]P} = \lambda(x_P - x_{[2]P}) - y_P$$
$$\lambda = (3x_P^2 + 2Ax_P + 1)/2By_P$$

Punktaddition

$$x_\oplus = B\lambda^2 - (x_P + x_Q) - A$$
$$y_\oplus = \lambda(x_P - x_\oplus) - y_P$$
$$\lambda = (y_Q - y_P)/(x_Q - x_P)$$

$\mathbb{P}^2(\mathbb{K})$



Definition:
Projektiver Raum.

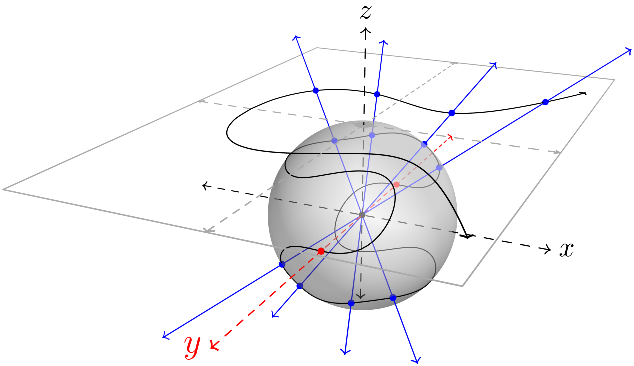
$$\mathbb{P}^n(\mathbb{K}) = \left(\mathbb{K}^{n+1} \setminus \{0\}\right) / \sim$$

mit der Äquivalenzrelation

$$\sim: x \sim y \Leftrightarrow \exists \lambda \in \mathbb{K} \setminus \{0\} : x = \lambda y.$$

$P = (X : Y : Z) \in \mathbb{P}^2(\mathbb{K})$
$(X : Y : Z) \sim (\lambda X : \lambda Y : \lambda Z)$

$\mathbb{P}^2(\mathbb{K})$



$$(x,y) = \left(\frac{X}{Z}, \frac{Y}{Z}\right)$$

$$\mathcal{E}_{(A,B)} : BY^2Z = X^3 + AX^2Z + XZ^2 \subseteq \mathbb{P}^2$$

(i) $Z \neq 0$
Alle Punkte der affinen Kurve liegen auf der Ebene $(x : y : 1)$
(ii) $Z = 0$
$0 = X^3 \Leftrightarrow X = 0$ und Y beliebig
$\mathcal{O} = (0 : 1 : 0)$

Montgomery Ladder

Definition: Montgomery Ladder.
Algorithmus zur Skalarmultiplikation auf Montgomery-Kurven:

- Berechnung nur mit x-Koordinaten
- Konstante Laufzeit

$$\mathbf{x} : \mathcal{E} \to \mathcal{E}/\langle \ominus \rangle = \mathbb{P}^1$$

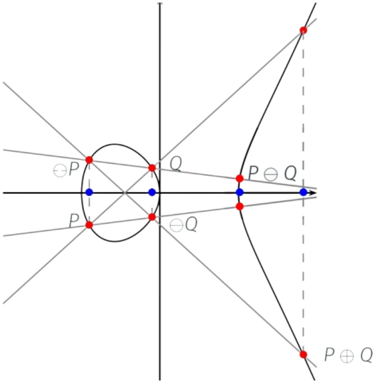
$$\mathbf{x} : P \longmapsto \begin{cases} (X_P : 1) & \text{für } P = (X_P : Y_P : 1), \\ (1 : 0) & \text{für } P = \mathcal{O} = (0 : 1 : 0). \end{cases}$$

Montgomery Ladder

Definition: Pseudo-Operationen.

$$\texttt{xDBL}: \mathbf{x}(P) \longmapsto \mathbf{x}([2]P)$$

$$\texttt{xADD}: (\mathbf{x}(P), \mathbf{x}(Q), \mathbf{x}(P \ominus Q)) \longmapsto \mathbf{x}(P \oplus Q)$$



Montgomery Ladder

xDBL

$$
\begin{aligned}
x_{[2]P} &= B\lambda^2 - 2x_P - A \\
&= \frac{(3x_P + 2Ax_P + 1)^2}{4By_P^2} \\
&\quad - 2x_P - A \\
&= \frac{(x_P^2 - 1)^2}{4(x_P^3 + Ax_P^2 + x_P)}
\end{aligned}
$$

xADD

$$
\begin{aligned}
& x_{P\oplus Q} x_{P\ominus Q} \\
&= \left(B \left(\frac{(y_Q - y_P)^2}{(x_Q - x_P)^2} \right) - (x_P + x_Q) - A \right) \\
&\quad \cdot \left(B \left(\frac{(-y_Q - y_P)^2}{(x_Q - x_P)^2} \right) - (x_P + x_Q) - A \right) \\
&= \frac{(x_Q x_P - 1)^2}{(x_Q - x_P)^2}
\end{aligned}
$$

- Inversionen sind in endlichen Körpern rechenintensiv
- Projektive Koordinaten vermeiden direkte Inversion
- Darstellung der x-Koordinate als Bruch $\frac{X}{Z}$
- Rückwandlung in die affine Koordinate durch $x = XZ^{p-2}$

Montgomery Ladder

Notation

$$(X_P : Z_P) := \mathbf{x}(P), \quad (X_Q : Z_Q) := \mathbf{x}(Q)$$

$$(X_\oplus : Z_\oplus) := \mathbf{x}(P \oplus Q), \quad (X_\ominus : Z_\ominus) := \mathbf{x}(P \ominus Q)$$

xDBL

$$X_{[2]P} = (X_P^2 - Z_P^2)^2$$
$$Z_{[2]P} = 4X_P Z_P (X_P^2 + AX_P Z_P + Z_P^2)$$

xADD

$$X_\oplus = 4(X_P X_Q - Z_P Z_Q) Z_\ominus$$
$$Z_\oplus = 4(X_P Z_Q - Z_P X_Q) X_\ominus$$

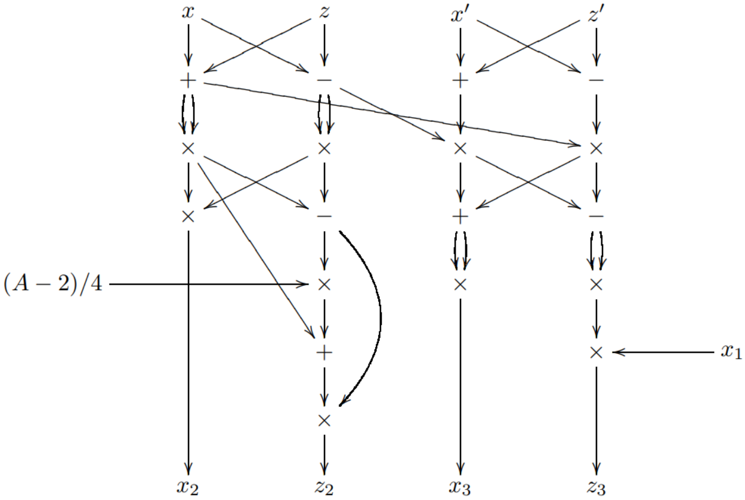
Montgomery Ladder

xDBL

$$X_{[2]P} = (X_P + Z_P)^2 (X_P - Z_P)^2$$
$$Z_{[2]P} = (4X_P Z_P)((X_P - Z_P)^2 + ((A+2)/4)(4X_P Z_P))$$

xADD

$$X_{\oplus} = Z_{\ominus}[(X_P - Z_P)(X_Q + Z_Q) + (X_P + Z_P)(X_Q - Z_Q)]^2$$
$$Z_{\oplus} = X_{\ominus}[(X_P - Z_P)(X_Q + Z_Q) - (X_P + Z_P)(X_Q - Z_Q)]^2$$



Montgomery Ladder

Algorithmus: Montgomery Ladder

Input: (1) $k = \sum_{i=0}^{l-1} k_i 2^i$ mit $k_{l-1} = 1$ `//Binärdarstellung des Skalars`
(2) (X_P, Z_P), so dass $(X_P : Z_P) = \mathbf{x}(P)$

Output: (X_k, Z_k), so dass $(X_k : Z_k) = \mathbf{x}([k]P)$

$(\mathtt{x}_0,\mathtt{x}_1) \leftarrow ((X_P, Z_P), \mathtt{xDBL}(X_P, Z_P))$ $\quad$ `//`$\mathtt{x}_1$`-`$\mathtt{x}_0$ `ist immer` (X_P, Z_P)
for $i = l-2$ **downto** 0 **do**
  if $k_i = 0$ **then**
    $(\mathtt{x}_0,\mathtt{x}_1) \leftarrow (\mathtt{xDBL}(\mathtt{x}_0), \mathtt{xADD}(\mathtt{x}_0, \mathtt{x}_1, (X_P, Z_P)))$
  else
    $(\mathtt{x}_0,\mathtt{x}_1) \leftarrow (\mathtt{xADD}(\mathtt{x}_0, \mathtt{x}_1, (X_P, Z_P)), \mathtt{xDBL}(\mathtt{x}_1))$
return $\mathtt{x}_0$ $\quad$ `//`$\mathtt{x}_0 = \mathbf{x}([k]P)$, $\mathtt{x}_0 = \mathbf{x}([k+1]P)$

SCA

Definition: Side-channel attack.
Angriff, der physische Informationen wie Zeit oder Stromverbrauch nutzt, um geheime Daten aus einem kryptographischen System zu erhalten

Beispiele Schwachstellen:

- Verzweigungen basierend auf geheimen Bits
  z.B. if($\mathtt{b}_i == 0$): mit `b` geheim
- Speicherzugriff mit einem geheimen Index z.B. $\mathtt{x} = \mathtt{T}[\mathtt{b}]$ mit `b` geheim

Montgomery Ladder

SWAP() Bedingter Tausch in konstanter Zeit

Input: (1) $b \in \{0, 1\}$
(2) `(x`$_0$`,x`$_1$`)` `//x`$_0$ `und x`$_1$ `als n-bit Strings`

Output: `(x`$_b$`,x`$_{b-1}$`)`

1: `b` $\leftarrow (b, \ldots, b)_n$
2: `v` $\leftarrow$ `b` $\wedge$ `(x`$_0$ `xor x`$_1$`)`
3: **return** `(x`$_0$ `xor v, x`$_1$ `xor v)`

Montgomery Ladder

Algorithmus: Montgomery Ladder mit `SWAP()`

Input: (1) $k = \sum_{i=0}^{l-1} k_i 2^i$ mit $k_{l-1} = 1$
(2) (X_P, Z_P), so dass $(X_P : Z_P) = \mathbf{x}(P)$

Output: (X_k, Z_k), so dass $(X_k : Z_k) = \mathbf{x}([k]P)$

1: $(\mathtt{x}_0,\mathtt{x}_1) \leftarrow ((X_P, Z_P), \mathtt{xDBL}(X_P, Z_P))$
2: **for** $i = l - 2$ **downto** 0 **do**
3: $\quad (\mathtt{x}_0,\mathtt{x}_1) \leftarrow \mathtt{SWAP}((k_{i+1}\ \mathtt{xor}\ k_i), (\mathtt{x}_0,\mathtt{x}_1))$
4: $\quad (\mathtt{x}_0,\mathtt{x}_1) \leftarrow (\mathtt{xDBL}(\mathtt{x}_0), \mathtt{xADD}(\mathtt{x}_0, \mathtt{x}_1, (X_P, Z_P)))$
5: $(\mathtt{x}_0,\mathtt{x}_1) \leftarrow \mathtt{SWAP}(k_0, (\mathtt{x}_0,\mathtt{x}_1))$
6: **return** $\mathtt{x}_0$

$\mathbb{F}_{p^2}$

Definition: $\mathbb{F}_{p^2}$.
Sei δ das kleinste nicht-quadratische Element in $\mathbb{F}_p$.
Dann ist die quadratische Erweiterung von $\mathbb{F}_p$ definiert als

$$\mathbb{F}_{p^2} = \mathbb{F}_p(\sqrt{\delta}) = \{a + b\sqrt{\delta} \mid a, b \in \mathbb{F}_p\}.$$

Bemerkung:

- Ganau $(p-1)/2$ nicht-quadratische Zahlen in $\mathbb{F}_p$
- α ist nicht-quadratisch $\Leftrightarrow$ α/δ ist eine Quadratzahl

$\mathbb{F}_{p^2}$

Reguläre Kurve:
$\mathcal{E}_A : y^2 = x^3 + Ax^2 + x \quad$ über $\mathbb{F}_p$
$\mathcal{E}'(\mathbb{F}_{p^2}) = \{\mathcal{O}\} \cup (\mathcal{E}(\mathbb{F}_{p^2}) \cap (\mathbb{F}_p \times \mathbb{F}_p))$ $\mathcal{E}_A : y^2 = x^3 + Ax^2 + x$

Quadratischer Twist:
$\mathcal{E}_{(A,B)} : By^2 = x^3 + Ax^2 + x \quad$ über $\mathbb{F}_p$ (B nicht-quadratisch)
$\mathcal{E}''(\mathbb{F}_{p^2}) = \{\mathcal{O}\} \cup (\mathcal{E}(\mathbb{F}_{p^2}) \cap (\mathbb{F}_p \times \sqrt{\delta}\mathbb{F}_p))$

$$\mathcal{E}(\mathbb{F}_{p^2}) = \mathcal{E}'(\mathbb{F}_{p^2}) \cup \mathcal{E}''(\mathbb{F}_{p^2})$$

Montgomery Ladder auf der regulären Kurve und
ihrem quadratischen Twist gleich, da sie nicht von B abhängig ist

$\mathcal{K}_{pub}$

$$\mathcal{K}_{pub} = \{0, 1, \ldots, 255\}^{32}$$

keine Validierung des öffentlichen Schlüssels notwendig

$2^{255} - 19$

Bedingungen:

- Primzahl nahe einer Zweierpotenz
  - Schnelle Modulo-Berechnungen
- Knapp unter 32k Bits (k: Ganzzahl)
  - Öffentlicher Schlüssel kann als 32-bit Words mit geringer Speicherverschwendung übermittelt werden

Optionen:

$$\{2^{255} + 95, 2^{255} - 19, 2^{255} - 31, 2^{254} + 79, 2^{253} + 51, 2^{253} + 39\}$$

$\mathbf{2^{255}}$**-19**, da 19 kleiner als 31,39,51,79,95

$X(P) = 9$

Satz: Satz von Lagrange. Sei G eine endliche Gruppe und H eine Untergruppe von G. Dann gilt

$$\#H \mid \#G.$$

Für jeden Punkt $Q \in \mathcal{E}'(\mathbb{F}_{p^2})$ gilt folglich:

$$\#\langle Q \rangle \in \{1, 2, 4, 8, l, 2l, 4l, 8l\}$$

$$(\#\mathcal{E}'(F_{p^2}) = 8l)$$

- $1, 2, 4, 8$ zu klein
- Vielfache von l für den Pohlig-Hellman-Algorithmus anfällig
  $\rightarrow$nur l geeignet

$$\boxed{\begin{aligned} X(P) &= \min\{x \mid Q = (x, y) \in \mathcal{E}'(\mathbb{F}_{p^2}),\ \#\langle Q \rangle = l\} \\ &= 9 \end{aligned}}$$

SSCA

Definition: Small subgroup confinement attack.
Angriff, der kleine Untergruppen nutzt, um Teile des privaten Schlüssels zu enthüllen

Beispiele:

- Pohlig-Hellman-Algorithmus: Nur wenn $\#\langle P\rangle$ eine zusammengesetzte Zahl ist
- Ungültiger Generator: Bewusste Auswahl eines Generator aus einer kleinen Untergruppe

$$\begin{aligned}\text{Öffentlicher Schlüssel von B}&: H\text{, Generator einer kleinen Untergruppe}\\ \text{Gemeinsamer Schlüssel von A}&: aH\text{, welcher ebenfalls in der Untergruppe liegt}\end{aligned}$$

$$a \mod \#\langle H\rangle$$

$\mathcal{K}_{pr}$

CLAMP() Anpassung des Skalars für X25519

Input: $k \in \{0, 1, \ldots, 255\}^{32}$ `// 32-Byte zufällig generierter Wert`

Output: Angepasster Skalar k'

```
1:  k[0] ← k[0] ∧ 248        // 248 = (11111000)₂
2:  k[31] ← k[31] ∧ 127      // 127 = (01111111)₂
3:  k[31] ← k[31] ∨ 64       //  64 = (01000000)₂
4:  k' ← k
5:  return k'
```

$\mathcal{K}_{pr}$

`1:  k[0]` $\wedge$ `(11111000)`$_2$	Niedrigstwertige 3 Bits $\leftarrow 0$	$8 \mid \mathsf{k}$
`2: k[31]` $\wedge$ `(01111111)`$_2$	Das höchstwertige Bit $\leftarrow 0$	$\mathsf{k} \leq 2^{255} - 1$
`3: k[31]` $\vee$ `(01000000)`$_2$	Das zweithöchstwertige Bit $\leftarrow 1$	$\mathsf{k} \geq 2^{254}$

$$
\begin{aligned}
\mathcal{K}_{pr} &= \{0, 8, 16, 24, \cdots, 248\} \times \{0, 1, \cdots, 255\}^{30} \times \{64, 65, 66, \cdots, 127\} \\
&= \{\underline{n} : n \in 2^{254} + 8\{0, 1, \ldots, 2^{251} - 1\}\}
\end{aligned}
$$

- SSCA: $\mathsf{k} \equiv 0 \mod 2^i$ für $1 \leq i \leq 3$
- Vermeidung von kleinen Skalaren
- Skalar zwischen 2^{251} und $2^{252} - 1 (\approx \#\langle P \rangle)$
  (Verschiebung von 3 Bits durch Faktor 8) $\rightarrow 2^{254} \leq \mathsf{k} \leq 2^{255} - 1$
- SCA: Feste effektive Schlüssellänge von 255 Bits erzwingt konstante Iterationen

486662

Bedingungen:

- $(A-2)/4$ kleine Ganzzahl (Montgomery Ladder)
- $\#\mathcal{E}'(\mathbb{F}_{p^2}) = 8l_1$
- $\#\mathcal{E}''(\mathbb{F}_{p^2}) = 4l_2$

Optionen:

$$\{358990, 464586, 486662\}$$

486662, da sonst l_1 oder l_2 kleiner als 2^{252}

Quellen

[1] Daniel J. Bernstein, Curve25519: new Diffie-Hellman speed records, 2006.

[2] Craig Costello and Benjamin Smith, Montgomery curves and their arithmetic: The case of large characteristic fields, 2017.

[3] Daniel J. Bernstein and Tanja Lange, Montgomery curves and the Montgomery ladder, 2017.

[4] Risen Crypto, Clamping & Cofactor clearing in Curve25519, 2022, https://risencrypto.github.io/CofactorClearing - (28.03.2025 23:10).

[5] SafeCurve, choosing safe curves for elliptic-curve cryptography, 2013, https://safecurves.cr.yp.to/twist.html - (29.03.2025 21:02).