Elliptic-curve cryptography

Projective coordinates and Curve25519

Tanja Lange

Eindhoven University of Technology

2MMC10 – Cryptology

How to implement curve arithmetic in practice?

In $\mathbf{F}_p$ with large p divisions are a lot more expensive than multiplications.

How to implement curve arithmetic in practice?

In $\mathbf{F}_p$ with large p divisions are a lot more expensive than multiplications.

We use projective coordinates to delay inversions:
Use $(X_1 : Y_1 : Z_1)$ with $Z_1 \neq 0$ to represent $(x_1, y_1) = (X_1/Z_1, Y_1/Z_1)$,
i. e., $(X_1 : Y_1 : Z_1) = (\lambda X_1 : \lambda Y_1 : \lambda Z_1)$ for $\lambda \neq 0$.
Delay division till the end of scalar multiplication aP.

How to implement curve arithmetic in practice?

In $\mathbf{F}_p$ with large p divisions are a lot more expensive than multiplications.

We use projective coordinates to delay inversions:
Use $(X_1 : Y_1 : Z_1)$ with $Z_1 \neq 0$ to represent $(x_1, y_1) = (X_1/Z_1, Y_1/Z_1)$,
i. e., $(X_1 : Y_1 : Z_1) = (\lambda X_1 : \lambda Y_1 : \lambda Z_1)$ for $\lambda \neq 0$.
Delay division till the end of scalar multiplication aP.

Derive formulas stating with $(x_i, y_i) = (X_i/Z_i, Y_i/Z_i)$,
bring result onto same denominator Z_3.

How to implement curve arithmetic in practice?

In $\mathbf{F}_p$ with large p divisions are a lot more expensive than multiplications.

We use projective coordinates to delay inversions:
Use $(X_1 : Y_1 : Z_1)$ with $Z_1 \neq 0$ to represent $(x_1, y_1) = (X_1/Z_1, Y_1/Z_1)$,
i. e., $(X_1 : Y_1 : Z_1) = (\lambda X_1 : \lambda Y_1 : \lambda Z_1)$ for $\lambda \neq 0$.
Delay division till the end of scalar multiplication aP.

Derive formulas stating with $(x_i, y_i) = (X_i/Z_i, Y_i/Z_i)$,
bring result onto same denominator Z_3.

Feature: These formulas capture ∞ on Weierstrass curve as $(0 : 1 : 0)$.

How to implement curve arithmetic in practice?

In $\mathbf{F}_p$ with large p divisions are a lot more expensive than multiplications.

We use projective coordinates to delay inversions:
Use $(X_1 : Y_1 : Z_1)$ with $Z_1 \neq 0$ to represent $(x_1, y_1) = (X_1/Z_1, Y_1/Z_1)$,
i. e., $(X_1 : Y_1 : Z_1) = (\lambda X_1 : \lambda Y_1 : \lambda Z_1)$ for $\lambda \neq 0$.
Delay division till the end of scalar multiplication aP.

Derive formulas stating with $(x_i, y_i) = (X_i/Z_i, Y_i/Z_i)$,
bring result onto same denominator Z_3.

Feature: These formulas capture ∞ on Weierstrass curve as $(0 : 1 : 0)$.

Sometimes cheaper to keep separate denominators:
$(x_i, y_i) = (X_i/Z_i, Y_i/T_i)$ represented as $((X : Z), (Y : T))$.

How to implement curve arithmetic in practice?

In $\mathbf{F}_p$ with large p divisions are a lot more expensive than multiplications.

We use projective coordinates to delay inversions:
Use $(X_1 : Y_1 : Z_1)$ with $Z_1 \neq 0$ to represent $(x_1, y_1) = (X_1/Z_1, Y_1/Z_1)$,
i. e., $(X_1 : Y_1 : Z_1) = (\lambda X_1 : \lambda Y_1 : \lambda Z_1)$ for $\lambda \neq 0$.
Delay division till the end of scalar multiplication aP.

Derive formulas stating with $(x_i, y_i) = (X_i/Z_i, Y_i/Z_i)$,
bring result onto same denominator Z_3.

Feature: These formulas capture ∞ on Weierstrass curve as $(0 : 1 : 0)$.

Sometimes cheaper to keep separate denominators:
$(x_i, y_i) = (X_i/Z_i, Y_i/T_i)$ represented as $((X : Z), (Y : T))$.

This is also the best way to see points at infinity on Edwards curves

$$((1 : 0), (\pm\sqrt{d} : \sqrt{a})) \text{ and } ((1 : \pm\sqrt{d}), (1 : 0))$$

if these exist.

Projective coordinates for Edwards curves

Taking inputs $P_1 = (X_1 : Y_1 : Z_1), P_2 = (X_2 : Y_2 : Z_2)$,
producing $P_1 + P_2 = P_3 = (X_3 : Y_3 : Z_3)$.

Optimized formulas:

$$\begin{aligned}
A &= Z_1 \cdot Z_2;\ B = A^2;\ C = X_1 \cdot X_2;\ D = Y_1 \cdot Y_2;\\
E &= d \cdot C \cdot D;\ F = B - E;\ G = B + E;\\
X_3 &= A \cdot F \cdot ((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D);\\
Y_3 &= A \cdot G \cdot (D - C);\\
Z_3 &= F \cdot G.
\end{aligned}$$

Needs $10\mathrm{M} + 1\mathrm{S} + 1\mathrm{d}\text{-mult} + 7\mathrm{add}$.

Projective coordinates for Edwards curves

Taking inputs $P_1 = (X_1 : Y_1 : Z_1), P_2 = (X_2 : Y_2 : Z_2)$,
producing $P_1 + P_2 = P_3 = (X_3 : Y_3 : Z_3)$.

Optimized formulas:

$$\begin{aligned}
A &= Z_1 \cdot Z_2;\ B = A^2;\ C = X_1 \cdot X_2;\ D = Y_1 \cdot Y_2; \\
E &= d \cdot C \cdot D;\ F = B - E;\ G = B + E; \\
X_3 &= A \cdot F \cdot ((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D); \\
Y_3 &= A \cdot G \cdot (D - C); \\
Z_3 &= F \cdot G.
\end{aligned}$$

Needs $10\mathbf{M} + 1\mathbf{S} + 1\mathbf{d}$-mult $+ 7$add.

See the EFD for many more formulas and the whole zoo of curve shapes.

As designer choose curves with small constants (under the condition that the system is secure – we will see what that means soon).

Example: Curve25519 (Bernstein 2006)

Let $p = 2^{255} - 19, A = 486662, B = 1.$

$$v^2 = u^3 + 486662u^2 + u$$

Is standardized for DH computations for the Internet in RFC 7748

$(A + 2)/4 = 121666$ is smallest with all properties from `http://safecurves.cr.yp.to/`.

Example: Curve25519 (Bernstein 2006)

Let $p = 2^{255} - 19, A = 486662, B = 1.$

$$v^2 = u^3 + 486662u^2 + u$$

Is standardized for DH computations for the Internet in RFC 7748

$(A+2)/4 = 121666$ is smallest with all properties from
http://safecurves.cr.yp.to/.

This curve is birationally equivalent to Edwards curve

$$x^2 + y^2 = 1 + dx^2y^2 \text{ for } d = 121665/121666.$$

Example: Curve25519 (Bernstein 2006)

Let $p = 2^{255} - 19, A = 486662, B = 1.$

$$v^2 = u^3 + 486662u^2 + u$$

Is standardized for DH computations for the Internet in RFC 7748

$(A+2)/4 = 121666$ is smallest with all properties from
http://safecurves.cr.yp.to/.

This curve is birationally equivalent to Edwards curve

$$x^2 + y^2 = 1 + dx^2y^2 \text{ for } d = 121665/121666.$$

Note that the map given on the board maps to $a'x'^2 + y'^2 = 1 + d'x'^2y'^2$
with $a' = 486664, d' = 486660.$

Example: Curve25519 (Bernstein 2006)

Let $p = 2^{255} - 19, A = 486662, B = 1.$

$$v^2 = u^3 + 486662u^2 + u$$

Is standardized for DH computations for the Internet in RFC 7748

$(A+2)/4 = 121666$ is smallest with all properties from `http://safecurves.cr.yp.to/`.

This curve is birationally equivalent to Edwards curve

$$x^2 + y^2 = 1 + dx^2y^2 \text{ for } d = 121665/121666.$$

Note that the map given on the board maps to $a'x'^2 + y'^2 = 1 + d'x'^2y'^2$ with $a' = 486664, d' = 486660.$
Note $a' = b^2$ in F_p and change $x = bx'$, $y = y'$.

Example: Curve25519 (Bernstein 2006)

Let $p = 2^{255} - 19, A = 486662, B = 1.$

$$v^2 = u^3 + 486662u^2 + u$$

Is standardized for DH computations for the Internet in RFC 7748

$(A + 2)/4 = 121666$ is smallest with all properties from `http://safecurves.cr.yp.to/`.

This curve is birationally equivalent to Edwards curve

$$x^2 + y^2 = 1 + dx^2y^2 \text{ for } d = 121665/121666.$$

Note that the map given on the board maps to $a'x'^2 + y'^2 = 1 + d'x'^2y'^2$ with $a' = 486664, d' = 486660.$
Note $a' = b^2$ in F_p and change $x = bx'$, $y = y'$.
This maps to $x^2 + y^2 = 1 + dx^2y^2$ with $d = d'/a' = 121665/121666.$