A Proof of the Group Properties of an Elliptic Curve

David M. Russinoff

ACL2 Workshop 2017

May 22, 2017

Curve25519

Let $\wp = 2^{255} - 19$, $A = 486662$, and

$$E = \{(x, y) \in \mathbb{F}_\wp \times \mathbb{F}_\wp \mid y^2 = x^3 + Ax^2 + x\} \cup \{\infty\}.$$

Our goal is to show that E is an abelian group under the following operation:

(1) $P \oplus \infty = \infty \oplus P = P$.

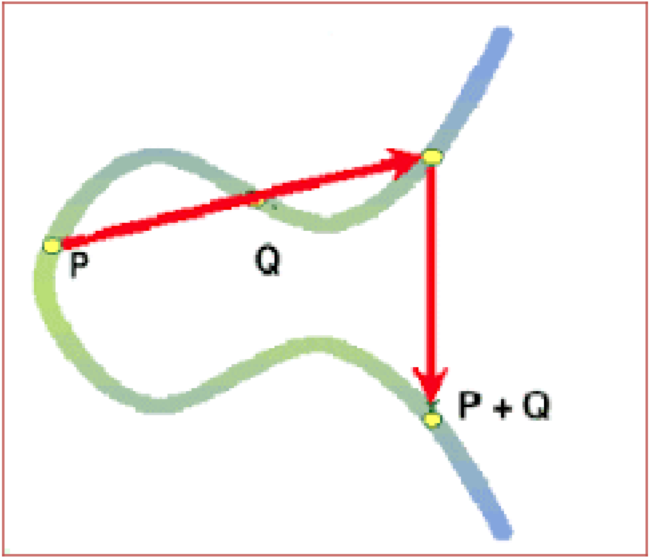
(2) If $P = (x, y)$, then $P \oplus (x, -y) = \infty$.

(3) If $P = (x_1, y_1)$, $Q = (x_2, y_2) \neq (x_1, -y_1)$, and

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2 \\ \frac{3x_1^2 + 2Ax_1 + 1}{2y_1} & \text{if } x_1 = x_2, \end{cases}$$

then $P \oplus Q = (x, y)$, where $x = \lambda^2 - A - x_1 - x_2$ and $y = \lambda(x_1 - x) - y_1$.

Elliptic Curve Addition



Curve25519

Let $\wp = 2^{255} - 19$, $A = 486662$, and

$$E = \{(x, y) \in \mathbb{F}_\wp \times \mathbb{F}_\wp \mid y^2 = x^3 + Ax^2 + x\} \cup \{\infty\}.$$

Our goal is to show that E is an abelian group under the following operation:

(1) $P \oplus \infty = \infty \oplus P = P$.

(2) If $P = (x, y)$, then $P \oplus (x, -y) = \infty$.

(3) If $P = (x_1, y_1)$, $Q = (x_2, y_2) \neq (x_1, -y_1)$, and

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2 \\ \frac{3x_1^2 + 2Ax_1 + 1}{2y_1} & \text{if } x_1 = x_2, \end{cases}$$

then $P \oplus Q = (x, y)$, where $x = \lambda^2 - A - x_1 - x_2$ and $y = \lambda(x_1 - x) - y_1$.

How hard could it be?

In principle, associativity could be verified by equating two compositions of the defining functions (for each of several cases), cross-multiplying, expanding into monomials, applying the curve equation, and canceling terms.

How hard could it be?

In principle, associativity could be verified by equating two compositions of the defining functions (for each of several cases), cross-multiplying, expanding into monomials, applying the curve equation, and canceling terms.

"Standard (although lengthy) calculations show that E is a commutative group under $\infty, -, +$."
– D. J. Bernstein, *Curve25519: new Diffie-Hellman speed records*

How hard could it be?

In principle, associativity could be verified by equating two compositions of the defining functions (for each of several cases), cross-multiplying, expanding into monomials, applying the curve equation, and canceling terms.

"Standard (although lengthy) calculations show that E is a commutative group under ∞, $-$, $+$."

– D. J. Bernstein, *Curve25519: new Diffie-Hellman speed records*

"Of course, there are a lot of cases to consider . . . . But in a few days you will be able to check associativity using these formulas. So we need say nothing more about the proof of the associative law!"

– J.H. Silverman and J.T. Tate, *Rational Points on Elliptic Curves*

How hard could it be?

In principle, associativity could be verified by equating two compositions of the defining functions (for each of several cases), cross-multiplying, expanding into monomials, applying the curve equation, and canceling terms.

"Standard (although lengthy) calculations show that E is a commutative group under $\infty, -, +$."
– D. J. Bernstein, *Curve25519: new Diffie-Hellman speed records*

"Of course, there are a lot of cases to consider .... But in a few days you will be able to check associativity using these formulas. So we need say nothing more about the proof of the associative law!"
– J.H. Silverman and J.T. Tate, *Rational Points on Elliptic Curves*

But the number of terms produced would exceed 10^{25}.

A Criterion of Proof

A proof may be said to be *computationally surveyable* if its only departure from strict surveyability is its dependence on unproved assertions that satisfy the following:

(1) Each such assertion pertains to a function for which a clear constructive definition has been provided, and merely specifies the value of that function corresponding to a concrete set of arguments.

(2) The computation of this value has been performed mechanically by the author of the proof in a reasonably short time.

(3) A competent reader could readily code the function in the programming language of his choice and verify the asserted result on his own computing platform.

Managing Computational Complexity

We combine three techniques:

- Sparse Horner Normal Form: an efficient method of establishing equality of multivariable polynomials
- Efficient reduction of SHNFs modulo the curve equation
- Encoding points on the curve as integer triples

Polynomial Terms

Standard encoding of polynomial terms as S-expressions:

Let

$$V = \texttt{(X Y Z)}.$$

If

$$\tau = \texttt{(* X (EXPT (+ Y Z) 3))} \in \mathcal{T}(V)$$

and

$$A = \texttt{((X . 2) (Y . 3) (Z . 0))},$$

then

$$evalp(\tau, A) = 2 \cdot (3 + 0)^3 = 54.$$

Sparse Horner Normal Form

A SHNF is an element of a certain set $\mathcal{H}$ of S-expressions.
We define two mappings:

- Given $V = (v_0 \ldots v_k)$ and $\tau \in \mathcal{T}(V)$, $norm(\tau, V) \in \mathcal{H}$.
- Given $N = (n_0 \ldots n_k)$ and $h \in \mathcal{H}$, $evalh(h, N) \in Z$.

Lemma Let $A = ((v_0 \;.\; n_0) \ldots (v_k \;.\; n_k))$.

$$evalh(norm(\tau, V), N) = evalp(\tau, A).$$

Corollary If $norm(\tau_1, V) = norm(\tau_2, V)$, then

$$evalp(\tau_1, A) = evalp(\tau_2, A).$$

SHNF Evaluation

A SHNF $h \in \mathcal{H}$ has one of three forms:

(1) $h \in \mathbb{Z}$:

$$evalh(h, N) = h.$$

(2) $h =$ `(POW i p q)`, where $i \in \mathbb{Z}^+$, $p \in \mathcal{H}$, and $q \in \mathcal{H}$:

$$evalh(h, N) = car(N)^i \cdot evalh(p, N) + evalh(q, cdr(N)).$$

(3) $h =$ `(POP i p)`, where $i \in \mathbb{Z}^+$, $p \in \mathcal{H}$:

$$evalh(h, N) = evalh(q, nthcdr(i, N)).$$

Normalization (Example)

Let $V = \texttt{(}x\ \ y\ \ z\texttt{)}$ and

$$\tau = 4x^4y^2 + 3x^3 + 2z^4 + 5 = x^3(4xy^2 + 3) + (2z^4 + 5).$$

Then

$$norm(\tau, V) = \texttt{(POW 3 } p\ \ q\texttt{)},$$

where

$$\begin{aligned} p &= norm(4xy^2 + 3, V) \\ &= \texttt{(POW 1 } norm(4y^2, V)\ \ norm(3, cdr(V))\texttt{)} \\ &= \texttt{(POW 1 (POP 1 (POW 2 4 0)) 3)}, \end{aligned}$$

$$q = norm(2z^4 + 5, cdr(V)) = \texttt{(POP 1 (POW 4 2 5))}.$$

Reduction Modulo the Curve Equation

Let $P_i = (x_i, y_i)$, $i = 0, 1, 2$, be fixed points on E.

$N =$ `(`y_0 y_1 y_2 x_0 x_1 x_2`)`, $V =$ `(Y0 Y1 Y2 X0 X1 X2)`,
$A =$ `((Y0 .` y_0`) (Y1 .` y_1`) (Y2 .` y_2`) (X0 .` x_0`) (X1 .` x_0`) (X2 .` x_2`))`.

We define a mapping

$$reduce : \mathcal{T}(V) \to \mathcal{H}$$

that effectively substitutes $x_i^3 + Ax_i^2 + x_i$ for y_i^2 wherever possible.

Lemma $evalh(reduce(\tau), N) \equiv evalh(norm(\tau), N) \pmod{\wp}$.

Corollary If $reduce(\sigma) = reduce(\tau)$, then

$$evalp(\sigma, A) \equiv evalp(\tau, A) \pmod{\wp}.$$

Encoding Points of E as Integer Triples

A point $P \in E$ is represented by $\mathcal{P} = (m, n, z) \in Z^3$ if

$$decode(\mathcal{P}) = \left(\frac{\bar{m}}{\bar{z}^2}, \frac{\bar{n}}{\bar{z}^3} \right) = P.$$

Note that every $P = (z, y) \in E$ admits the *canonical* representation $\mathcal{P} = (x, y, 1)$.
For two important cases, we define an efficiently computable operation "$\oplus$" on Z^3, involving no division in $\mathbb{F}_{\wp}$, such that if

$$decode(\mathcal{P}) = P \in E \text{ and } decode(\mathcal{Q}) = Q \in E,$$

then

$$decode(\mathcal{P} \oplus \mathcal{Q}) = P \oplus Q.$$

Case 1: $\mathcal{P} = (x, y, 1)$ and $P \neq Q$
Case 2: $\mathcal{P} = \mathcal{Q}$

CASE 1

If $\mathcal{P} = (x, y, 1)$ and $\mathcal{Q} = (m, n, z)$, define $\mathcal{P} \oplus \mathcal{Q} = (m', n', z')$, where

$$\begin{aligned} z' &= z(z^2 x - m), \\ m' &= \left(z^3 y - n\right)^2 - \left(z^2(A + x) + m\right)\left(z^2 x - m\right)^2 \\ n' &= \left(z^3 y - n\right)\left(z'^2 x - m'\right) - z'^3 y. \end{aligned}$$

Lemma If $decode(\mathcal{P}) = P \in E$, $decode(\mathcal{Q}) = Q \in E$, and $P \neq \pm Q$, then

$$decode(\mathcal{P} \oplus \mathcal{Q}) = P \oplus Q.$$

Case 2

If $\mathcal{P} = (m, n, z) \in Z^3$, define $\mathcal{P} \oplus \mathcal{P} = (m', n', z')$, where

$$\begin{aligned} z' &= 2nz, \\ w' &= 3m^2 + 2Amz^2 + z^4, \\ m' &= w'^2 - 4n^2(Az^2 + 2m), \\ n' &= w'(4mn^2 - m') - 8n^4. \end{aligned}$$

Lemma If $decode(\mathcal{P}) = P \in E$, then

$$decode(\mathcal{P} \oplus \mathcal{P}) = P \oplus P.$$

ENCODING POINTS ON THE CURVE AS TERM TRIPLES

Notation:

- $\mathcal{T} = \mathcal{T}(V)$.
- If $\tau \in \mathcal{T}$, then $\hat{\tau} = \mathit{evalp}(\tau, A)$.
- If $\Pi = (\mu, \nu, \zeta) \in \mathcal{T}^3$, then $\widehat{\Pi} = (\hat{\mu}, \hat{\nu}, \hat{\zeta})$ and $\mathit{decode}(\Pi) = \mathit{decode}(\widehat{\Pi})$.
- $\Pi_0 = (\texttt{X0}, \texttt{Y0}, \texttt{1})$, $\Pi_1 = (\texttt{X1}, \texttt{Y1}, \texttt{1})$, $\Pi_2 = (\texttt{X2}, \texttt{Y2}, \texttt{1})$.

Note that for $i = 0, 1, 2$,

$$\mathit{decode}(\Pi_i) = \mathit{decode}(\widehat{\Pi}_i) = \mathit{decode}(x_i, y_i, 1) = P_i.$$

The operation "$\oplus$" that we defined on $\mathbb{Z}^3$ may be lifted to $\mathcal{T}^3$ in a straightforward manner.

Case 1

If $\Pi = (\theta, \phi, 1) \in \mathcal{T}^3$ and $\Lambda = (\mu, \nu, \zeta) \in \mathcal{T}^3$, then we define $\Pi \oplus \Lambda = (\mu', \nu', \zeta')$, where

$$\zeta' = \texttt{(* } \zeta \texttt{(- (* (EXPT } \zeta \texttt{ 2) } \theta \texttt{) } \mu \texttt{),}$$

$$\begin{aligned} \mu' = \texttt{(- (EXPT (- (* (EXPT } \zeta \texttt{ 3) } \nu \texttt{) 2)} \\ \texttt{(* (+ (* (EXPT } \zeta \texttt{ 2) (+ } A\ \theta \texttt{)) } \mu \texttt{)} \\ \texttt{(EXPT (- (* (EXPT } \zeta \texttt{ 2) } \theta \texttt{) } \mu \texttt{) 2))),} \end{aligned}$$

$$\begin{aligned} nu' = \texttt{(- (* (- (* (EXPT} \zeta \texttt{ 3) } \phi \texttt{) } \nu \texttt{)} \\ \texttt{(- (* (EXPT } \zeta' \texttt{ 2) } \theta \texttt{) } \mu' \texttt{))} \\ \texttt{(* (EXPT } \zeta \texttt{ 3) } \phi \texttt{)).} \end{aligned}$$

Lemma If $decode(\Pi) = P \in E$, $decode(\Lambda) = Q \in E$, and $P \neq \pm Q$, then

$$decode(\Pi \oplus \Lambda) = P \oplus Q.$$

Case 2

Similarly, given $\Pi = (\mu, \nu, \zeta) \in \mathcal{T}^3$, we define $\Pi \oplus \Pi$ so that the following holds:

Lemma If $decode(\Pi) = P \in E$, then

$$decode(\Pi \oplus \Pi) = P \oplus P.$$

An Equivalence Relation on $\mathcal{T}^3$

Given $\Pi = (\mu, \nu, \zeta) \in \mathcal{T}^3$ and $\Pi' = (\mu', \nu', \zeta') \in \mathcal{T}^3$, let

$$\sigma = \texttt{(* } \mu \texttt{ (EXPT } \zeta' \texttt{ 2))}, \quad \sigma' = \texttt{(* } \mu' \texttt{ (EXPT } \zeta \texttt{ 2))},$$
$$\tau = \texttt{(* } \nu \texttt{ (EXPT } \zeta' \texttt{ 3))}, \quad \tau' = \texttt{(* } \nu \texttt{ (EXPT } \zeta \texttt{ 3))}.$$

If $\mathit{reduce}(\sigma) = \mathit{reduce}(\sigma')$ and $\mathit{reduce}(\tau) = \mathit{reduce}(\tau')$, then we shall write $\Pi \sim \Pi'$.

A consequence of our main result pertaining to *reduce*:

Lemma If $\mathit{decode}(\Pi) = P \in E$, $\mathit{decode}(\Pi') = P' \in E$, and $\Pi \sim \Pi'$, then $P = P'$.

Commutativity

We need only show that $P_0 \oplus P_1 = P_1 \oplus P_0$; commutativity follows by functional instantiation. We may assume $P_0 \neq \pm P_1$. By direct computation,

$$\Pi_0 \oplus \Pi_1 \sim \Pi_1 \oplus \Pi_0.$$

It follows that

$$decode(\Pi_0 \oplus \Pi_1) = decode(\Pi_1 \oplus \Pi_0),$$

where

$$decode(\Pi_0 \oplus \Pi_1) = decode(\Pi_0) \oplus decode(\Pi_1) = P_0 \oplus P_1$$

and

$$decode(\Pi_1 \oplus \Pi_0) = decode(\Pi_1) \oplus decode(\Pi_0) = P_1 \oplus P_0.$$

Associativity

The proof of associativity is similar in principle, but requires extensive case analysis.
By direct computation,

$$(\Pi_0 \oplus \Pi_1) \oplus \Pi_2 \sim \Pi_0 \oplus (\Pi_1 \oplus \Pi_2)$$

and therefore

$$\mathit{decode}((\Pi_0 \oplus \Pi_1) \oplus \Pi_2) = \mathit{decode}(\Pi_0 \oplus (\Pi_1 \oplus \Pi_2)).$$

Associativity follows under the conditions $P_0 \neq \pm P_1$, $P_0 \oplus P_1 \neq \pm P_2$, $P_1 \neq \pm P_2$, and $P_1 \oplus P_2 \neq \pm P_0$.
Other cases require additional computations:

$$(\Pi_0 \oplus \Pi_0) \oplus \Pi_1 \sim \Pi_0 \oplus (\Pi_0 \oplus \Pi_1),$$

$$(\Pi_0 \oplus \Pi_1) \oplus (\Pi_0 \oplus \Pi_1) \sim \Pi_0 \oplus (\Pi_1 \oplus (\Pi_0 \oplus \Pi_1)),$$

etc.