A Formal Library for Elliptic Curves

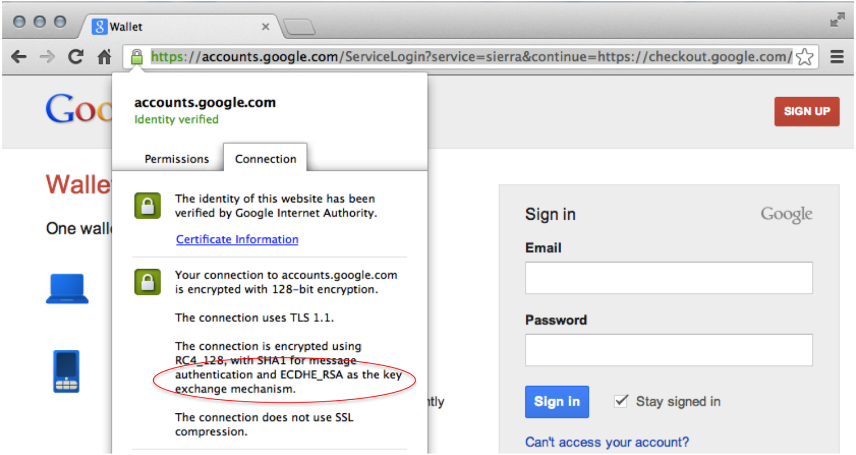
Iro Bartzia Pierre-Yves Strub

Inria Paris - Rocquencourt, France

IMDEA Software Institute, Spain

15/7/2014

Elliptic Curves and Cryptography

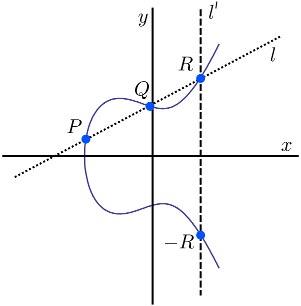


Definition

An elliptic curve E on a field K (characteristic $\neq 2, 3$) is defined

$$E = \{(x, y) \in K^2, y^2 = x^3 + Ax + B\} \cup \{P_\infty\}$$

where P_∞ is the point at infinity
and
$\Delta = 4A^3 + 27B^2 \neq 0$



Background

Elliptic curves are efficient groups for public-key cryptography:

- addition and scalar multiplication are computationally efficient
  $(m, P) \rightarrow [m]P = P \bigoplus P \bigoplus ... \bigoplus P$ (m-times)
- the Discrete Logarithm Problem is difficult relative to the size of the parameters
  $[m]P \rightarrow (m, P)$

Elliptic Curves and Cryptography

The realisation of cryptographic primitives requires the implementation of cryptographic algorithms.
This is a complicated task.

So how do we know that an implementation is correct ?
Use Formal Methods !

A formal theory for elliptic curves (outline)

A library for elliptic curves with SSReflect

- Elliptic curves of Weierstrass form
- Rational functions on elliptic curves
- Divisors

Previous formalizations in Coq

- Proving the group law for elliptic curves formally.
  L.Thery, G.Hanrot (2007)

The Picard theorem

The set of points of an elliptic curve is isomorphic to its Picard group of divisors.

- Consequence: An elliptic curve is a group.

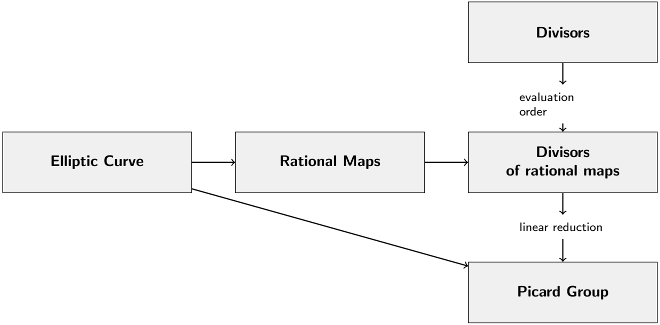


Figure 1: Roadmap to the proof

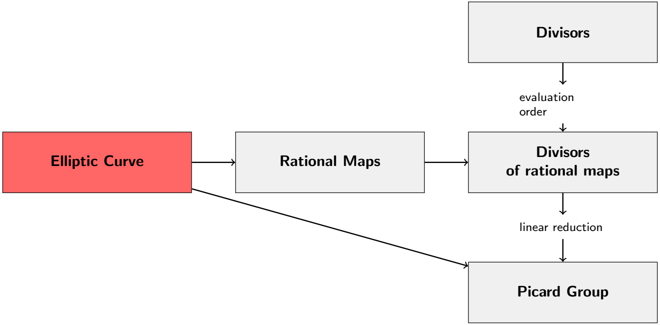


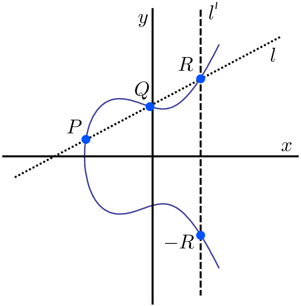
Figure 1: Roadmap to the proof

Definition

An elliptic curve E on a field K (characteristic $\neq 2, 3$) is defined

$$E = \{(x, y) \in K^2, y^2 = x^3 + Ax + B\} \cup \{P_\infty\}$$

where P_∞ is the point at infinity
and
$\Delta = 4A^3 + 27B^2 \neq 0$



Elliptic Curves of equation $y^2 = x^3 + Ax + B$

```
Record ecutype :=
  {A : K; B : K; _: 4*A^3 + 27*B^2 != 0 }

Inductive point := |EC_Inf |EC_In : K -> K -> point
Notation "(|x,y|)" := (EC_In x y)

Definition oncurve (p : point) :=
  match p with
  |EC_Inf => true
  |(|x,y|) => y^2 == x^3 + A*x + B
  end

Inductive ec :=
  |EC : forall p : point K, oncurve p -> ec
```

Rational maps

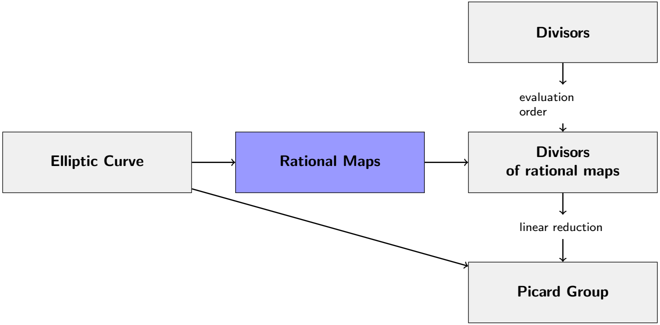


Figure 1: Roadmap to the proof

Rational functions on elliptic curves

The ring $K[E]$

The ring $K[E]$ of polynomials over the curve E is defined as the ring $K[x, y]$ quotiented by the ideal $\langle y^2 - (x^3 + ax + b) \rangle$.

For every class of $K[E]$ there exists a representative of the form $p(x) + yq(x)$.

example: $x^2(y^2 + y) = x^2(x^3 + ax + b) + yx^2$ in K[E]

```
Inductive ecring :=
 |ECRing : {poly K} * {poly K} -> ecring.

Notation "[ecp p1 *Y + p2]" := (ECRing (p1, p2)).
```

Rational functions on elliptic curves

The function field K(E)

The field $K(E)$ is the field of fractions of $K[E]$.

$f \in K(E)$ is of the form $f = \frac{n_1(x)+yn_2(x)}{d_1(x)+yd_2(x)}$

To formalize $K(E)$ we use the type

```
{fraction ecring}
```

Order

- A function $u \in K(E)$ is called a uniformizer at $P \in E$ if
  - $u(P) = 0$ and
  - every non-zero function $f \in K(E)$ can be written in the form $f = u^v g$ with $g(P) \neq 0, \infty$.
  - There exists a uniformizer for every point on the curve.
- The exponent v is independent from the choice of the uniformizer and is called the order of f at P.

- $v > 0 \Leftrightarrow f(P) = 0$ (P is a zero of f)
- $v < 0 \Leftrightarrow f(P) = \infty$ (P is a pole of f)

Order and evaluation of rational functions

- There exists a uniformizer for every point on the curve.
- For a rational function f,
  the sum of orders on all points of the curve is equal to zero.

Formally, we explicitly give the uniformizers for all points and a way to calculate the order.

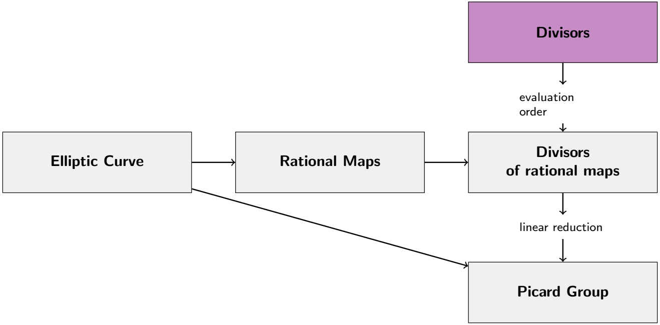


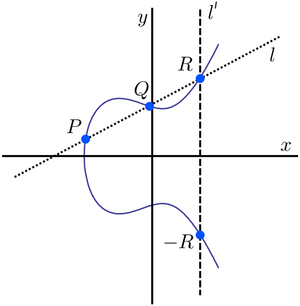
Figure 1: Roadmap to the proof

Divisors

A divisor on an elliptic curve E is a formal sum of points

$$D = \sum_{P \in E} n_P (P),$$

where $n_P \in \mathbb{Z}$, only finitely many nonzero.
The degree of D is the sum of the coefficients n_P for all $P \in E$.



example of a divisor $D = 3(P) - 7(R)$

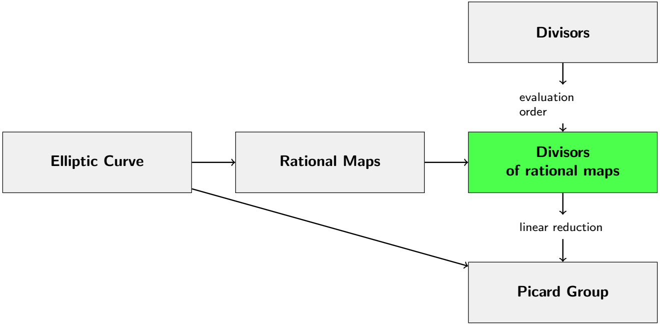
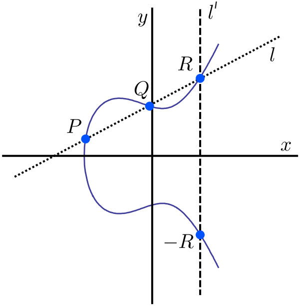


Figure 1: Roadmap to the proof

Principal Divisors - Princ(E)

Given $f \in K(E)$, $f \neq 0$, the principal divisor $Div(f)$ of f is defined as the formal (finite) sum:

$$Div(f) = \sum_{P \in E}(ord_f(P))(P).$$



$Div(l) = (P) + (Q) + (R) - 3(P_\infty)$
$Div(l') = (R) + (-R) - 2(P_\infty)$

Principal divisors

Given $f \in K(E)$, $f \neq 0$, the principal divisor $Div(f)$ of f is defined as the formal (finite) sum:

$$Div(f) = \sum_{P \in E} (ord_f(P))(P).$$

```
Definition ecdivp (f : ecring) : {freeg (point)} :=
\sum_((|x, y|) <- ecroots f)
    << (order f (|x,y|)) * (|x,y|) >>
  + << (order f EC_Inf) * EC_Inf >>.
```

To formalize divisors of $f \in K(E)$ we use the fraction construction of ssr.

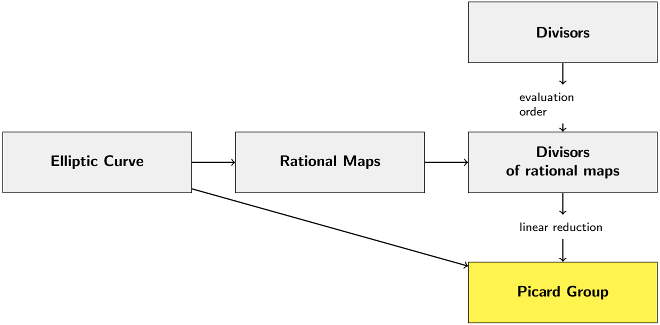


Figure 1: Roadmap to the proof

The Picard Group

$Princ(E) \subset Div^0 \subset Div$

$Pic^0(E)$ is defined as the quotient of zero-degree divisors by the equivalence relation $\sim$

$$D_1 \sim D_2 \text{ if and only if } \exists f \in K(E) \text{ such that } div(f) = D_1 - D_2.$$

For every class of $Pic^0(E)$ there exists a unique representative of the form $(P) - (P_\infty)$.

Linear Reduction

Linear Reduction is used to find the representative of the form $(P) - (P_\infty)$.
Lemma: $(P) + (Q) \sim (P \oplus Q) + (P_\infty)$

Linear Reduction

$$\begin{array}{rl} D = & (P_0) + ... + (P_n) - (Q_0) - ...(Q_n) \\ \sim & (P_0 \oplus ... \oplus P_n) + n(P_\infty) - (Q_0 \oplus ... \oplus Q_n) - n(P_\infty) \\ = & (P) - (Q) \\ \sim & (P \ominus Q) - (P_\infty). \end{array}$$

Linear reduction

$$D = \quad (P_0) + \ldots + (P_n) - (Q_0) - \ldots (Q_n)$$

```
Definition fgpos (D : {freeg K}) :=
\sum_(p <- dom D | coeff p D > 0) << |coeff p D| * p >>.

Definition fgneg (D : {freeg K}) :=
\sum_(p <- dom D | coeff p D < 0) << |coeff p D| * p >>.
```

Linear reduction

$$
\begin{aligned}
D = \quad & (P_0) + \ldots + (P_n) - (Q_0) - \ldots (Q_n) \\
\sim \quad & (P_0 \oplus \ldots \oplus P_n) + n(P_\infty) - (Q_0 \oplus \ldots \oplus Q_n) - n(P_\infty) \\
= \quad & (P) - (Q) \\
\sim \quad & (P \ominus Q) - (P_\infty).
\end{aligned}
$$

```
Definition lr_r (D : {freeg point}) :=
let iter p n := iterop _ n + p EC_Inf in
\sum_(p <- dom D | p != EC_Inf) (iter p |coeff p D|).

  Definition lr (D : {freeg point}) : point :=
let: (Dp, Dn) := (fgpos D, fgneg D) in
lr_r Dp - lr_r Dn.
```

Linear reduction

$$
\begin{array}{rl}
D = & (P_0) + ... + (P_n) - (Q_0) - ...(Q_n) \\
\sim & (P_0 \oplus ... \oplus P_n) + n(P_\infty) - (Q_0 \oplus ... \oplus Q_n) - n(P_\infty) \\
= & (P) - (Q) \\
\sim & (P \ominus Q) - (P_\infty).
\end{array}
$$

```
Lemma ecdeqv_lr D: all oncurve (dom D) ->
D :~: << lr D >> + << (deg D - 1) * EC_Inf >>.
```

Linear reduction

$$
\begin{aligned}
D = \quad & (P_0) + \ldots + (P_n) - (Q_0) - \ldots(Q_n) \\
\sim \quad & (P_0 \oplus \ldots \oplus P_n) + n(P_\infty) - (Q_0 \oplus \ldots \oplus Q_n) - n(P_\infty) \\
= \quad & (P) - (Q) \\
\sim \quad & (P \ominus Q) - (P_\infty).
\end{aligned}
$$

$$
\begin{aligned}
D = \quad & (P_0) + \ldots + (P_n) - (Q_0) - \ldots(Q_n) \\
\sim \quad & ((P_0 \oplus P_1) \oplus \ldots \oplus P_n) \ominus (Q_0 \oplus \ldots \oplus Q_n) - (P_\infty) \\
\sim \quad & (P_0 \oplus \ldots \oplus (P_{n-1} \oplus P_n)) \ominus (Q_0 \oplus \ldots \oplus Q_n) - (P_\infty)
\end{aligned}
$$

```
Lemma L_2_40: forall p q, << p >> :~: << q >> -> p = q.
```

The isomorphism

For every class of $Pic^0(E)$ there exists a unique representative of the form $(P) - (P_\infty)$.

The mapping

$$\begin{array}{rccc} \phi : & E & \rightarrow & Pic^0(E) \\ & P & \mapsto & [(P) - (P_\infty)] \end{array}$$

is an isomorphism.

To sum up

We have developed a formal library for

- Elliptic curves
- Rational functions on elliptic curves
- Evaluation theory for rational functions
- Divisors and principal divisors
- Formal proof of the Picard theorem
  10000 lines of code (3500 definitions, 6500 proof)
  available at http://strub.nu/research/ec/

Ongoing and future work

- Fast algorithms for scalar multiplication on ECs (GLV)
- Different coordinate systems
- Generalization to more general curves
- EC group structure theorem (needed for EasyCrypt)

Questions ?

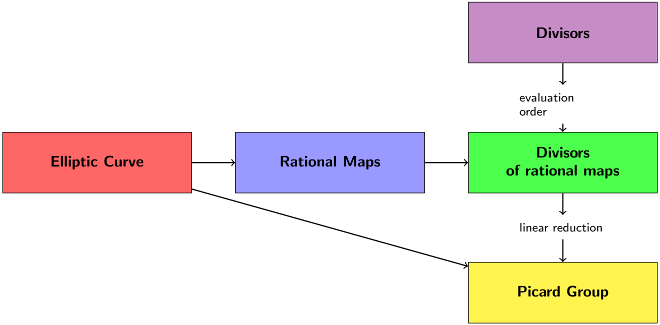


Figure 1: Roadmap to the proof

Thank you for your attention !

Order and evaluation of rational functions

Evaluation

- $f \in K(E)$ is regular at $P = (x_P, y_P)$ if there exists a representative $\frac{g}{h}$ of f such that $h(x_P, y_P) \neq 0$.
- If f is regular at P, then $f(P) = \frac{g(x_P, y_P)}{h(x_P, y_P)}$.
- If f is not regular at P, then P is called a pole of f and $f(P) = \infty$.

example: $E : y^2 = x^3 - x$ and $f(x, y) = \frac{x}{y}$
$f(0,0) = \frac{x}{y}(0,0) = \frac{xy}{y^2}(0,0) = \frac{xy}{x^3 - x}(0,0) = \frac{y}{x^2 - 1}(0,0) = 0$

The Galland-Lambert-Vanstone Algorithm

The GLV Algorithm (2000)

Let E elliptic curve over F_p .

- Let $\phi : E \to E$ an efficiently computable endomorphism of E such that $\forall P \in E$, $\phi(P) = \lambda \times P$ for some $\lambda \in \mathbb{Z}$.
- A decomposition algorithm D such that $\forall n \in \mathbb{Z}$, $D(n) = (n_1, n_2) \in \mathbb{Z} \times \mathbb{Z}$ such that $n = n_1 + \lambda n_2 [modN]$ where N is the order of P.
- A multiexponentiation algorithm that computes efficiently $a \times P + b \times Q$ $\forall a, b \in \mathbb{Z}$ and $\forall P, Q \in E$.

Then $n \times P = n_1 \times P + n_2 \times \phi(P)$.

Group structure over finite fields

Group structure of elliptic curves over F_q (Cassel)

Let E elliptic curve over F_q.

$E(F_q)$ is

- either cyclic or
- isomorphic to to a product of two cyclic groups $\mathbb{Z}\backslash L\mathbb{Z} \times \mathbb{Z}\backslash M\mathbb{Z}$ with $L|M$.

The definition of the operations requires a proof that the operations are internal.

```
Inductive elt : Set :=
|inf_elt : elt
|curve_elt x y (is_eq y^2 (x^3 + A*x + B) = true) : elt

Definition opp p :=
match p with
|inf_elt => inf_elt
|curve_elt x y H => curve_elt x (-y) opp_lem
end
```

Divisors over elliptic curves

A divisor on an elliptic curve E is a formal sum of points

$$D = \sum_{P \in E} n_P (P),$$

where $n_P \in \mathbb{Z}$, only finitely many nonzero.

```
Definition reduced (D : seq (int * T)) :=
(uniq [seq zx.2 | zx <- D]) && (all [pred zx | zx.1 !=
    0]).

Record prefreeg : Type := mkPrefreeg {
seq_of_prefreeg : seq (int * T) ;
_ : reduced seq_of_prefreeg}.
```

To define the type freeg T of divisors we quotient by the perm-eq equivalence relation.

The function ecroots

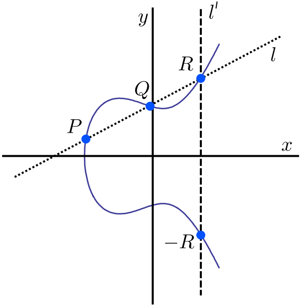
```
Definition ecdivp (f : ecring) : {freeg (point)} :=
 \sum_((|x, y|) <- ecroots f)
     << (order f (|x,y|)) * (|x,y|) >>
   + << (order f EC_Inf) * EC_Inf >>.
```

```
  Definition ecroots f : seq (K * K) :=
 let forx := fun x =>
   let sqrts := roots ('X^2 - ('X^3 + A *: 'X + B).[x])
       in
     [seq (x, y) | y <- sqrts & f.[x, y] == 0]
 in
   undup (flatten ([seq forx x | x <- roots (norm f)])).
```

Linear Reduction

Lemma

$(P) + (Q) \sim (P \oplus Q) + (P_\infty)$



$$Div(l) = (P) + (Q) + (R) - 3(P_\infty)$$
$$Div(l') = (R) + (-R) - 2(P_\infty)$$
$$Div(l/l') = Div(l) - Div(l') = (P) + (Q) - (P + Q) - (P_\infty)$$