Nov19

Wednesday, November 18, 2020 9:52 PM

Elliptic Curve Addition Algorithm over $\mathbb{F}_p$

$E: \; y^2 = x^3 + ax + b. \qquad \Delta_E \not\equiv 0 \mod p.$

Input: $P_1, P_2 \in E(\mathbb{F}_p)$

Output: $P_1 + P_2$.

→ ① If $P_1 = \mathcal{O}$ return $P_1 + P_2 = P_2$

→ ② If $P_2 = \mathcal{O}$ return $P_1 + P_2 = P_1$

→ ③ Else $P_i = (x_i, y_i)$ $i = 1, 2$.

④ If $y_1 = -y_2$ and $x_1 = x_2$ ← $P_1 = -P_2$

return $P_1 + P_2 = \mathcal{O}$.

$O(\log(p))$

⑤ Else

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq P_2 \\[2ex] \dfrac{3x_1^2 + a}{2y_1} & \text{if } P_1 = P_2 \end{cases}$$

$x_3 = \lambda^2 - x_1 - x_2$ & $y_3 = \lambda(x_1 - x_3) - y_1$

Return (x_3, y_3).

Theorem This works.

Example: $E: \; y^2 = x^3 + x + 2$ over $\mathbb{F}_3$

$E(\mathbb{F}_3) = \{\mathcal{O}, \underset{P}{(1,1)}, \underset{Q}{(1,2)}, \underset{R}{(2,0)}\}$ ←

$2 \cdot P$ Compute $(x_1, y_1) = (1,1)$

$$\lambda = \frac{3x_1^2 - a}{2 \cdot y_1} = (3+1) \cdot \left(\frac{1}{2}\right)$$

$$\equiv \frac{1}{2} = 2$$

$$x_3 = \lambda^2 - x_1 - x_2 = 2^2 - 1 - 1 = 2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$= 2(1-2) - 1 = -3 \equiv 0.$$

get $(x_3, y_3) = (2,0) = R$

$\underline{P + P = R}$

$2 \cdot R$ $(x_1, y_1) = (2,0) = (x_2, y_2) \leftarrow$

$x_1 = x_2$ $y_1 = -y_2$

$\Longrightarrow 2 \cdot R = \mathcal{O}$

$E(\mathbb{F}_p)$	$\mathcal{O}$	P	Q	R
$\mathcal{O}$				
P		R		Q
Q				P
R				$\mathcal{O}$

Exercise Fill this in.

$$\underbrace{P+P+P}_{\substack{\parallel \\ R+P}} = Q$$

Lagrange $\neq \mathcal{O}$

$$E(\mathbb{F}_p) = \langle P \rangle \simeq \mathbb{Z}/4\mathbb{Z}.$$

$$P \neq Q + P$$

The Elliptic Curve Discrete Log

Recall: Standard DLP (for $\mathbb{F}_p^*$)

Alice publishes $g, h \in \mathbb{F}_p^*$

Alice knows secret x s.t. $g^x \equiv h \bmod p$.

Hard to find x.

Remark Index Calculus find x subexponential time.

Translate this to $E(\mathbb{F}_p)$

Alice publishes $P, Q \in E(\mathbb{F}_p)$.

Alice's secret number is n

$$\text{s.t.} \quad Q = \underbrace{P + P + \cdots + P}_{n\text{- times}} = nP.$$

Def^n E an elliptic curve over $\mathbb{F}_p$. $P, Q \in E(\mathbb{F}_p)$ are points.

The Elliptic Curve Discrete Log Problem (ECDLP)

is the problem of finding n s.t.

$$Q = n\cdot P.$$

$(= Q \div P)$

We denote this integer by $n = \log_P Q$

Call n the elliptic discrete log of P

Remarks & Difficulties

① There may be points $P, Q \in E(\mathbb{F}_p)$ wher

$Q \neq nP$ for any n.

Ex $R \in E(\mathbb{F}_3)$ $R = (2,0)$.

$$2R = R+R = \mathcal{O}$$
$$3R = R+R+R = \mathcal{O}+R = R$$
$$4R = \underbrace{R+R+R}+R = R+R = \mathcal{O} \ \cdots$$

$$nR = \begin{cases} R & n \text{ odd} \\ \mathcal{O} & n \text{ even.} \end{cases}$$

$Q = (1,2)$ ←

$\log_R Q$ DNE.

$1\cdot R = 3\cdot R = 5\cdot R = \cdots$

$\log_R R = 1, 3, 5, 7, \ldots$

In practice not a huge issue. Bc alice publishes P, computes $Q = nP$. Publishes Q.

② If there exists an n w/ $Q = n\cdot P$. Then there exist ∞ly many such n.

Proof

$\mathcal{O} \neq P \in E(\mathbb{F}_p)$ in a finite group. So

$\exists$ $s > 0$ s.t. $sP = \mathcal{O}$.

Let $|P| = s > 0$

If $n_0 \cdot P = Q$ $\quad (n_0 + is)P = Q \quad \forall\, i \in \mathbb{Z}$.

Solution: $\log_p Q \in \mathbb{Z}/s\mathbb{Z}$ well defined.

Prop

$$\begin{tikzcd}
\log_p : \langle P \rangle \arrow[r] \arrow[d, phantom, "\parallel"] & \mathbb{Z}/s\mathbb{Z}. \\
\{0, P, 2P, \dots\} \arrow[d, phantom, "\cap"] & \\
E(\mathbb{F}_p) &
\end{tikzcd}$$

well defined homomorphism. i.e.

$$\log_p(Q_1 \underset{\uparrow}{+} Q_2) = \log_p Q_1 \underset{\uparrow}{+} \log_p Q_2 .$$

Rmk Book writes $\log_p : E(\mathbb{F}_p) \longrightarrow \mathbb{Z}/s\mathbb{Z}$

WRONG!!

proof $\log_p Q_1 = n_1$ $\qquad$ $\log_p Q_2 = n_2$

so $(n_1 + n_2)P = n_1 P + n_2 P = Q_1 + Q_2$

so $n_1 + n_2 \equiv \log_p(Q_1 + Q_2) \mod s.$

Example

Saw $P + P = R$

$P + P + P = Q$

$\log_p R = \underline{2}$ $\qquad$ $\log_p \mathcal{O} = \underline{0}$

$\log_p Q = \underline{3}$ $\qquad$ $\log_p P = \underline{1}.$

$$\rightsquigarrow \log_P : E(\mathbb{F}_3) \xrightarrow[\simeq]{} \mathbb{Z}/4\mathbb{Z}.$$

Seems difficult to recover n from P and $Q = nP$.

One approach

$\longrightarrow P, 2P, 3P, \ldots\ldots$ until you find Q.

$\rightsquigarrow$ If I choose $\#E(\mathbb{F}_p) >> 0$ I'm safe.

$$\#E(\mathbb{F}_p) = p+1 - \underbrace{t_p}_{\text{at most } 2\sqrt{p}} = O(p)$$

But....

I need to be able to compute $n \cdot P$ rather quickly.

If I do this

$P + P + \ldots\ldots$ n-times

too slow

Need a better way to compute the fn.

$$\left.\begin{aligned} \mathbb{Z} &\longrightarrow E(\mathbb{F}_p) \\ n &\longmapsto nP. \end{aligned}\right\} \text{fast to hide my } n.$$

Need a version of fast powering

Double and Add Algorithm

Recall Fast powering was square & multiply.

$\mathbb{F}_p^*$

$\uparrow$

× operation

$E(\mathbb{F}_p)$

$\uparrow$

+ our operation

Given E, a pt $P \in E(\mathbb{F}_p)$

Input: $n \in \mathbb{N}$. Output: $n \cdot P$.

Step 1: Compute n in binary. $\log_2(n) \approx r$

$$\underline{n = n_0} + n_1 \cdot 2 + \cdots + n_r \cdot 2^r \qquad n_i \in \{0, 1\}. \quad n_r = 1$$

Step 2: Double P $\underline{r \text{ times}}$. r doubles

$$Q_0 = P. \quad Q_1 = 2P. \quad Q_2 = 2Q_1. \quad \cdots \quad Q_r = 2 \cdot Q_{r-1}.$$

notice $\underline{Q_i = 2^i \cdot P.}$

Step 3

$n_i = 0, 1.$ $\underline{\le r \text{ additions}}$

$$Q = n_0 \cdot Q_0 + n_1 \cdot Q_1 + \cdots + n_r \cdot Q_r.$$

Return Q.

Proof

$$Q = n_0 \cdot 2^0 \cdot P + n_1 \cdot 2^1 \cdot P + \cdots + n_r \cdot 2^r \cdot P$$

$$= \underline{(n_0 \cdot 2^0 + n_1 \cdot 2^1 + \cdots + n_r \cdot 2^r)} \cdot P.$$

$$= n \cdot P$$

Time analysis

$r \approx \log_2(n)$

Time complexity of Ell Curve addition?

$$\text{Fixed \# operations} + \underbrace{\text{one } \mathbb{F}_p\text{-division}}_{O(\log(p))}.$$

$$n = O(|P|) = O(\underset{\overset{\uparrow}{p+1-t_p}}{\#E(\mathbb{F}_p)}) = O(p).$$

$r = O(\log(p))$.

1. Binary $O(\log(p)) \leftarrow$
2. r doublings $O(\log(p)\cdot\log(p)) = O(\log(p)^2)$
3. r addition $O(\log(p)^2) \leftarrow$

$O(\log(p)^2)$.

Example:

$n = 947 = 1 + 2 + 2^4 + 2^5 + 2^7 + 2^8 + 2^9$

⤳ Double 9 times → $Q_i = 2^i P$

⤳ Add 6 times $Q_0 + Q_1 + Q_4 + Q_5 + Q_7 + Q_8 + Q_9$.

15 point operations.

$947 = 1 + 2 - 2^4 - 2^6 + 2^{10} \leftarrow$

$$Q_0 + Q_1 - Q_4 - Q_6 + \underline{\underline{Q_{10}}} = nP$$

$\rightsquigarrow$ Double 10 times.

$\rightsquigarrow$ Add 4 times. $\leftarrow$

14 point operation

$\nwarrow$ 1 pt less

$Q_i = (x_i, y_i)$

$-Q_i = (x_i, -y_i)$

$\uparrow$ easy to compute!

Def$^{\underline{n}}$

A expansion $n = a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 + \cdots + a_r \cdot 2^r$ w/ $a_i = \{-1, 0, 1\}$ is called a Ternary expansion.

Comparison (after computing expansions)

$k = \lfloor \log_2 n \rfloor + 1$

Double and add	Ternary
k doubles	$k+1$ doubles
$\leq k$ additions	$\leq \frac{k}{2}$ adds * $\leftarrow$
$\leq 2k$ steps	$\leq \frac{3}{2}k + 1$ steps.
	$\boxed{75\%}$

(on average Ternary is $\leq \frac{1}{3}k$ adds) *

Rmks Not an improvement for $\mathbb{F}_p^*$ DLP

$$g_i \xmapsto{\text{logarithmic}} g_i^{-1}$$

But for ECDLP

$$Q_i \xrightarrow{O(1)} -Q_i$$

Prop $n \in \mathbb{N}$. $k = \lfloor \log_2 n \rfloor + 1$

$2^k > n \geq 2^{k-1}$ Then we can write $n = u_0 + u_1 \cdot 2 + u_2 \cdot 2^2 + \dots + u_k \cdot 2^k$ w/ $u_i := \{-1, 0, 1\}$ & at most $k/2$ of the u_i are nonzero.

Proof

Find binary expansion for n

log n $n = n_0 + n_1 \cdot 2 + \dots + n_{k-1} \cdot 2^{k-1}$ $n_i \in \{0, 1\}$

Start from left & look for 2 or more consecutive 1s.

$$n_s = n_{s+1} = \dots = n_{s+t-1} = 1$$

$$n_{s+t} = 0.$$

$$\{$$

$$\cdots \underset{\substack{\uparrow \\ -1}}{+} 2^s + \underbrace{2^{s+1} + \cdots + 2^{s+t-1}}_{0} + \underset{\substack{\uparrow \\ 1}}{0} \cdot 2^{s+t} + \cdots$$

$$= \cdots + 2^s(1 + 2 + \cdots + 2^{t-1}) + 0 \cdot 2^{s+t} + \cdots$$

$$= \cdots + 2^s(2^t - 1) + 0 \cdot 2^{s+t} + \cdots$$

$$= \cdots - 2^s + \underset{\uparrow}{2^{s+t}} + \cdots$$

Continue from 2^{s+t}.

By design, no consecutive nonzero terms

Avg log n.

Attacking the ECDLP

1) Use a collision algorithm.

$Q = nP$. Eve choose $j_1, \ldots, j_r, k_1, \ldots, k_r \in [1, p)$ at random

List 1: $j_1 \cdot P, j_2 \cdot P, \ldots, j_r \cdot P$

List 2: $k_1 \cdot P + Q, k_2 \cdot P + Q, \ldots, k_r \cdot P + Q$

} Computed w/ Double & add.

Collision!

$$j_u P = k_v P + Q$$

$$Q = (\underbrace{j_u - k_v}) P$$

$\hookrightarrow \log_P Q = j_u - k_v.$

Saw $r \approx 3\sqrt{p}$ then very good chance 2 random lists of length r in a set of size p collide.

2) Baby Steps Giant Steps

$r > \sqrt{|P|} \approx \sqrt{p}$ $< \sqrt{p+1+2\sqrt{p}}$

Defined for any group.

$L_1: O, P, 2P, \dots, rP$

$L_2: Q, Q-rP, Q-2rP, \dots, Q-r^2P.$

each step 1 addition on $E(\mathbb{F}_p)$

know ∃ collision

$$jP = Q - kP$$

$$\Longrightarrow (j+k)P = Q \qquad \log = j+k.$$

Both needed long lists... ☹

③ Adapt Pollard's ρ algorithm to find a collision.

HW

$$\dots R \longmapsto \left\{ \begin{matrix} PR \\ R^2 \\ RQ \end{matrix} \right\}$$

Rmk: For $\mathbb{F}_p^*$ DLP ∃ subexp alg.

* For ECDLP, best attack is $O(\sqrt{p})$

exponential on # bits.

Next Time: Elliptic Curve Cryptosystems

- Diffie Hellman
- Elgamal
- DSA

relied on DLP for $\mathbb{F}_p^*$. Can easily be adapted to any group G.