

# Elliptic Curve Discrete Logarithm Problem

Dr. F. Vercauteren

Katholieke Universiteit Leuven

13 May 2005



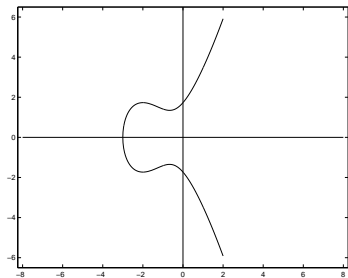
# Definition of Elliptic Curve

- ▶ Elliptic curve  $E$  over field  $\mathbb{K}$  is defined by

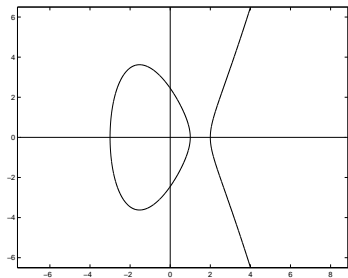
$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, a_i \in \mathbb{K}$$

- ▶ The set  $E(\mathbb{K})$  consists of all  $(x, y) \in \mathbb{K} \times \mathbb{K}$ , which satisfy this equation together with  $\mathcal{O}$
- ▶  $\mathcal{O}$  is called *point at infinity*
- ▶  $\exists$  addition law on  $E$  and the set  $E(\mathbb{K})$  is a group

# Elliptic Curves over $\mathbb{R}$

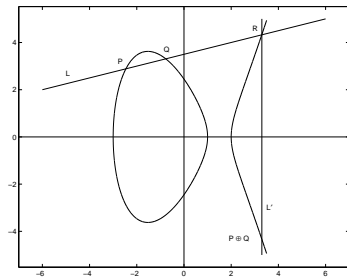


$$y^2 = x^3 + 4x^2 + 4x + 3$$

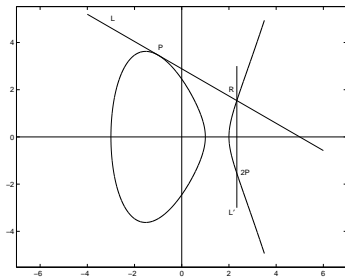


$$y^2 = x^3 - 7x + 6$$

# Addition Law on Elliptic Curve



Adding two points



Doubling a point

$$y^2 = x^3 - 7x + 6$$

## Exercise: Addition Law on Elliptic Curve

Let  $P_1 \oplus P_2 = P_3$ , with  $P_i = (x_i, y_i) \in E$

▶ If  $x_1 = x_2$  and  $y_1 + y_2 + a_1x_2 + a_3 = 0$ , then  $P_1 \oplus P_2 = \mathcal{O}$ ,

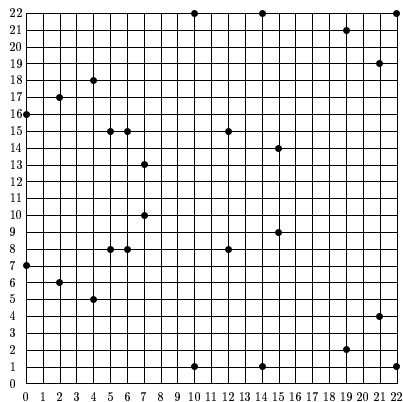
▶ Else

$$\begin{array}{l} x_1 \neq x_2 \\ x_1 = x_2 \end{array} \begin{cases} \lambda = (y_2 - y_1)/(x_2 - x_1) \\ \nu = (y_1x_2 - y_2x_1)/(x_2 - x_1) \\ \lambda = (3x_1^2 + 2a_2x_1 + a_4 - a_1y_1)/(2y_1 + a_1x_1 + a_3) \\ \nu = (-x_1^3 + a_4x_1 + 2a_6 - a_3y_1)/(2y_1 + a_1x_1 + a_3) \end{cases}$$

The point  $P_3 = P_1 \oplus P_2$  is given by

$$\begin{aligned} x_3 &= \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \\ y_3 &= -(\lambda + a_1)x_3 - \nu - a_3 \end{aligned}$$

# Elliptic Curves over Finite Fields



The elliptic curve  $y^2 = x^3 + x + 3 \pmod{23}$

# Elliptic Curve Discrete Logarithm Problem

Let  $E$  be elliptic curve over finite field  $\mathbb{F}_q$

- ▶ Take point  $P \in E(\mathbb{F}_q) \setminus \{\mathcal{O}\}$  and compute  $Q = \mathbf{d} \cdot P$
- ▶ ECDLP: given  $P$  and  $Q$ , compute  $\mathbf{d}$

Analogy with finite fields: Let  $\mathbb{F}_p$  be finite field

- ▶ Take element  $g \in \mathbb{F}_p^\times$  and compute  $h = g^{\mathbf{d}}$
- ▶ DLP: given  $g$  and  $h$ , compute  $\mathbf{d}$

$\Rightarrow$  Translate primitives based on DLP to ECDLP setting

# Pohlig and Hellman Reduction

- ▶ Let  $G$  be a group of order  $n = \prod_{i=1}^k p_i^{e_i}$  and let  $Q = \mathbf{d}P$
- ▶ Assume prime  $p|n$  and let  $t = n/p$ , then

$$Q' = tQ = t\mathbf{d}P = \mathbf{d}(tP) = \mathbf{d}P' = \mathbf{d}_0P'$$

- ▶ Note  $P', Q'$  are in subgroup of order  $p$ , thus  $\mathbf{d}_0 \equiv \mathbf{d} \pmod{p}$
- ▶ Find  $\mathbf{d} \pmod{p}$  and recover  $\mathbf{d} \pmod{p_1 \cdots p_k}$  using CRT
- ▶ Exercise: algorithm to compute  $\mathbf{d} \pmod{p^e}$  with  $e > 1$
- ▶ Hint: write  $\mathbf{d} \pmod{p^e} = \mathbf{d}_0 + \mathbf{d}_1p + \cdots + \mathbf{d}_{e-1}p^{e-1}$
- ▶ Conclusion: order of  $G$  should contain large prime factor

# Pollard Rho Algorithm: General Case

- ▶ Let  $S$  be a set  $|S| = n$  and  $f : S \rightarrow S$  random mapping
- ▶ Starting with random value  $x_0 \in S$ , compute

$$x_{i+1} = f(x_i) \text{ for } i \geq 0$$

- ▶ Sequence  $x_0, x_1, x_2, \dots$  is deterministic *random walk*
- ▶ Since  $S$  is finite we must eventually obtain

$$x_i = x_j \Rightarrow x_{i+1} = f(x_i) = f(x_j) = x_{j+1}$$

- ▶ Sequence  $x_0, x_1, x_2, \dots$ , becomes cyclic
- ▶ Picture of sequence looks like the Greek letter rho  $\rho$

# Pollard Rho Algorithm: General Case

- ▶ The rho-shape has an initial tail and cyclic part
- ▶ Expected length of both tail and cyclic part is

$$\sqrt{\pi n/8}$$

- ▶ Using this in naive way also requires  $O(\sqrt{n})$  memory
- ▶ Main problem in Baby-Step-Giant-Step algorithm
- ▶ Use Floyd's cycle finding algorithm

# Floyd's Cycle Finding Algorithm

- ▶ Given  $(x_1, x_2)$ , compute  $(x_2, x_4)$ , then  $(x_3, x_6)$  and so on ...
- ▶ Given the pair  $(x_i, x_{2i})$  we compute

$$(x_{i+1}, x_{2i+2}) = (f(x_i), f(f(x_{2i})))$$

- ▶ We stop when we find  $x_m = x_{2m}$
- ▶ Exercise: proof that if tail has length  $\lambda$  and cycle length  $\mu$

$$m = \mu \cdot \left\lceil \frac{\lambda}{\mu} \right\rceil$$

- ▶ Since  $\lambda \leq m \leq \lambda + \mu$  we see that  $m = O(\sqrt{n})$
- ▶ Detect a collision with  $O(1)$  storage

# Pollard Rho: Discrete Logarithms

- ▶ Let  $G$  denote a group of order  $n$  and let  $Q = \mathbf{d}P$
- ▶ Partition  $G$  into three sets  $S_1, S_2, S_3$  ( $\mathcal{O} \notin S_2$ )
- ▶ Define the following *random walk*

$$X_{i+1} = f(X_i) = \begin{cases} Q + X_i & X_i \in S_1, \\ 2X_i & X_i \in S_2, \\ P + X_i & X_i \in S_3. \end{cases}$$

# Pollard Rho: Discrete Logarithms

Let  $X_j = a_jP + b_jQ$ , then

$$a_{i+1} = \begin{cases} a_i & x_i \in \mathcal{S}_1, \\ 2a_i \pmod{n} & x_i \in \mathcal{S}_2, \\ a_i + 1 \pmod{n} & x_i \in \mathcal{S}_3, \end{cases}$$

and

$$b_{i+1} = \begin{cases} b_i + 1 \pmod{n} & x_i \in \mathcal{S}_1, \\ 2b_i \pmod{n} & x_i \in \mathcal{S}_2, \\ b_i & x_i \in \mathcal{S}_3. \end{cases}$$

# Pollard Rho: Discrete Logarithms

- ▶ Start with the triple

$$(X_0, a_0, b_0) = (\mathcal{O}, 0, 0)$$

- ▶ Then we have, for all  $i$ ,

$$\log_P(X_i) = a_i + b_i \log_P(Q) = a_i + b_i \mathbf{d}.$$

- ▶ Exercise: using Floyd's cycle finding algorithm we find

$$X_m = X_{2m} \Rightarrow \mathbf{d} = \frac{a_{2m} - a_m}{b_m - b_{2m}} \pmod{n}$$

# Parallel Pollard Rho (van Oorschot and Wiener)

- ▶ With  $m$  processors,  $m$  pseudo-random walks starting at

$$X_0^{(i)} = a_i P + b_i Q$$

- ▶ Each processor need to compute  $O(\sqrt{\pi n/2}/m)$  iterations
- ▶ Central server need to store all  $O(\sqrt{\pi n/2})$  points
- ▶ Define distinguished points  $S_D \subset G$  and  $\theta = |S_D|/|G|$
- ▶ Processors only send distinguished points to central server

$$O\left(\frac{\sqrt{\pi n/2}}{m} + \frac{1}{\theta}\right) \text{ time} \quad O(\theta \sqrt{\pi n/2}) \text{ space}$$

# Pollard Lambda Attack

- ▶ Assume that the discrete logarithm lies in a certain interval

$$\mathbf{d} \in [a, \dots, b].$$

- ▶ Let  $w = b - a$  denote length of interval and set  $N = \sqrt{w}$
- ▶ Divide  $G$  up into  $k \simeq \log_2(w)/2$  subsets  $S_i$  for  $0 \leq i < k$
- ▶ Let  $s_i = 2^i$  for  $0 \leq i < k$  and define pseudo-random walk

$$X_{i+1} = X_i + s_j P \text{ if } X_i \in S_j.$$

- ▶ Let  $R_0 = bP$  and  $R_i = R_{i-1} + s_j P$  for  $i = 1, \dots, N$
- ▶ Let  $c_0 = b$  and  $c_{i+1} = c_i + s_j \pmod{n}$ , then  $\log_P(R_N) = c_N$

# Pollard Lambda Attack

- ▶ Second random walk:  $H_0 = Q = \mathbf{d}P$  and compute

$$H_{i+1} = H_i + s'_j P$$

- ▶ Set  $d_0 = 0$  and  $d_{i+1} = d_i + s'_j \bmod n$ , then  $\log_P(H_i) = \mathbf{d} + d_i$
- ▶ Once path of  $H_i$  meets that of  $R_i$  it follows it
- ▶ Iterate  $M$  times until  $H_M = R_N$ , then

$$c_N = \log_P(R_N) = \log_P(H_M) = \mathbf{d} + d_M \bmod n$$

- ▶ Expected running time  $O(\sqrt{w})$  and storage  $O(\log(w))$

# Additive Reduction (Semaev, Araki/Satoh, Smart)

- ▶ Let  $E$  be an elliptic curve defined over  $\mathbb{F}_p$
- ▶  $E$  is called anomalous when  $|E(\mathbb{F}_p)| = p$
- ▶ Transfer DLOG via group homomorphism

$$E(\mathbb{F}_p), \oplus \rightarrow \mathbb{F}_p, +$$

- ▶ Computing DLOG's in  $\mathbb{F}_p, +$  is trivial using Euclides
- ▶ Uses  $p$ -adic lift and  $p$ -adic elliptic logarithm
- ▶ Requires only  $O(\log p)$  elliptic curve operations

# Multiplicative Reduction (MOV, Frey-Rück)

- ▶ Let  $E$  be an elliptic curve defined over  $\mathbb{F}_q$
- ▶ Let  $E(\mathbb{F}_q)[l]$  be prime order subgroup we work in
- ▶ Transfer DLOG via group homomorphism

$$E(\mathbb{F}_q)[l], \oplus \rightarrow \mathbb{F}_{q^k}^\times, \cdot$$

- ▶  $k$  is called *security multiplier* and is smallest  $k \in \mathbb{N}$

$$l \mid q^k - 1$$

- ▶ Note:  $k$  is order of  $q$  in  $\mathbb{Z}/(l\mathbb{Z})$ , so in general  $O(l)$

# Multiplicative Reduction (MOV, Frey-Rück)

- ▶ Weil pairing: bilinear map  $\langle \cdot, \cdot \rangle : E(\mathbb{F}_q)[l] \times E(\mathbb{F}_q)[l] \rightarrow \mathbb{F}_{q^k}^\times$
- ▶ Bilinearity:  $\langle aR, bS \rangle = \langle R, S \rangle^{ab}$
- ▶ Given  $Q = \mathbf{d}P$ , find  $T$  such that  $\langle P, T \rangle \neq 1$ , then

$$\langle Q, T \rangle = \langle \mathbf{d}P, T \rangle = \langle P, T \rangle^{\mathbf{d}}$$

- ▶ Find  $\mathbf{d}$  using sub-exponential algorithm in  $\mathbb{F}_{q^k}^\times$
- ▶ Only efficient for small  $k$ , but highly unlikely
- ▶ If  $E$  is supersingular, i.e.  $p \mid (|E(\mathbb{F}_q)| - q - 1)$  then  $k \leq 6$

# Weil Descent (Frey, Hess, Gaudry, Diem, Scholten)

- ▶ Let  $E$  be an elliptic curve over  $\mathbb{F}_{q^k}$  with  $k > 1$
- ▶ Define abelian variety  $W_E$  of dimension  $k$  over  $\mathbb{F}_q$  with

$$W_E(\mathbb{F}_q) = E(\mathbb{F}_{q^k})$$

- ▶  $W_E$  is called *Weil restriction* of  $E$
- ▶ Try to find a curve  $C$  on  $W_E$  and map the DLOG

$$E(\mathbb{F}_{q^k}) \rightarrow J_C(\mathbb{F}_q)$$

- ▶ Apply index calculus algorithms on  $J_C(\mathbb{F}_q)$
- ▶ Use  $\mathbb{F}_{p^k}$ ,  $k$  large prime and order of  $2 \in \mathbb{Z}/(k\mathbb{Z})^\times$  large

# Conclusions

- ▶ Well chosen elliptic curve  $E/\mathbb{F}_q$ , only square root attacks
  - ▶  $|E(\mathbb{F}_q)|$  is divisible by a large prime  $l$  (Pohlig-Hellman)
  - ▶  $p \nmid l$  to avoid additive reduction
  - ▶  $\text{ord}(q) \bmod l$  large to avoid multiplicative reduction
  - ▶ either  $q = p$  or  $q = p^k$  with  $k$  large prime
- ▶ More future proof than RSA/DL since no sub-exp attacks
- ▶ Be aware of extra structure ...

# References

- ▶ I. F. Blake, G. Seroussi and N. P. Smart, *Elliptic curves in cryptography*, Cambridge Univ. Press, Cambridge, (1999).
- ▶ I. F. Blake, G. Seroussi and N. P. Smart, *Advances in Elliptic Curve Cryptography*, Cambridge Univ. Press, Cambridge, (2005).
- ▶ N. Koblitz, *A Course in Number Theory and Cryptography*, Graduate Texts in Mathematics, Vol. 114, Springer (1995).
- ▶ A. J. Menezes, P. van Oorschot and S. Vanstone, *The handbook of applied cryptography*, CRC press (1997).

# References

- ▶ J. M. Pollard, *Monte Carlo methods for index computation (mod  $p$ )*, Math. Comp. **32** (1978), no. 143, 918–924.
- ▶ S. C. Pohlig and M. E. Hellman, *An improved algorithm for computing logarithms over  $GF(p)$  and its cryptographic significance*. IEEE Trans. Information Theory IT-24 (1978), no. 1, 106–110.
- ▶ N. Smart, *Cryptography: an introduction*, McGraw-Hill (2003).