Elliptic Curve Discrete Logarithm Problem and the Pollard-ρ Algorithm

Nils Reimers

07. October 2011

CARL VON OSSIETZKY universität OLDENBURG

Structure

1. Discrete Logarithm
2. Elliptic Curves
3. Elliptic Curve Cryptography
4. Pollard-ρ Algorithm
5. Runtime Analysis
6. Conclusion

Discrete Logarithm

- Be G a finite, cyclic group with operation $\otimes$.
- Define $k \in \mathbb{Z}$: $g^k := \underbrace{g \otimes \ldots \otimes g}_{k-times}$.

Discrete Logarithm

- Be G a finite, cyclic group with operation $\otimes$.
- Define $k \in \mathbb{Z}$: $g^k := \underbrace{g \otimes ... \otimes g}_{k-times}$.

Discrete Logarithm Problem

Be G a finite, cyclic group. Be $g \in G$ and be $h \in \langle g \rangle = \{g^k | k \in \mathbb{Z}\}$. Find k such that:

$$h = g^k.$$

Discrete Logarithm Problem

- Hardness of the problem (find k such that $h = g^k$) depends on the group

Example

Be $G = (\mathbb{F}_p, +)$ the additive group of the finite field $\mathbb{F}_p$. Then $g^k \underset{\text{Def}}{\equiv} k \cdot g \pmod{p}$.

Be $k \cdot g \equiv h \pmod{p}$, then $k \equiv g^{-1}h \pmod{p}$.

g^{-1} kann efficiently be determined by the extended euclidean algorithm.

Discrete Logarithm Problem $\mathbb{F}_p^*$

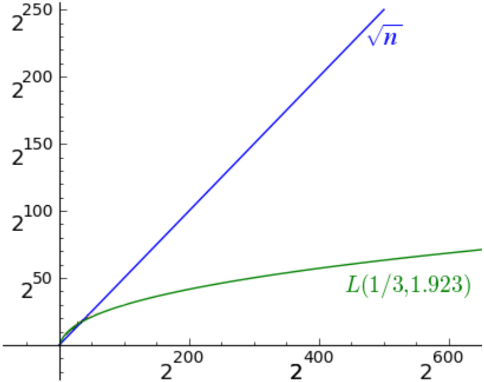
- Be G the multiplicative group $\mathbb{F}_p^*$.
- It is hard to find x such that: $g^x \equiv h \pmod{p}$.
- Security of the Diffie-Hellman key exchange is based on the hardness of this problem.

Discrete Logarithm Problem

- Every discrete logarithm problem can be solved in at max $O(\sqrt{n})$ time, n is the order of the group.

Discrete Logarithm Problem

- Every discrete logarithm problem can be solved in at max $O(\sqrt{n})$ time, n is the order of the group.
- **Problem**: The discrete logarithm problem for $\mathbb{F}_p^*$ can be solved in subexponential time.



Motivation

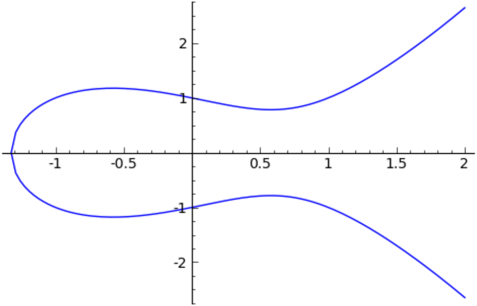
- Cryptography based on elliptic curves offer high security while using smaller key sizes compared to RSA.
- Security of the digital functions of the German machine readable passport and identify card are based on elliptic curves.

Motivation

- Cryptography based on elliptic curves offer high security while using smaller key sizes compared to RSA.
- Security of the digital functions of the German machine readable passport and identify card are based on elliptic curves.

$\Rightarrow$ Elliptic curves are an interesting and active research area.

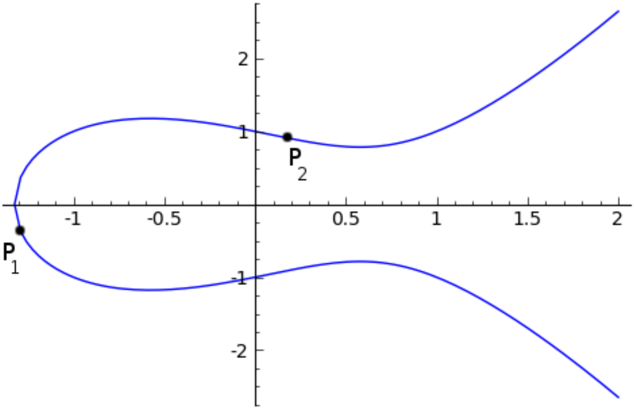
Elliptic Curve



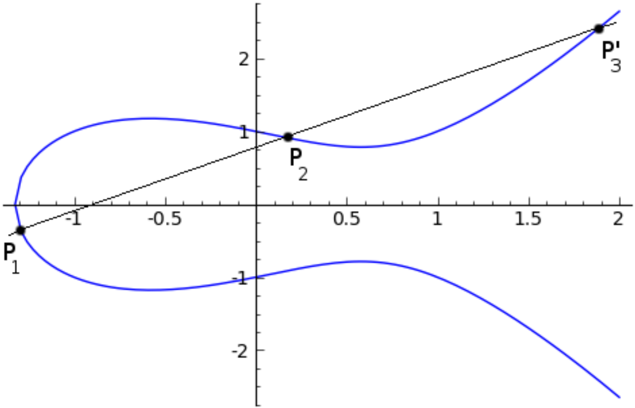
Be $\mathbb{K}$ a field and $\text{char}(\mathbb{K}) \neq 2, 3$. For $a, b \in \mathbb{K}$ and $4a^3 + 27b^2 \neq 0$ define the following set as an elliptic curve over $\mathbb{K}$:

$$E(\mathbb{K}) := \{\mathcal{O}\} \cup \{(x, y) \in \mathbb{K} \times \mathbb{K} \mid y^2 = x^3 + ax + b\}$$

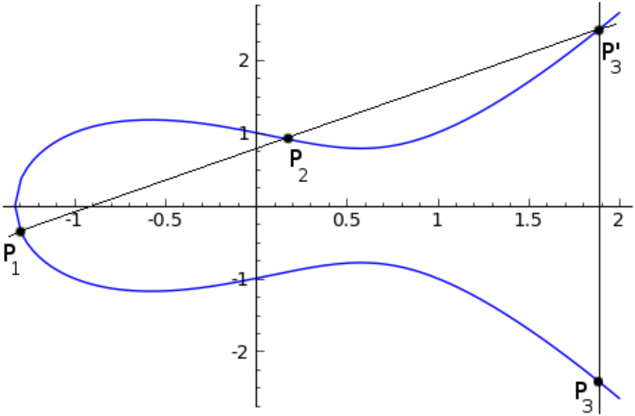
Addition on elliptic curves - 1. Case



Addition on elliptic curves - 1. Case

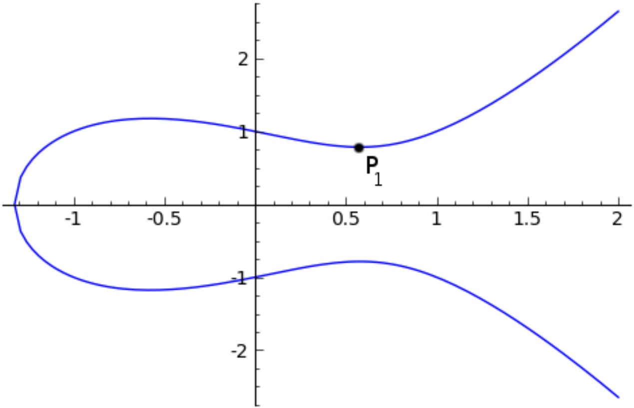


Addition on elliptic curves - 1. Case



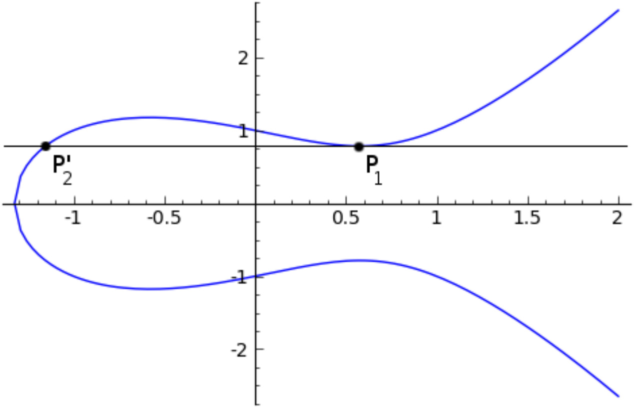
$$P_1 + P_2 = P_3$$

Addition on elliptic curves - 2. Case



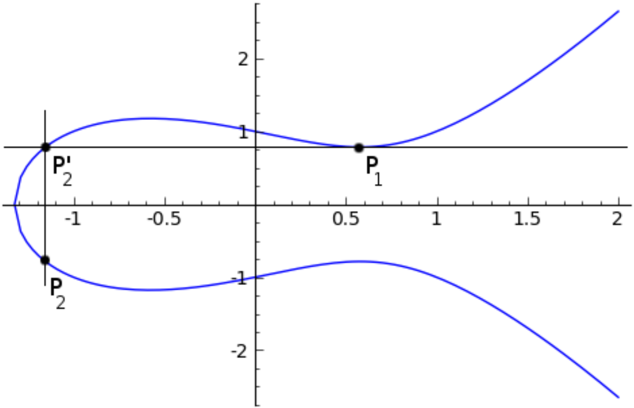
$$P_1 + P_1 = 2P_1 = P_2$$

Addition on elliptic curves - 2. Case



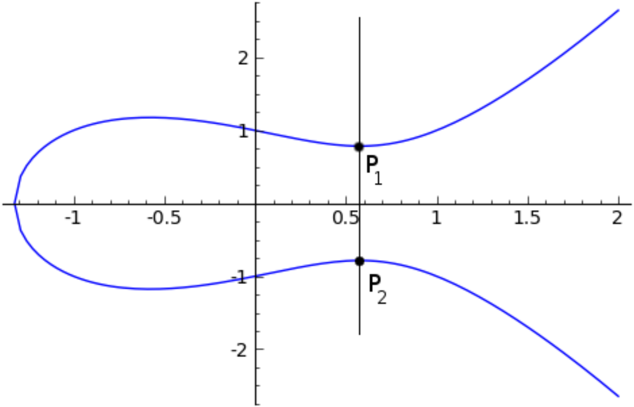
$$P_1 + P_1 = 2P_1 = P_2$$

Addition on elliptic curves - 2. Case



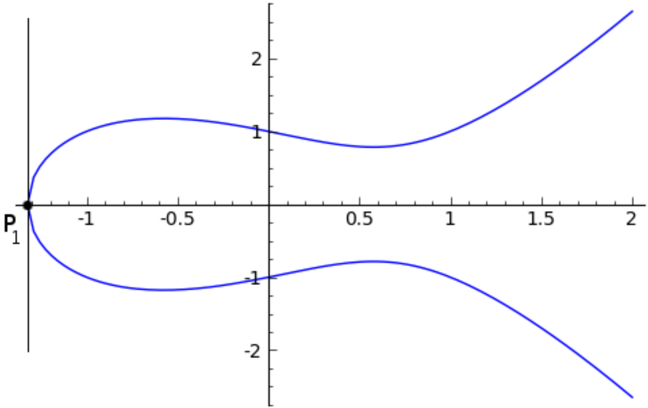
$$P_1 + P_1 = 2P_1 = P_2$$

Addition on elliptic curves - 3. Case



$$P_1 + P_2 = P_1 + (-P_1) = \mathcal{O}$$

Addition on elliptic curves - 4. Case



$$2P_1 = 2(x_1, 0) = \mathcal{O}$$

Addition on elliptic curves

Be $E(\mathbb{K})$ an elliptic curve of the form $y^2 = x^3 + ax + b$. Be $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ and $P_1, P_2 \neq \mathcal{O}$. Define $P_1 + P_2 = P_3 = (x_3, y_3)$ as:

1. If $x_1 \neq x_2$:

$$x_3 = m^2 - x_1 - x_2, \;\; y_3 = m(x_1 - x_3) - y_1, \;\text{where } m = (y_2 - y_1)(x_2 - x_1)^{-}$$

2. If $P_1 = P_2$ and $y_1 \neq 0$:

$$x_3 = m^2 - 2x_1, \;\; y_3 = m(x_1 - x_3) - y_1, \;\text{where } m = (3x_1^2 + a)(2y_1)^{-1}$$

3. If $x_1 = x_2$, but $y_1 \neq y_2$:

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

4. If $P_1 = P_2$ and $y_1 = 0$:

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

Group Law

The set $E(\mathbb{K})$ with the defined addition forms an abelian group with $\mathcal{O}$ as neutral element.

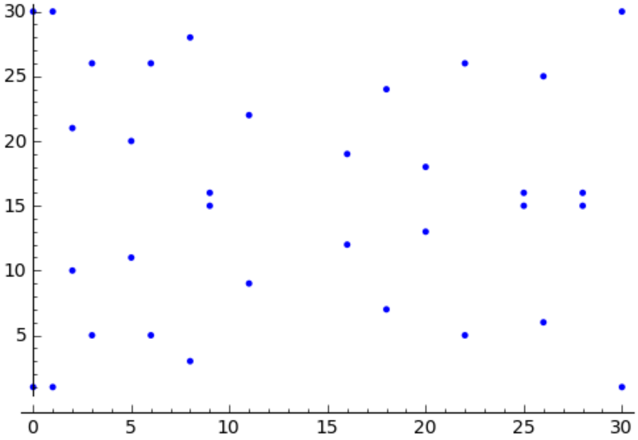
Group Law

The set $E(\mathbb{K})$ with the defined addition forms an abelian group with $\mathcal{O}$ as neutral element.

Scalarmuliplication:

$$kP := \underbrace{P + P + ... + P}_{k-times}$$

Elliptic Curves over $\mathbb{F}_p$



$$y^2 = x^3 - x^2 + 1 \text{ over } \mathbb{F}_{31}$$

Elliptic Curve Diffie-Hellman

Alice and Bob want to exchange a session key.

Elliptic Curve Diffie-Hellman

Alice and Bob want to exchange a session key.

1. Alice and Bob agreed on a secure, elliptic curve $E(\mathbb{F}_p)$ and on a point $G \in E(\mathbb{F}_p)$ with $\text{ord}(G) \approx \text{ord}(E(\mathbb{F}_p))$ and $\text{ord}(G)$ prime.

Elliptic Curve Diffie-Hellman

Alice and Bob want to exchange a session key.

1. Alice and Bob agreed on a secure, elliptic curve $E(\mathbb{F}_p)$ and on a point $G \in E(\mathbb{F}_p)$ with $\text{ord}(G) \approx \text{ord}(E(\mathbb{F}_p))$ and $\text{ord}(G)$ prime.
2. Alice randomly chooses a, computes $G_a = aG$ and sends the result to Bob.
3. Bob randomly chooses b, computes $G_b = bG$ and sends the result to Alice.

Elliptic Curve Diffie-Hellman

Alice and Bob want to exchange a session key.

1. Alice and Bob agreed on a secure, elliptic curve $E(\mathbb{F}_p)$ and on a point $G \in E(\mathbb{F}_p)$ with $\mathrm{ord}(G) \approx \mathrm{ord}(E(\mathbb{F}_p))$ and $\mathrm{ord}(G)$ prime.
2. Alice randomly chooses a, computes $G_a = aG$ and sends the result to Bob.
3. Bob randomly chooses b, computes $G_b = bG$ and sends the result to Alice.
4. Alice and Bob compute $G_{ab} = aG_b = bG_a = abG$.
5. Alice and Bob extract a session key from G_{ab}.

Elliptic Curve Diffie-Hellman - Attacker

- An attacker knows $E(\mathbb{F}_p)$, G, $G_a = aG$ and $G_b = bG$ and wants to compute $G_{ab} = abG$.
- In the chase he can compute a or b, he would be able to extract the session key.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Be $E(\mathbb{F})$ an elliptic curve, $P \in E(\mathbb{F})$ and be $Q \in \langle P \rangle = \{kP | k \in \mathbb{Z}\}$. Find k such that:

$$Q = kP.$$

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Be $E(\mathbb{F})$ an elliptic curve, $P \in E(\mathbb{F})$ and be $Q \in \langle P \rangle = \{kP | k \in \mathbb{Z}\}$. Find k such that:

$$Q = kP.$$

- Computation of kP is simple, when k is given.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Be $E(\mathbb{F})$ an elliptic curve, $P \in E(\mathbb{F})$ and be $Q \in \langle P \rangle = \{kP | k \in \mathbb{Z}\}$. Find k such that:

$$Q = kP.$$

- Computation of kP is simple, when k is given.
- To find k such that $Q = kP$ is hard.

Pollard's ρ-Algorithm

Given P and $Q = kP$. Find distinct pairs (c, d), (c', d') such that:

$$cP + dQ = c'P + d'Q$$

Pollard's ρ-Algorithm

Given P and $Q = kP$. Find distinct pairs (c, d), (c', d') such that:

$$cP + dQ = c'P + d'Q$$

$$\Rightarrow (c - c')P = (d' - d)Q = (d' - d)kP$$

$$\Rightarrow (c - c') \equiv (d' - d)k \pmod{n}$$

$$\Rightarrow k \equiv (c - c')(d' - d)^{-1} \pmod{n}$$

Pollard's Iteration Functions

Be $h : E(\mathbb{F}) \rightarrow \{0, 1, 2\}$ a hash function.

$$R_{i+1} = f(R_i) = \begin{cases} R_i + P, & \text{if } h(R_i) = 0 \\ 2R_i, & \text{if } h(R_i) = 1 \\ R_i + Q, & \text{if } h(R_i) = 2 \end{cases}$$

Pollard's Iteration Functions

Be $h : E(\mathbb{F}) \rightarrow \{0, 1, 2\}$ a hash function.

$$R_{i+1} = f(R_i) = \begin{cases} R_i + P, & \text{if } h(R_i) = 0 \\ 2R_i, & \text{if } h(R_i) = 1 \\ R_i + Q, & \text{if } h(R_i) = 2 \end{cases}$$

Start the random walk at $R_0 = P$. Define the sequence (c_i, d_i) such that $R_i = c_i P + d_i Q$. Then:

$$(c_{i+1}, d_{i+1}) = \begin{cases} (c_i + 1, d_i), & \text{if } h(R_i) = 0 \\ (2c_i, 2d_i), & \text{if } h(R_i) = 1 \\ (c_i, d_i + 1), & \text{if } h(R_i) = 2 \end{cases}$$

Teske's Adding Walk

Be $h : E(\mathbb{F}) \rightarrow \{0, 1, \ldots, s-1\}$ a hash function. Choose random integers $a_j, b_j \pmod{n}$ and compute for $j = 0, \ldots, s-1$:

$$M_j = a_j P + b_j Q.$$

Define:

$$f(R) = R + M_{h(R)}.$$

Teske's Adding Walk

Be $h : E(\mathbb{F}) \rightarrow \{0, 1, \ldots, s-1\}$ a hash function. Choose random integers $a_j, b_j \pmod{n}$ and compute for $j = 0, \ldots, s-1$:

$$M_j = a_j P + b_j Q.$$

Define:

$$f(R) = R + M_{h(R)}.$$

Then:

$$R_{i+1} = R_i + M_j = (c_i + a_j)P + (d_i + b_j)Q$$

Cycle Detection

- The expected number of iterations for Pollard's ρ-algorithm is $O(\sqrt{n})$.
- How to find a match, without storing all generated points?

Floyd's Cycle-Detection Algorithm

- We compute the pairs (R_i, R_{2i}) for $i = 1, 2, ...$ and only keep the current pair.
- These pairs can be computed easily:

$$(R_{i+1}, R_{2(i+1)}) = (f(R_i), f(f(R_{2i})))$$

Floyd's Cycle-Detection Algorithm

- We compute the pairs (R_i, R_{2i}) for $i = 1, 2, \ldots$ and only keep the current pair.
- These pairs can be computed easily:

$$(R_{i+1}, R_{2(i+1)}) = (f(R_i), f(f(R_{2i})))$$

- It can be proven, that we will find a match $R_i = R_{2i}$ and $i < d$, d the length of the ρ.

Brent's Algorithm

- Floyd's algorithm evaluates f thrice in each iteration.

Brent's Algorithm

- Floyd's algorithm evaluates f thrice in each iteration.
- Instead check in each iteration whether $R_i = R_{\lfloor \log_2 i \rfloor}$.
- On average about 36% faster than Floyd's algorithm.

Runtime Analysis

Assuming a truly random iteration function is used. Then:

- The expected length of the ρ is $\sqrt{\pi n/2} \approx 1.25\sqrt{n}$.

Runtime Analysis

Assuming a truly random iteration function is used. Then:

- The expected length of the ρ is $\sqrt{\pi n/2} \approx 1.25\sqrt{n}$.
- Floyd's algorithm requires on average $1.03\sqrt{n}$ iterations, which equals $3.09\sqrt{n}$ evaluations of f.

Runtime Analysis

Assuming a truly random iteration function is used. Then:

- The expected length of the ρ is $\sqrt{\pi n/2} \approx 1.25\sqrt{n}$.
- Floyd's algorithm requires on average $1.03\sqrt{n}$ iterations, which equals $3.09\sqrt{n}$ evaluations of f.
- Brent's algorithm requires on average $1.98\sqrt{n}$ iterations.
- Teske's improvment of Brent's algorithm requires on average $1.42\sqrt{n}$ iterations.

Experimental Results

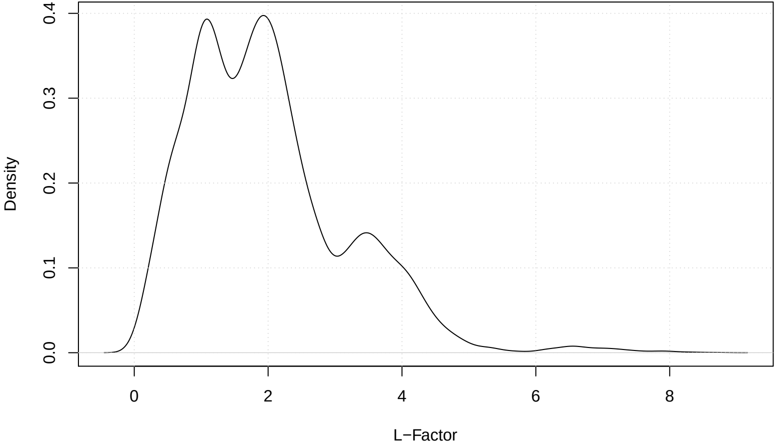
- 10,000 ECDLPs attacked by the variations of the Pollard-ρ algorithm.

Experimental Results

- 10,000 ECDLPs attacked by the variations of the Pollard-ρ algorithm.

Iteration Function	Difference to Optimum
Pollard's function	28.8%
4-adding walk	34.9%
8-adding walk	8.6%
16-adding walk	3.4%
32-adding walk	0.9%

Experimental Results



Kernel density estimation for Brent's algorithm

Parallelization

- Executing m Pollard-ρ algorithms in parallel leads to a speedup factor of $\sqrt{m}$.

Parallelization

- Executing m Pollard-ρ algorithms in parallel leads to a speedup factor of $\sqrt{m}$.
- Be D a set of rarely occurring *distinguished points*, e.g. a fixed number of leading bits of the x-coordinate equals 0.

Parallelization

- Executing m Pollard-ρ algorithms in parallel leads to a speedup factor of $\sqrt{m}$.
- Be D a set of rarely occurring *distinguished points*, e.g. a fixed number of leading bits of the x-coordinate equals 0.

Runtime Analysis:

- Be θ the probability that a random point is a distinguished point.
- Choose $\theta = \alpha m/(\sqrt{\pi n/2})$ for some α.
- Expected memory: $O(m(1+\alpha))$.
- Expected runtime: $\left(1+\frac{1}{\alpha}\right)\frac{(\sqrt{\pi n/2})}{m}$ iterations.

Conclusion

- Elliptic Curve Cryptography offer same security, while using smaller key sizes.

Security Level in Bits	Elliptic Curve Size	RSA/DSA
80	160	1024
96	192	1536
112	224	2048
128	256	3072
192	384	7680
256	512	15360