Introduction to elliptic curve cryptography
Attacks on the discrete logarithm problem

Sorina Ionica

Laboratoire MIS, Université de Picardie Jules Verne, France

May 14, 2018

The discrete logarithm problem (DLP)

The Discrete Logarithm Problem (DLP) in $\mathbb{G}$

Given $P, Q \in \mathbb{G}$ find (if it exists) λ such that

$$Q = \lambda P$$

The simplest attack: exhaustive search can find $\lambda \in \{1, \ldots, \#G\}$ in time $O(\#G)$

Generic attacks

Pohlig-Hellman reduction

Let $\mathbb{G}$ be a group such that $\mathbb{G} \simeq \prod_{i=1}^{N} \mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}$.

Let $n = \prod p_i^{\alpha_i}$ be the order of $\mathbb{G}$.

- For every i, solve the discret logarithm in $\mathbb{G}_i = \mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}$.
- Iteratively compute $\lambda \pmod{p^{\alpha_i}}$, for $i \in \{1 \dots N\}$
- Use the Chinese Remainder Theorem to compute the the discrete log in $\mathbb{G}$.

Pohlig-Hellman reduction : an example

Consider $E : y^2 = x^3 + 225x + 239$ over a finite finite of char 359.

The number of points is $\#E(\mathbb{F}_q) = 2^4 \cdot 23$ and $P = (2, 112)$ is a point of order $2^4 \cdot 23$.

The point $Q = (327, 174)$ is such that there is $x \in \mathbb{Z}$ with $xP = Q$.

- Mod 23: Solve $x(16P) = 16Q \Rightarrow x = 12 \pmod{23}$.
- Mod 2^4: Solve $x(23P) = 23Q$.

We write $x = x_0 + 2x_1 + 2^2x_2 + 2^3x_3$ and compute $23P = (71, 117)$.

$$\begin{aligned} (x_0 + 2x_1 + 2^2x_2 + 2^3x_3)(71, 117) &= (90, 343) \\ \Rightarrow 2^3x_0(71, 117) &= 2^3(90, 343) \\ \Rightarrow x_0 &= 1. \end{aligned}$$

Pohlig-Hellman reduction : an example

$$\begin{aligned}(1+2x_1+2^2x_2+2^3x_3)(71,117) &= (90,343) \Rightarrow \\ (2x_1+2^2x_2+2^3x_3)(71,117) &= (90,343)-(71,117) \Rightarrow \\ \Rightarrow x_1+2x_2+2^2x_3(93,189) &= (93,189) \Rightarrow x_1=1\end{aligned}$$

$$2x_2+2^2x_3(93,189)=0 \Rightarrow x_2=0, x_3=0.$$

To sum up, we have

$$\begin{array}{l} x \equiv 12 \pmod{23} \\ x \equiv 3 \pmod{2^4} \end{array} \Rightarrow \quad x \equiv 35 \pmod{2^4 \cdot 23}$$

The Pohlig-Hellman reduction : complexity

Let $\mathbb{G} \simeq \prod_{i=1}^{N} \mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}$ and $n = \prod p_i^{\alpha_i}$.

Denote by $DLP(k)$ the complexity of solving DLP in a group of order k.

The complexity of the Pohlig-Hellman algorithm is

$$O(\sum_{i=1}^{N} \alpha_i DLP(p_i)).$$

The DLP in $\mathbb{G}$ is as hard as the DLP in its largest group of prime order.

Le paradoxe des anniversaires

- Si on choisit k balles parmi N balles de manière aléatoire et independante, alors la probabilité que deux soient identiques est $> \frac{1}{2}$ si $k \geq \sqrt{N}$.
- Si $N = 365$, alors la probilité que parmi k personnes se trouvant dans une salle à un moment donné il y ait 2 avec le même jour d'anniversaire est $> \frac{1}{2}$ si $k > 23$.

Pollard's rho method

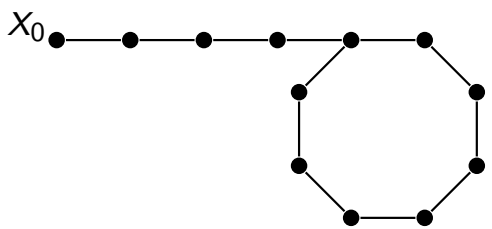
- Let $(\mathbb{G}, +)$ be a group of prime order r and $P, Q \in \mathbb{G}$. Find x s.t. $xP = Q$.
- Partition $\mathbb{G}$ into three sets S_1, S_2, S_3.
- Define the following sequence: $X_0 = P$,

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

Every $X_i = u_i P + v_i Q$, with $u_i, v_i \in \mathbb{Z}$ given by

$$u_{i+1} = \begin{cases} u_i + 1 & \text{if } X_i \in S_1 \\ 2u_i & \text{if } X_i \in S_2 \\ u_i & \text{if } X_i \in S_3 \end{cases} \qquad v_{i+1} = \begin{cases} v_i & \text{if } X_i \in S_1 \\ 2v_i & \text{if } X_i \in S_2 \\ v_i + 1 & \text{if } X_i \in S_3 \end{cases}$$

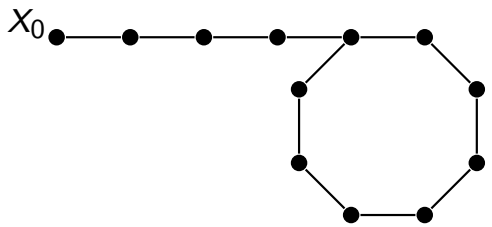
Pollard's rho method



There are m and n s.t. $X_n = X_{m+n}$.

$$u_n P + v_n Q = u_{m+n} P + v_{m+n} Q \Rightarrow x = -\frac{u_n - u_{m+n}}{v_n - v_{m+m}}$$

Pollard's rho method



There are m and n s.t. $X_n = X_{m+n}$.

By the birthday paradox $m + n \sim \sqrt{r}$.

Time complexity $O(\sqrt{r})$

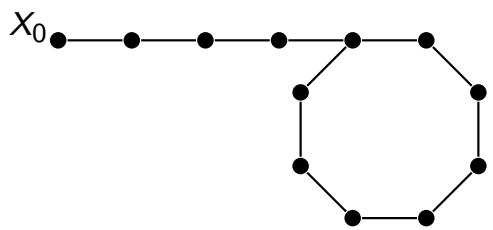
Memory complexity $O(\sqrt{r})$

Floyd's algorithm

$X_0 = P,\ Y_0 = f(P)$

$X_{i+1} = f(X_i)$

$Y_{i+1} = f(f(Y_i))$



Actually $Y_i = X_{2i}$.

There exists e s.t. $X_e = X_{2e}$ and that $n \leq e \leq n + m$.

Floyd's cycle finding algorithm

1. Compute $X = P$, $(u_X, v_x) = (1, 0)$.
2. Compute $Y = f(P)$, $(u_Y, v_Y) = f(1, 0)$.
3. while $(X \neq Y)$ do
$$X \leftarrow f(X); (u_X, v_X) = f(u_X, v_X)$$
$$Y \leftarrow f(f(Y)); (u_Y, v_Y) = f(f(u_Y, v_Y))$$
4. Return $-(u_X - u_Y)/(v_X - v_Y)$

Complexity $O(\sqrt{r})$ in time and $O(1)$ in memory