Lecture 23 "Towards the Equivalence of Breaking the Diffie-Hellman Protocol & Computing Discrete Logarithms"

On Monday, we saw an application of Elliptic curves to factoring integers: we took Pollard's $p-1$ method which relied on $p-1$ being smooth & produced an Elliptic Curve version which instead relied on $E(\mathbb{Z}/N)$ being smooth.

We will do the same thing today.

Agenda

(I) Diffie-Hellman Oracles & setup.

(II) Using DH oracle + Pohlig-Hellman to break discrete log when $\text{ord}(g) = q$ & $q-1$ is smooth

(III) when $q-1$ not smooth, find an Elliptic curve & use it instead.

(I) Diffie-Hellman Oracles

Start w/ log setup.

Suppose we have a group G of order q a prime and a generator $g \in G$ s.t.

$$G = \{ g^0, g^1, \cdots, g^{q-1} \}.$$

We have the exponential $\mathbb{F}_q \xrightarrow{\exp_g} G$

want to efficiently compute the inverse

$\log_g : G \longrightarrow \mathbb{F}_q$

Our goal is to prove computing $\log_g$ is no harder than breaking Diffe-Hellman.

So we give ourselves a Diffe-Hellman oracle that magically solves DHP & we try to use it to solve DHP.

Def A Diffe-Hellman Oracle is a map $\Theta : G \times G \to G$ such that

- $\Theta(g^x, g^y) = g^{xy}$ and
- Θ is efficient to compute.

Given this, we have access to a ring structure on G where "addition" is multiplication in G &

"multiplication" is Θ.

Negation: inverse in G.

Multiplicative inverse: $g^x \mapsto g^{x^{q-2}}$

can do this w/ $\log_2(q)$ applications of the oracle Θ.

Can perform any algorithm on implicitly given (but hidden) logarithms provided that the algorithm only involves $+, -, *, \div$, & equality checks.

Turns out this includes most algorithms.

(II) Pohlig-Hellman w/ the Oracle

Suppose c is a generator of $\mathbb{F}_q^\times$.

With access to this oracle, we can turn computing $\log_g : G \to \mathbb{F}_q$ into computing $\log_c : \mathbb{F}_q^\times \to \mathbb{Z}/q-1$.

How?

(1) Check if $\log_g(h) = 0$.

Equivalently, is $h = e$?

(2) Otherwise, try to compute w so that $h = g^{c^w}$. Then $\log_g(h) = c^w$.

We could always have done this, but w/ the Diffie-Hellman oracle, we can apply

Pohlig-Hellman in the exponent!

Write $q-1 = \prod r_i^{e_i}$.

& suppose $h = g^x$.

$$g^{x \frac{q-1}{r_i}} = g^{c^{w_{i0}} \frac{q-1}{r_i}}$$

↖ Can compute in $\log(q)$ applications of Θ

↖ Try each value of w_{i0} until we hit a match.

Then Compute $g^{x c^{-w_{i0}}}$ (one application of Θ)

& solve $g^{(x c^{-w_0}) \frac{q-1}{r_i^2}} = g^{c^{w_{i1}} \frac{q-1}{r_i}}$

by trying each value $w_{i,1} \in \{0, 1, 2, \ldots, r_i - 1\}$.
until we find a match. keep going.

Then use CRT to compute w from $w_{i,j}$'s
as in normal Pohlig-Hellman.

We deduce:

Thm If $|G| = q$ a prime & $q-1$ is smooth,
then $DHP_G = DLP_G$.

Q: But what if $q-1$ not smooth?

III Elliptic Curves

Idea Find an elliptic curve E w/ $E(\mathbb{F}_q)$ cyclic of order T where T is smooth.

Find a generator $P \in E(\mathbb{F}_q)$.

Find d s.t. $(x+d)^3 + A(x+d) + B$ is a QR, & find y s.t. $Q = (x+d, y) \in E(\mathbb{F}_q)$.

Then solve ECDLP $wP = Q$.

This can be done w/ Pohlig-Hellman b/c T is smooth.

Trick is, we only know $\left(g^{x(Q)}, g^{y(Q)}\right)$.

But Elliptic curve addition only uses $+, -, *, \div$ & $=$.

Given $R \in \mathbb{F}_q^{\times 2}$ denote by $\exp(R) \in G^{\times 2}$ the pair $\left(g^{x(R)}, g^{y(R)}\right)$.

Given $\exp(R)$ & $\exp(S)$ we can compute $\exp(R+S) = \left(g^{x(R+S)}, g^{y(R+S)}\right)$ with the oracle Θ.

Write $T = \prod r_i^{e_i}$.

Solve $\exp\left(\frac{T}{r_i} Q\right) = \exp\left(w_{i0} \frac{T}{r_i} P\right)$

for w_{i0} by trying $0, \ldots, r_i - 1$ until we find a match.

Then find $\exp(Q - w_{i0} P)$.

Solve next

$$\exp\left(\frac{T}{r_i^2}(Q - w_{i0} P)\right) = \exp\left(w_{i1} \frac{T}{r_i} P\right)$$

$$\exp\left(\frac{T}{r_i^3}\left(Q-(w_{i0}-w_{i1}r_i)\right)\right)=\exp\left(w_{i1}\frac{T}{r_i}P\right)$$

$$\vdots$$

Until we find all w_{ij}. Then use CRT to compute w s.t. $\exp(Q)=\exp(wP)$.

So we can compute $qP=Q$.

Then we can compute $x(wP)=x(Q)=x+d$.

$\Rightarrow x=x(wP)-d=\log_g(h)$!