Computational Number Theory Part IV

Cyclic Groups, related hardness assumptions

Cyclic Groups

- Generator: Let G be a group, and $|G| = p$. $g \in G$ is a generator of G if every element $a \in G$ is equal to g^x for some $x \in \{0,\ldots,p\text{-}1\}$
- Restated: every element $a \in G$ is in $\{g^0, g^1, g^2, \ldots., g^{p-1}\}$, where $g \in G$ is generator

- Groups can have multiple generators
- G is *cyclic* if it has a generator

Cyclic Groups

- If p is a prime, Z_p^* is a *cyclic* group of order p-1[1]

- All cyclic groups mutually isomorphic
- A bijective mapping f exists, s.t. f: G —> H, for cyclic groups G,H
- Of course, just because an isomorphism exists, doesn't mean it is efficiently computable…

1: And hence, for n>3, not prime-order

Cyclic Groups Useful?

- Many hard problems defined over cyclic groups
  - Discrete Logarithms
  - Diffie-Hellman (DH): Computational/decisional (CDH/DDH)

- Not all problems assumed hard in all cyclic groups, some hard in specific groups

Discrete Logarithms

- Let G be a cyclic group, $|G| = p$, let $g \in G$ be the generator. For every $h \in G$, there exists a unique $x \in Z_q$, s.t., $g^x = h$

- x is called the *discrete logarithm* of h to base g

- Why "discrete"?
  - Take on values in finite range
  - As opposed to values in (infinite) set of real numbers

Discrete Log (DL) Problem

- DLog Experiment: $DLog_{A,G}(n)$:

There exists a cyclic group, $|G| = p$, $\|p\| = n$; let $g \in G$ be a generator

Choose $h \in G$

Poly-time algorithm A: $A(G,p,g,h) \longrightarrow x$

Output 1 if $g^x = h$, else output 0

- The discrete log problem is hard w.r.t. G, if for all PPT algorithms A, there exists a negligible function, negl, s.t.:

$$\Pr[DLog_{A,G}(n) = 1] \leq negl(n)$$

DL Problem

- $DLog_{A,G}(n)$ simply says *there exists* such a group G
  - Doesn't mean $DLog_{A,G}(n)$ is hard in *all* groups!
  - Not hard in $(Z_p,+)$
- Some candidate groups in which $DLog_{A,G}(n)$ is believed hard:
  - Composite-order cyclic groups
  - Prime-order cyclic groups
  - Elliptic curve groups

Diffie-Hellman (DH) Problems

- Related, but not known to be equivalent to the DL problem
- Two problems:
  - Computational DH (CDH)
  - Decisional DH (DDH)
- General hardness relations:
  - If DL is *easy* in some group G, CDH is *easy* (in G) too
  - If DL is *hard* in some group G, is CDH *hard* too? – not known!
  - If CDH is easy in some group G, DDH is easy too
  - If DDH is easy in some group G, is CDH and DL easy too? No – counterexamples exist

First proposed by Whitfield Diffie and Martin Hellman, circa 1976

CDH Problem

- Let G be a cyclic group, and |G| = p, let generator $g \in G$, let $h_1, h_2 \in G$, such that $h_1 = g^{x_1}$, $h_2 = g^{x_2}$, let $x_1, x_2 <\!\!- Z_p$
- Informally, problem is to compute $g^{(x1 \cdot x2)}$, given (p, g, h_1, h_2)

- The CDH problem is hard relative to G, if for all PPT algorithms, A, there is a negligible function, negl, such that

$$\Pr[(g^{(x1 \cdot x2)}) <\!\!- A(G, p, g, g^{x_1}, g^{x_2})] \leq negl(n)$$

where $x_1, x_2 <\!\!- Z_p$, and n is a security parameter

Easy to see that if DL is tractable, we can easily find an $x_1 = \log_g h_1$, then do $h_2^{x_1}$, which is the right answer.

DDH Problem

- Let G be a cyclic group, and $|G| = p$, let generator $g \in G$, let $h_1, h_2 \in G$, such that $h_1 = g^{x_1}$, $h_2 = g^{x_2}$, let $h_3 = h_1^{x_2} = g^{(x1 \cdot x2)}$, let $x_1, x_2, y \leftarrow Z_p$
- Informally, problem is to distinguish $g^{(x1 \cdot x2)}$ from random g^y, given (p, g, h_1, h_2, h_3)

- The DDH problem is hard relative to G, if for all PPT algorithms, A, there is a negligible function, negl, such that

$$\left| \Pr[A(G, p, g, g^{x_1}, g^{x_2}, g^{(x1 \cdot x2)}) = 1] - \Pr[A(G, p, g, g^{x_1}, g^{x_2}, g^{y}) = 1] \right| \le negl(n)$$

where $x_1, x_2 \leftarrow Z_p$, and n is a security parameter

Easy to see if CDH is solvable, so is DDH

Group Order

- Ok, but what is p? Prime or composite? ($|G| = p$)

- Actually, DL, CDH hold in both prime/composite-order cyclic groups
- But DL considered hardest in prime-order cyclic groups
- DL (relatively) easier if $|G|=q$, and q has small prime factors[1]
- DDH easy if $|G|=q$, and q has small prime factors

1: (Pohlig-Hellman lemma, we'll see it later)

Does Order Matter?

- Marked preference for cyclic G, $|G| = p$, $p>1$ is a prime

- Because of reasons on previous slide
- Also, finding generator $g \in G$ is easy, if p is prime
- All elements of G, except identity element are generators of G!

- Finally, if we require DDH to be hard[1], we better use prime-order groups!

1: That is, if we're building a protocol that is based on the hardness of DDH

Subgroups of Z_p^*

- Ok, so we need cyclic groups of prime order

- One possibility: Z_p^*
- Is this prime order? Not for $p > 3$[1]. Ugh :-(

- What about prime-order subgroups of Z_p^* ?
- Pick 2 primes p, q, s.t., $p = rq + 1$, $r \geq 1$. Then the subgroup of *r^{th} residues modulo p* is defined as:

$$G = \{[h^r \bmod p] \mid h \in Z_p^*\}$$

- Known result: G is a subgroup of of Z_p^* of order q

1: Order of Z_p^* is p-1, if p is prime

Group Generation Algorithm

GroupGen(1^n) —> (G,g,q)

- Generate a uniform n-bit prime q
- Generate an l-bit prime p, s.t., q | (p-1) /* Use Miller-Rabin (or any) algorithm */
- Choose a uniform h, s.t., $h \in Z_p^*$ with $h \neq 1$
- Set $g = [h^{(p-1)/q} \bmod p]$
- return p, g, q, where |G| = q, and G is subgroup of Z_p^*

In practice, no need to run this, just use standardized values (recommended by NIST for specific algorithms)

Generator Example

- Consider a group $G = Z_{11}^*$, $|Z_{11}^*| = 10$. How many generators of Z_{11}^* can you find? Subgroups? Verify them

- $Z_{11}^* = \{1,2,3,4,5,6,7,8,9,10\}$
- If G is prime-order cyclic, easy — $|G|$ = no. of generators (see slide 12).
- But Z_{11}^* not prime-order cyclic. Ugh!

Example

- Candidate generator: 2

- If 2 is generator, then every $a \in Z_{11}^*$ should be $\in \{2^0, 2^1, 2^2, \ldots., 2^9\}$ (see slide 2)

- Values generated by $2^x \bmod 11$, $x \in \{0,\ldots 9\}$: $\{1,2,4,8,5,10,9,7,3,6\}$ – this is all of G

- Yes, that works; 2 is a generator

Example

- Next candidate generator: 3
- If 3 is generator, then every $a \in Z_{11}^{*}$ should be $\in \{3^0, 3^1, 3^2, \ldots, 3^9\}$
- Values generated by $3^x \bmod 11$, $x \in \{0,\ldots 9\}$: $\{1,3,9,5,4,1,3,9,5,4\}$; $\{1,3,4,5,9\} \neq G$
- 3 is not a generator of G
- But generator of subgroup $H_1 \subset G$, $H_1 = \{1,3,4,5,9\}$; $|H_1| = 5$[1]

1: Note that H_1 is a prime-order (5-order) subgroup, but $H_1 \neq Z_5^{*}$! Cool, isn't it?!

Example

- Next, try 10
- If 10 is generator, then every $a \in Z_{11}^*$ should be $\in \{10^0, 10^1, 10^2, \ldots, 10^9\}$
- Values generated by $10^x \bmod 11$, $x \in \{0, \ldots 9\}$: $\{1,10,1,10,1,10,1,10,1,10\}$; $\{1,10\} \neq G$
- 10 is not a generator of G
- But generator of subgroup $H_2 \subset G$, $H_2 = \{1,10\}$; $|H_2| = 2$

Again, H_2 is a prime-order (2-order) subgroup, but $H_2 \neq Z_2^*$!

Example

- Check if any of $\{1,4,5,6,7,8,9\} \in G$ are generators as an exercise

- Do the subgroups tally with our formula?

- Pick 2 primes p, q, s.t., $p = rq + 1$, $r \geq 1$. Then the subgroup of r^{th} *residues modulo p* is defined as:

$$G = \{[h^r \bmod p] \mid h \in Z_p^*\}$$

- Known result: G is a subgroup of of Z_p^* of order q

Example

- Verify H_1:

- $p = 11$, $q = 5$, so $r = 2$
- Compute $G' = \{[h^r \bmod p] \mid h \in Z_p^*\}$ (set of squares in this case, since $r = 2$)

- $G' = \{1^2 \bmod 11, 2^2 \bmod 11, 3^2 \bmod 11, \ldots, 10^2 \bmod 11\}$
- $G' = \{1,4,9,5,3,3,5,9,4,1\} = \{1,4,9,5,3\}$
- So yes, $G' = H_1$

Example

- Verify H_2:

- $p = 11$, $q = 2$, so $r = 5$
- Compute $G' = \{[h^r \bmod p] \mid h \in Z_p^*\}$

- $G' = \{1^5 \bmod 11, 2^5 \bmod 11, 3^5 \bmod 11, \ldots, 10^5 \bmod 11\}$
- $G' = \{1,10,1,10,1,10,1,10,1,10\} = \{1,10\}$
- So yes, $G' = H_2$

Example, etc.

- Exercise: If you find any other subgroups, verify them using the formula (just the same as we did H_1, H_2)

- Migraine-inducing? Bear with me…

- Course about math/theory underpinning crypto, after all :-)