3 May 2024 — Discrete Log & Diffie—Hellman

Announcements: Pre-enroll for Fall '24 began yesterday.

- Bowers CIS new AI Minor

$$\left.\begin{array}{l} 4780 \\ 4700 \\ \text{STSCI } 4740 \end{array}\right\} \quad \lambda x. \quad x - 1000$$

- CS 6820. Enrollment will be open to ugrads.

  [Pre-enrollment is not?]

"One-Way Function": easy to compute
hard to invert.
↳ Given x, finding x' s.t. $f(x') = f(x)$ is hard <u>on average</u> over random $x \in \{0,1\}^n$.

<u>Modular exponentiation.</u>

Given g, x, N, compute $g^x \pmod N$.

Repeated multiplication takes $\Omega(x) = \Omega(2^n)$ in n is # of bits in binary representation of x.

To compute g^x: $\pmod N$

if x even:
compute $g^{x/2}$ $\pmod N$ recursively.
square it

if x odd:
Compute $g^{\frac{x-1}{2}}$

Square it

multiply by g.

If x has n bits,

$$T(n) = T(n-1) + O(n \log n)$$

$$T(n) = O(n^2 \log n)$$

The inverse operation is the <u>discrete log</u>.

Given N and g and $g^x \pmod N$

find x.

Can again solve by repeated multiplication.

Repeated squaring <u>doesn't</u> work this time.

In fact, we believe discrete log is computationally hard (for classical computers).

We will be using g, N such that

$g^p \equiv 1 \pmod N$ for p prime.

In general, define the <u>order</u> of g mod N to be the least d s.t. $g^d \equiv 1 \pmod N$.

Powers of g: $1, g, g^2, g^3, \ldots, \underset{\underset{1}{\equiv}}{g^d}, \underset{\underset{g}{\equiv}}{g^{d+1}}, \underset{\underset{g^2}{\equiv}}{g^{d+2}}, \ldots$

The sequence repeats with period d and the remainder 1 appears as $g^x \pmod N$ if and only if x is divisible by d.

Sophie Germain primes:

p is a Sophie Germain prime if $q = 2p+1$ is also prime.

<u>Fact.</u> If p, $q=2p+1$ are both prime, the order of $4 \pmod q$ is p.

<u>Proof.</u> $$4^p = 2^{2p} = 2^{q-1} = \frac{1}{2}\left(2^q\right)$$

$$2^q = \sum_{i=0}^{q} \binom{q}{i} \qquad \binom{q}{i} = \binom{q}{q-i}$$

$$= \sum_{i=0}^{p} \binom{q}{i} + \sum_{i=p+1}^{q} \binom{q}{i}$$

These are equal

$$4^p = 2^{q-1} = \sum_{i=0}^{p} \binom{q}{i} = \sum_{i=0}^{p} \frac{q!}{i!(q-i)!}$$

divisible by q except $i=0$

$$\equiv 1 \pmod q$$

The order of 4 is a divisor of p.
It's not 1, so it must be p.

<u>Diffie Hellman key agreement</u>

Alice and Bob communicate over a public channel. Eve (attacker) listens.

<u>Goal</u>: A, B agree on an n-bit secret without Eve knowing the secret. (Unless Eve computes discrete log.)

<u>Protocol.</u> Alice chooses p, q (prime).
g such that $g^p \equiv 1 \pmod q$.

Sends (p, q, g) ~~to~~ Bob in public.

Alice picks secret, $a \in \{0, \dots, p-1\}$.

Bob picks secret, $b \in \{0, \dots, p-1\}$.

Eve

Alice

$A \equiv g^a \pmod{q}$

Bob

$B = g^b \pmod{q}$

Knows a, B

Knows A, B

Knows b, A

Compute $B^a \pmod{q}$

$\equiv$

$(g^b)^a = g^{ab}$

Compute A^b

$\equiv$

$(g^a)^b = g^{ab}$

They use g^{ab} as their secret key.

We believe g^{ab} is hard to compute, given g^a and g^b. (Diffie-Hellman assumption).

Public-Key Crypto. The communication channel is public. The key to encrypt messages is also public.

Alice generates (randomly) a pair of keys.

pk = public key used for encryption

sk = secret key -- -- decryption.

Goal. Anyone (Bob, Eve, whoever) can easily encrypt msgs to Alice.

No one can easily decrypt without knowing sk.

$$g^p \equiv 1 \pmod q$$

$$pk = (p, q, g, A = g^a) \qquad sk = a$$

Encryption. Sample $b \in \{0, \ldots, p-1\}$

$$B = g^b \pmod q.$$

$$Enc(pk, m) = (B, A^b \cdot m)$$

Decryption. Receive (B, C).

Know $\exists\, b$ s.t. $B = g^b$

$C = g^{ab} \cdot m$

Alice computes

$$B^{(p-1)a} \cdot A^b \cdot m$$

sk

C

$$= g^{b(p-1)a} g^{ab} \cdot m = g^{pab} \cdot m = m \pmod q$$

ElGamal public-key encryption.