3/15/17

L11.1

Today: Group theory review

Diffie Hellman Key Exchange

Five crypto groups: $\mathbb{Z}_p^*$, Q_p, $\mathbb{Z}_n^*$, Q_n,

Elliptic curves.

Reading: Katz-Lindell: 7, 8

Can use mult. or additive notation

Def: A (finite) abelian group $(G, \cdot)$ satisfies the following:

- Identity: $1 \in G$ s.t. $\forall a \in G$ $a \cdot 1 = 1 \cdot a = a$
- Inverse: $\forall a \in G$ $\exists b \in G$ s.t. $a \cdot b = 1$ $(b = a^{-1})$
- Associativity: $\forall a, b, c \in G$ $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- Commutativity: $\forall a, b \in G$ $a \cdot b = b \cdot a$

Recall:

Def Order(a) = least $u > 0$ s.t. $a^u = 1$ (in G).

Lagrange's Thm: In a finite group G of size t

$$\forall a \in G \quad \text{order}(a) \mid t$$

Corollary: In a finite group G of size t

$$\forall a \in G \quad a^t = 1$$

Example: $\forall a \in \mathbb{Z}_p^*$ $\quad a^{p-1} = 1 \mod p$ (Fermat's Thm)

$\forall a \in \mathbb{Z}_n^*$ $\quad a^{\varphi(n)} = 1 \mod n$ (Euler's Thm)

(since $|\mathbb{Z}_p^*| = p-1$ & $|\mathbb{Z}_n^*| = \varphi(n)$)

Recall:

Def: $\langle a \rangle = \{a^i : i \geq 0\}$ = subgroup generated by a

Def: If $\langle a \rangle = G$ then a is a generator of G, and G is cyclic

Claim: $|\langle a \rangle| = \text{order}(a)$.

Exercise: In a finite abelian group G of prime order $\forall a \in G$ if $a \neq 1$ then G is a generator of G.

Thm: $\mathbb{Z}_n^*$ is cyclic iff n is $2, 4, p^m$ or $2 \cdot p^m$

Fact: If G is a cyclic group of order t, and g is a generator, then the relation $x \longleftrightarrow g^x$ is 1-to-1 between $[0,1,..,t-1]$ & G.

$x \longmapsto g^x$ exponentiation ← ~~eff~~ !
↖ assuming mult. is eff.

$g^x \longmapsto x$ discrete logarithm (DL)

- Computing discrete logarithms (the DL problem) is assumed to be hard for "well-chosen" groups. E.g. for $\mathbb{Z}_p^*$, where p is a large random prime, or large random safe prime.

  ↗ Not in all groups! $(\mathbb{Z}_p, +)$

  - Fastest alg. takes time $> 2^{\log p^{1/3}}$ ↖ sub-exp. alg.

- Common public-key setup:

Public system parameters:

p – large prime (eg. 1024 bits)
g – generator of $\mathbb{Z}_p^*$

User: $SK = x$ random in $\{0,1,..,p-2\}$
$PK = y = g^x \pmod{p}$

Secrecy of x follows from the DL assumption that asserts that it is hard to find discrete logarithms

(Appears to be roughly as hard as factoring an integer of the same size as p)

For both, best known alg $\sim 2^{k^{1/3}}$ time

Not a thm!

- We often need to be able to <u>represent msgs as group elements</u>:

If M is a msg space & G is a group, we need an injective (1-to-1) map

$$f : M \to G$$

such that $f(m)$ can "represent" msg m.

Eg., if $p > 2^k$ then we can identify k-bit msgs with the integers $1, 2, \ldots, 2^k$ mod p (in $\mathbb{Z}_p^*$)

- In some groups this can be tricky.

Diffie-Hellman Key Exchange

Q: How to establish shared secret in presence of eavesdropper? (Eve is passive - only listens)

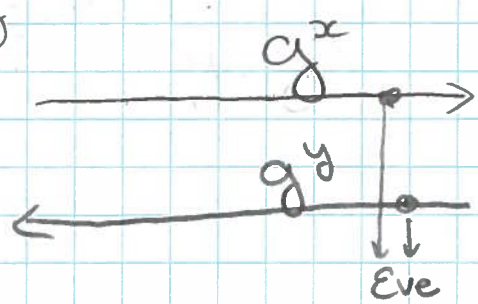
(Precursor to true public key cryptography).

- Let G be a cyclic group w. generator g
  G & g fixed and public.

A | B

- Alice chooses a random secret x from $\{0, 1, .., |G|-1\}$
- Alice computes g^x

- Bob similarly chooses secret y from $\{0, 1, .., |G|-1\}$
- Bob computes g^y



Alice computes $K = (g^y)^x$
Bob computes $K = (g^x)^y$
$= !$

- If DL hard, Eve can't compute x or y.
  That doesn't mean she can't compute K!

Computational Diffie-Hellman Assumption (CDH):

Given g^x, g^y it is hard to compute g^{xy}
(i.e. negligible chance to succeed).

CDH $\Rightarrow$ Eve doesn't learn K except w. negligible probability.

Q: Can Alice & Bob use K as a shares secret key to Encrypt and/or MAC later traffic?

Eve may learn a lot of information about K (such as 200 msb's ?).

Decisional Diffie-Hellman Assumption (DDH)

Given g^x, g^y it is hard to distinguish between g^{xy} & g^u where u is random in $\{0,1,\ldots,|G|-1\}$ w.p. $> \frac{1}{2}$ + negl.

Thm: DDH $\Rightarrow$ DH Key exchange is secure.
(Eve cannot distinguish between K and a fresh random key).

<u>Pf</u>: Follows immediately from the assumption!

Assuming DDH, we can use K to encrypt and/or MAC later.

- Don't use same K for both!
  A MAC can leak enough information to break the enc but not enough to allow forgery, and vice versa.
- Use K to derive 2 fresh keys: one for MAC & one for enc (using PRG).

<u>Next week</u>: commitment scheme & public key encryption scheme under DL (DDH/CDH).

5 Common Groups:

① $\mathbb{Z}_p^* = \{0, 1, \ldots, p-1\}$ p prime.

$\mathbb{Z}_p^*$ is always cyclic

Often, we use $p = 2q+1$ (q is prime) ← p safe prime

- Half of $\mathbb{Z}_p^*$ are generators, the others are squares ($\mathbb{Q}_p$). ← Easy to test!
- $\mathbb{Z}_p^*$ has a large subgroup of prime order (i.e. order q) ← very useful (we will see next week)

② $\mathbb{Q}_p$ = Quadratic residues (squares) mod prime p

$= \{a^2 : a \in \mathbb{Z}_p^*\} \subsetneq \mathbb{Z}_p^*$

- $|\mathbb{Q}_p| = \frac{1}{2}|\mathbb{Z}_p^*| = \frac{p-1}{2}$ ("half of $\mathbb{Z}_p^*$ are squares")
- $\mathbb{Q}_p$ is cyclic: If $\langle g \rangle = \mathbb{Z}_p^*$ then $\langle g^2 \rangle = \mathbb{Q}_p$

If $p = 2q+1$ (p is safe prime) then

$|\mathbb{Q}_p| = q$ ← prime order subgroup.

$\Rightarrow$ Any element (other than 1) generates $\mathbb{Q}_p$.

$\Rightarrow$ to find a generator, take the square of any element $a \notin \{1, p-1\}$.

③ $\mathbb{Z}_n^* = \{a \in \{1, \ldots, n-1\} : \gcd(a,n) = 1\} \leftarrow \boxed{\text{RSA}}$

<u>Def</u>: $\varphi(n) = |\mathbb{Z}_n^*|$.

If $n = p \cdot q$, where p, q distinct odd primes then $\mathbb{Z}_n^*$ is not cyclic

$$\mathbb{Z}_n^* \approx \underbrace{\mathbb{Z}_p^*}_{\substack{\text{order} \\ p-1}} \times \underbrace{\mathbb{Z}_q^*}_{\substack{\text{order} \\ q-1}}$$

the order of each element $a \in \mathbb{Z}_n^* \leq \text{lcm}(p-1, q-1) < \varphi(n)$
$= (p-1)\cdot(q-1)$

④ $Q_n = \{a^2 : a \in \mathbb{Z}_n^*\}$ = "squares mod n"
= "quadratic residues mod n"

If $n = p \cdot q$ where

$p = 2r+1$ safe prime (r prime)

$q = 2s+1$ safe prime (s prime)

then $|Q_n| = r \cdot s$ & Q_n is <u>cyclic</u>.

⑤ Elliptic Curves

Recall: In $\mathbb{Z}_p^*$ we have sub-exp. alg. for finding DL.

We would like a group G for which solving DLP takes time $2^{O(\log |G|)}$ (exp. time).

Elliptic Curves !

- Very different from $\mathbb{Z}_p^*$, $\mathbb{Z}_n^*$, Q_p, Q_n
- Appear in many diverse areas of mathematics: number theory, complex analysis, crypto, mathematical physics ...

[Koblitz, Miller 85]
Used in Bitcoins !

Def: An elliptic curve is a curve given by an equation of the form

$$y^2 = x^3 + Ax + B$$

s.t. the discriminant

$$\Delta = 4A^3 + 27B^2 \quad \text{is non-zero}$$

$\equiv$ the polynomial $x^3 + Ax + B$ has distinct roots

For reasons to be explained later we also toss in an extra point ∞.

$$E = \{(x,y) : y^2 = x^3 + Ax + B\} \cup \{\infty\}.$$

makes E a group

∞ is the identity.

the coordinates can be in any field:

$$\mathbb{Q}, \mathbb{R}, \mathbb{C}, \underbrace{GF[q]}_{\text{used in crypto!}}$$

$$E(q) = \{(x,y) \in GF[q]^2 : y^2 = x^3 + Ax + B \pmod q\} \cup \{\infty\}$$

$A, B \in GF[q]$

<u>Claim</u>: $E(q)$ is a finite group.

How is the operation $(+)$ defined ?? (coming up...)

- Best known alg that solves DLP takes time $\sim \sqrt{q}$ (exponential).

- Clearly $|E(q)| < 2q+1$

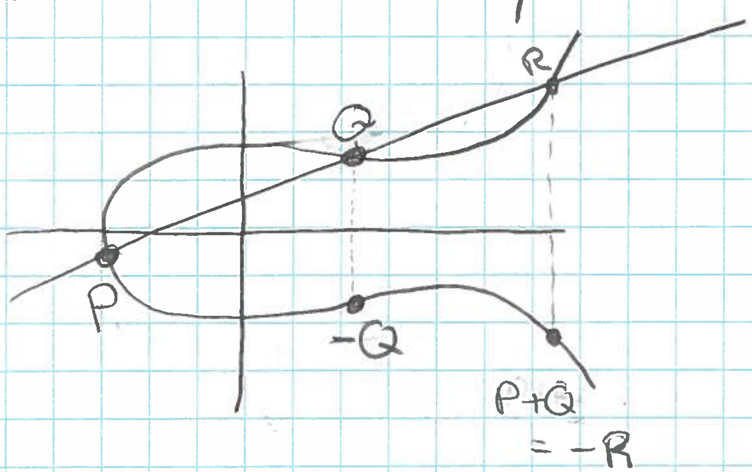
<u>Thm</u> [Hasse, 1922] $|E(q)| = q+1+t$

$$-2\sqrt{q} \leq t \leq 2\sqrt{q}$$

<u>Note</u>: We would expect $|E(q)| \approx q+1$
if x^3+Ax+B acted "randomly":
~ ½ the values are squares, each of which
has two square roots.

<u>Fact</u>: $|E(q)|$ can be computed "efficiently" (time $< (\log q)^6$)
This is important since for crypto we want $E(q)$ to
contain a subgroup of large <u>prime</u> order.

<u>Group operation</u>: Geometrically



identity $= \infty$
$\forall Q \in E(q) \quad Q + \infty = Q$
$Q + (-Q) = \infty$

- Addition of 2 points P, Q is performed by
drawing the line connecting P, Q, finding it's 3rd

intersection with $E(q)$, denoted by R, and letting

$$P+Q=-R$$

* $P+P=?$ Draw the tangent line through P, and continue as before.

This can be done over any finit field!

$$R=(x_3,y_3)$$

Let $P=(x_1,y_1)$ $\quad Q=(x_2,y_2)$, $\quad -R=P+Q=(x_3,-y_3)$

The line through P, Q can be written as

$$y=\underbrace{\left(\frac{y_2-y_1}{x_2-x_1}\right)}_{=\lambda}(x-x_1)+y_1 \quad = \lambda x+\underset{y_1-\lambda x_1}{\underset{\|}{\nu}}$$

slope $\longrightarrow \lambda$

To find R we need to find the intersection of

$E(q)$: $\quad y^2=x^3+Ax+B$

L : $\quad y=\lambda x+\nu$

$$=x^3+Ax+B-(\lambda x+\nu^2)=$$

since we already know that x_1, x_2 are solutions, so we can find x_3 by comparing

$$=(x-x_1)\cdot(x-x_2)\cdot(x-x_3)$$

$$=x^3-(x_1+x_2+x_3)\cdot x^2+(x_1x_2+x_2x_3+x_1x_3)\cdot x-x_1x_2x_3$$

$$\Longrightarrow \lambda^2=x_1+x_2+x_3$$

$$\Longrightarrow \boxed{x_3=\lambda^2-x_1-x_2} \qquad \boxed{y_3=\lambda x_3+\nu} = \lambda(x_3-x_1)+y_1$$

<u>Note</u>: If $x_1 = x_2$, $y_1 \neq y_2$ $\quad P+Q = \infty$ (vertical line)

If $P=Q$ & $y=0$ $\quad P+Q=\infty$ (vertical tangent)

* If $P=Q$ & $y_1 \neq 0$ $\quad \lambda = \dfrac{3x_1^2+A}{2y_1}$ (tangent)

$$x_3 = \lambda^2 - 2x_1$$

$$y_3 = \lambda(x_3 - x_1) + y_1$$

<u>Thm</u>:

identity → 1. $P+\infty = \infty + P = P \quad \forall P \in E(g)$

inverse → 2. $P+(-P) = \infty \quad \forall P \in E(g)$

associativity → 3. $P+(Q+R) = (P+Q)+R \quad \forall P,Q,R \in E(g)$

commutativity → 4. $P+Q = Q+P \quad \forall P,Q \in E(g)$

$\Longrightarrow$ $(E,+)$ is a finite commutative gp.

DLP seems to be very hard (requiring $\sim |E|^{1/2}$ steps) for "well-chosen" $E(g)$ (see NIST standard curves)

* Some elliptic curves admit "bilinear maps" enabling wonderful crypto (stay tuned!)