Today

Public-key Cryptography

- Basic group theory
- Diffie-Hellman key exchange
- Definition of public key cryptography.
- El-Gamal encryption scheme

Group Theory - Recap

Def: A group G consists of a set of elements & an operation $\cdot : G \times G \to G$ s.t.:

- $\forall a,b,c \quad a \cdot (b \cdot c) = (a \cdot b) \cdot c$ (associative)
- $\exists$ identity $1 \in G$ s.t. $1 \cdot a = a \cdot 1 = a \quad \forall a \in G$
- $\forall a \in G \ \exists a^{-1} \in G$ s.t. $a \cdot (a^{-1}) = 1$ (inverse)

A group is commutative if $\forall a,b \in G \quad a \cdot b = b \cdot a$

* All groups we will work with are Commutative.

Common Groups: $\mathbb{Z}_p^*$, $\mathbb{Z}_n^*$, Q_p, Q_n, Elliptic curves.

$\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$ mult. mod p

prime

$\mathbb{Z}_n^* = \{a \in \{1, \ldots, n-1\} \text{ s.t. } \gcd(a,n) = 1\}$

$n = p \cdot q$ product of 2 primes (used in RSA)

<u>Def</u>: The order of a group is the number of elements in the group

$$|\mathbb{Z}_p^*| = p-1$$
$$|\mathbb{Z}_n^*| = n-1-(p-1)-(q-1) = n-p-q+1$$
$$= (p-1)(q-1) \triangleq \varphi(n)$$

Euler's function.

<u>Note</u>: The order of $\mathbb{Z}_n^*$ is hard to compute given only n. ← used in security of RSA

(*) For crypto applications we often need a group of <u>prime</u> <u>order</u>

<u>Note</u>: $\mathbb{Z}_p^*$ is <u>not</u> prime order.

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

<u>Claim:</u> $|Q_p| = \frac{p-1}{2}$

Consider $f : \mathbb{Z}_p^* \to Q_p$ defined by $f(a) = a^2 \bmod p$.

By Fundamental Thm of Algerbra every degree d poly over a field $\mathbb{F}$ has at most d roots.

$\Rightarrow$ a^2 has only two pre-images $a, p-a$.

(since $g(x) = x^2 - x$ is a deg 2 poly over the field $GF[p]$.)

(This is not true over $\mathbb{Z}_n^*$, since it is not a field)

$\Rightarrow$ $|Q_p| = \frac{p-1}{2}$

If p is a prime st $q \triangleq \frac{p-1}{2}$ is prime then $|Q_p|$ is prime.

Such p is called <u>safe prime</u>.

<u>Recall:</u> Exponentiation can be done efficiently (repeated squaring)

Inverses can be computed efficiently

By Fermat's Little Thm: $a^{p-1} = 1 \mod p$

$$\Rightarrow a^{-1} = a^{p-2} \mod p$$

- A prime (or safe prime) can be chosen efficiently by choosing a random element in $\{0,1\}^k$ and testing if it is prime (or safe prime).

Order of Elements & Generators

* $\forall$ finite group G $\forall a \in G$, consider the subgroup

$$\langle a \rangle \triangleq \{a, a^2, \ldots, \underset{\overset{\shortparallel}{1}}{a^u}\}$$

subgroup generated by a.

Lagrange Thm: $\forall$ finite group G $\forall a \in G$ $a^{|G|} = 1$

Def: $\text{order}(a) = |\langle a \rangle| = $ least $u \geq 1$ s.t. $a^u = 1$.

Corollary: $\forall a \in G$ $\text{order}(a) \mid |G|$

(If $\text{order}(a) = u$ & $|G| = \alpha u + \beta$ $\beta \in \{1, \ldots, u-1\}$
then $1 = a^{|G|} = a^{\alpha u + \beta} = a^{\beta}$ - contradiction).

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

Def: A finite group is cyclic if $\exists$ generator $a \in G$.

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

When we use $\mathbb{Z}_p^*$ we often use it together w. a generator g so that

$f_g : x \longmapsto g^x$ is a bijection from $\{1, \ldots, p-1\}$ to $\mathbb{Z}_p^*$.

$g^x \longmapsto x$ discrete log, believed to be HARD.

In $\mathbb{Z}_p^*$ fastest alg for computing discrete log takes time $\geq 2^{\log p^{1/3}}$

sub-exp alg

How do we efficiently find a generator ?

Note: A random element in $\mathbb{Z}_p^*$ is not a generator w.p. at least $1/2$. (if it is in Q_p it is not a generator)

- It is easy to find generator in a prime order group! Every element except the identity is a generator.

- In $\mathbb{Z}_p^*$ we need to know the factorization of $p-1$ to find a generator.

Diffie-Hellman Key Exchange

(precurser to public-key crypto)

Allows Alice & Bob to share a secret key in the presence of a passive eavesdropper.

Let G a cyclic group w. generator g
(i.e. $G = \{g, g^2, \ldots, g^{|G|}\}$)

G, g fixed & public

A		B
	$\xrightarrow{g^x}$	
Choose at random $x \leftarrow \{1, \ldots, \|G\|\}$		Choose random $y \leftarrow \{1, \ldots, \|G\|\}$
	$\xleftarrow{g^y}$	

Shared Secret: $K = g^{x \cdot y}$

Computation Diffie-Hellman Assumption (CDH)

Given g^x, g^y it is hard to compute $g^{x \cdot y}$

(i.e., there is only negl probability of succeeding).

CDH $\Rightarrow$ Eve doesn't learn K except w. negl prob.

This guarantee is not strong enough to then use K as a secret key, since Eve may learn ½ the bits of K.

Decisional Diffie-Hellman Assumption (DDH)

Giving g^x, g^y it is hard to distinguish g^{xy} from g^u where u is random in $\{1, \ldots, |G|\}$.

<u>Thm</u>: DDH $\Rightarrow$ DH key exchang is secure; i.e, Eve cannot dist. between K and a fresh random key.

(Follows immediate from DDH assumption)

DDH does not hold in $\mathbb{Z}_p^*$ (HW).

We believe DDH holds in a prime order subgroup of $\mathbb{Z}_p^*$. (eg. Q_p for $p=2q+1$ safe prime).

Public-Key Encryption

Consists of 3 PPT algorithms: KeyGen, Enc, Dec.

KeyGen: Takes as input security parameter 1^λ (in unary, so that KeyGen will run in poly time). $\lambda \approx$ key-size. It outputs (PK, SK).

Enc: Takes as input (PK, m), outputs a ciphertext c
msg in msg space $\mathcal{M}$.

Dec: Takes as input (SK, c) and outputs m
ciphertext

Correctness: $\forall (PK, SK) \leftarrow \text{KeyGen}(1^\lambda)$ $\forall m \in \mathcal{M}$

$$\Pr[\text{Dec}(SK, \text{Enc}(PK, m)) = m] = 1$$

Semantic Security (CPA-security): $\forall m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

$$(PK, \text{Enc}(PK, m_0)) \cong (PK, \text{Enc}(PK, m_1))$$

- More generally, m_0, m_1 can be adv. chosen after seeing pk.

<u>Note</u>: We do not give the adv oracle access to $Enc(pk, \cdot)$ since it can be computed from pk.

El-Gamal Enc. Scheme

Let G be a cyclic group w. generator g s.t we believe DDH holds: $(g^x, g^y, g^{xy}) \approx (g^x, g^y, g^u)$

<u>KeyGen</u>: Choose $x \leftarrow \{1, .., |G|\}$

Let $sk = x$ $pk = g^x$

(Formally, choosing G & g should also be part of KeyGen: Choose safe prime p $G = Q_p$ g = any generator (any $g \in Q_p \setminus \{1\}$).)

$Enc(pk, m)$: Choose random $y \leftarrow \{1, .., |G|\}$
$m \in G$

Output $(g^y, \underbrace{g^{xy}}_{\text{DH key}} \cdot m)$

$\mathrm{Dec}(x, \underbrace{(g^y, g^{xy} \cdot m)}_{(a,b)})$ output b/a^x

<u>Semantic Security</u>: follows immediately from DDH:

$$(g^x, g^y, g^{xy} \cdot m_0) \approx (g^x, g^y, g^u)$$

$$\approx (g^x, g^y, g^{xy} \cdot m_1)$$

∎