3/24/2021
Lec 10

Today:

- Recap: Diffie-Hellman (DH) Key Exchange
- Assumptions: Discrete-Log (DL)

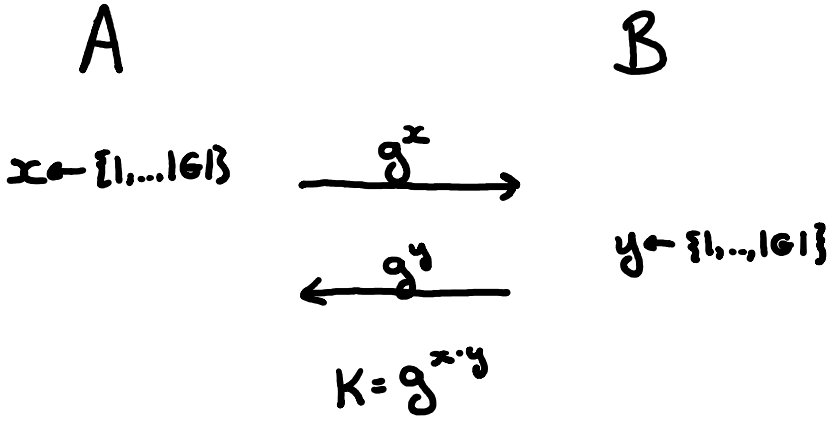
  Decisional Diffie Hellman (DDH)
- Public-Key Cryptography
- El-Gamal
- Review group theory for RSA ($\mathbb{Z}_n^*$)

  (probably won't get there in class...)

Recall: DH Key Exchange

G finite cyclic group $G = \{g, g^2, \ldots, g^{|G|}\}$

- g: generated by a single element g
- $g^{|G|} = 1$



against passive attacks

Thm: DH key exchange is secure if DDH holds in G.

DDH Assumption: Let G be a finite cyclic group w. generator g. Then

$$(g^x, g^y, g^{xy}) \cong (g^x, g^y, g^u)$$

$$x, y, u \leftarrow \{1, \ldots, |G|\}$$

* For DDH to hold G needs to be very large (eg. size 2^{1024}), since o.w. if G is small then $g^x \to x$ easy by exhaustive search.

At the minimum, the Discrete Log (DL) assumption should hold:

$g^x \to x$ hard

for random $x \leftarrow \{1, \ldots, |G|\}$

Namely: The DL assumption is that

$$f_{G,g} : x \to g^x$$

is a one-way function.

* Note that $f_{G,g} : \mathbb{Z}_{|G|} \to G$ is a bijection

isomorphism: $f(x_1 + x_2) = f(x_1) \cdot f(x_2)$ (in $\mathbb{Z}_{|G|}$ / in G)

* For $\mathbb{Z}_p^*$ the fastest DL alg runs in time $\approx 2^{\log p^{1/3}}$ ← sub-exponential

* For elliptic curves the fastest known DL alg runs in exponential time

↑ which is why elliptic curves are more efficient in practice.

Claim: DL assumption $\not\Longrightarrow$ DDH assumption

$\Longleftarrow$

Example: DDH does not hold in $\mathbb{Z}_p^*$!

$|\mathbb{Z}_p^*| = p-1$

order of $\mathbb{Z}_p^*$ = # of elements in $\mathbb{Z}_p^*$

$Q_p = \{a^2 : a \in \mathbb{Z}_p^*\}$ ← non-trivial subgroup

Q_p ← group of Quadratic Residues in $\mathbb{Z}_p^*$

Exercise: Prove that $Q_p \subseteq \mathbb{Z}_p^*$ is a group.

$|Q_p| = \frac{p-1}{2}$ half of the elements are quadratic residues.

from last lecture

* It is easy to test if $a \in \mathbb{Z}_p^*$ is in Q_p.

Recall: **Lagrange's Thm:** $\forall$ finite group G $\forall a \in G$

$a^{|G|} = 1$

Generalization of Fermat's little Thm

Given $a \in \mathbb{Z}_p^*$: $a \in Q_p$ if and only if $a^{\frac{p-1}{2}} = 1$

Breaking DDH in $\mathbb{Z}_p^*$:

Given (g^x, g^y, g^z): If $g^x \in Q_p$ & $g^z \notin Q_p$

then $z \neq x \cdot y \bmod |G|$

More generally, if G is not a prime order group then $\langle g^x \rangle$ may be a non-trivial subgroup.

↑ The (sub) group generated by the element g^x

If $g^z \notin \langle g^x \rangle$ then we know that $z \neq x \cdot y$.

This is why DDH is typically used in prime order groups (which have no non-trivial subgroups).

Example: $p = 2q + 1$ safe prime (← q prime)

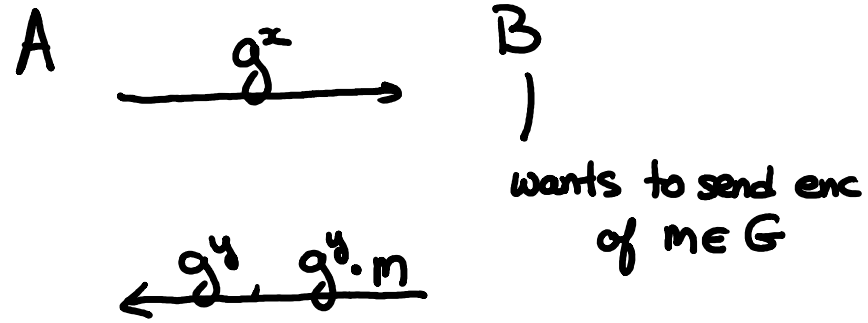
$$|Q_p| = \frac{p-1}{2} = q$$

Choose g to be any element in Q_p s.t. $g \neq 1$

g is a generator since recall:

$$\text{order}(g) \mid |G|$$ ← follows from Lagrange Thm.

DH Key Exchange implies Public-key cryptography



Bob can securely encrypt a msg to Alice (assuming DDH) without ever meeting to share a key!

Def: A public key encryption scheme consists of 3 PPT algorithms (Gen, Enc, Dec) s.t.

- $Gen(1^k)$ is a randomized alg that outputs a pair (sk, pk)
- $Enc(pk, m)$ is a randomized alg that outputs CT.
- $Dec(sk, CT)$ is a deterministic alg that outputs m s.t.

$$\Pr_{(sk,pk) \leftarrow Gen(1^k)}[\, Dec(sk, Enc(pk, m)) = m \,] = 1 - negl(k)$$

El-Gamal Encryption Scheme:

$Gen(1^\kappa)$: · Choose a random safe prime $p \in \{0,1\}^\kappa$

· Choose a random $x \leftarrow \mathbb{Z}_p^*$ and let $g = x^2 \bmod p$.

· Choose a random $s \leftarrow \mathbb{Z}_q$ $\quad q \triangleq \frac{p-1}{2}$

· Output $(PK, SK) = \left((p, g, g^s), (p, g, s) \right)$

$Enc\left((p, g, g^s), m\right)$: · Choose a random $r \leftarrow \mathbb{Z}_q$

$m \in Q_p$ · Output $(g^r, g^{sr} \cdot m)$

$Dec\left((p, g, s), (v, c)\right)$: · Output $m = c / v^s$ ← recall inverses can be computed efficiently using Lagrange Thm: $\forall a \in G,\ a^{|G|} = 1 \Rightarrow a^{|G|-1} = a^{-1}$

Correctness: $\forall\ p = 2q+1$ safe prime,

$\forall\ g \in Q_p \quad \forall s \in \mathbb{Z}_q \quad \forall m \in Q_p$

$Dec(\overset{(p,g,s)}{SK}, Enc(\overset{(p,g,g^s)}{PK}, m)) =$

$Dec(SK, (g^r, g^{sr} \cdot m)) =$

$g^{sr} \cdot m / (g^r)^s = m$ ✓

Security: Follows from DDH

(similar to the security of DH Key Exchange)

Defining Security in the Public Key Setting

Recall, in the secret key setting we considered 2 definitions:

- CPA security (against chosen plaintext attacks)
- CCA security (against chosen ciphetext attacks)

Same notions are defined in the public key setting:

- CPA: No need to give the adverary an encryption oracle, since it can encrypt on its own given PK

This notion is also known as:

Semantic security

Def: A public key encryption scheme (Gen, Enc, Dec) is **semantically secure** if $\forall$ poly-size $A = (A_1, A_2)$ $\exists$ negl μ s.t.

$$\Pr\left[A_2(PK, Enc(PK, m_b)) = b \right] \leq \frac{1}{2} + \mu(k)$$

$(PK, SK) \leftarrow Gen(1^k)$

$(m_0, m_1) = A_1(PK)$

Def: A public key encryption scheme (Gen, Enc, Dec) is **CCA secure** if $\forall$ poly-size $A = (A_1, A_2)$ $\exists$ negl μ s.t.

cannot query challenge CT

$$\Pr\left[A_2^{Dec(SK,\cdot)}(PK, Enc(PK, m_b)) = b \right] \leq \frac{1}{2} + \mu(k)$$

$(PK, SK) \leftarrow Gen(1^k)$

$(m_0, m_1) = A_1^{Dec(SK,\cdot)}(PK)$

Thm: El-Gamal is semantically secure assuming DDH holds in Q_p for a random k-bit safe prime.

k: security parameter

Group Theory for RSA

RSA crypto system performs operations in the group $\mathbb{Z}_n^*$ where $n = p \cdot q$ & p, q are random k-bit prime numbers.

$$\mathbb{Z}_n^* = \{ a \in \{1..n\} : \gcd(a,n) = 1 \},$$

multiplication is mod n.

Note: $\mathbb{Z}_n^*$ only includes elements that are co-prime to n to ensure that every element has an inverse.

How are inverses computed in $\mathbb{Z}_n^*$?

Note: We cannot use Lagrange Thm:

$$\forall a \in \mathbb{Z}_n^* \quad a^{|\mathbb{Z}_n^*|} = 1,$$

and compute $a^{-1} = a^{|\mathbb{Z}_n^*| - 1}$,

since $|\mathbb{Z}_n^*| = n - p - q + 1 = (p-1)(q-1) \triangleq \varphi(n)$ which is hard to compute given n.

Nevertheless, one can efficiently compute inverses in $\mathbb{Z}_n^*$!

Group Theory for RSA

RSA crypto system performs operations in the group $\mathbb{Z}_n^*$ where $n = p \cdot q$ & p, q are random k-bit prime numbers.

$$\mathbb{Z}_n^* = \{ a \in \{1..n\} : \gcd(a,n) = 1 \},$$

multiplication is mod n.

Note: $\mathbb{Z}_n^*$ only includes elements that are co-prime to n to ensure that every element has an inverse.

The reason every element $a \in \mathbb{Z}_n^*$ has an inverse follows from Lagrange Thm: $\forall a \in \mathbb{Z}_n^*$, $a^{|\mathbb{Z}_n^*|} = 1$,

and thus $a^{-1} = \underbrace{a^{|\mathbb{Z}_n^*|-1}}_{\text{computing this is inefficient!}} \in \mathbb{Z}_n^*$

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

which is hard to compute given n.

Computing inverses effiently in $\mathbb{Z}_n^*$ is not needed for RSA, but they <u>can</u> be computed efficiently.

Extended GCD Alg

Given $a, b \in \mathbb{Z}$ output $x, y \in \mathbb{Z}$ s.t.

$$x \cdot a + y \cdot b = \gcd(a, b)$$

To compute $a^{-1} \bmod n$: compute $x, y \in \mathbb{Z}$ s.t. $x \cdot a + y \cdot n = 1$

Let $a^{-1} = x$.