Lecture 38, Nov. 18

Cryptography

38.1 Example. Alice and Bob agrees on a permutation of alphabet, for example

1. $A \to Z$
2. $B \to Y$

38.2 Definition (Symmetric-Key Cryptosystem).

$$\begin{aligned} M &= \text{set of messages} \\ C &= \text{set of cipher text} \\ K &= \text{set of keys} \\ E &: K \times M \to C \\ D &: K \times C \to M \\ D(K, E(K, M)) &= M \text{ where } E(K, M) = C \end{aligned}$$

38.3 Definition (Advanced Encryption System). Public-key Cryptosystem

- In 1973, Ralph Markle
- In 1976, Diffie-Hellman
- In 1977, RSA public-key

38.4 Example (Merkle's puzzle).

38.5 Definition (Public-Key Cryptosystem).

$$\begin{aligned} &M, C \\ K_1 &= \text{set off public key} \\ K_2 &= \text{set of private key} \\ E &: K_1 \times M \to C \\ D &: K_2 \times C \to M \\ D(K_{private}, E(K_{public}, M)) &= M \text{ where } E(K_{public}, M) = C \\ (K_{private}, K_{public}) &\text{ is a valid pair} \end{aligned}$$

38.6 Algorithm (RSA Key Generation). *Bob*

1. *generates two large prime p, q*
2. *Compute $n = pq$.*
3. *compute $\phi(n) = (p-1)(q-1)$*
4. *Randomly choose $e \neq 1$, $gcd(e, \phi(n)) = 1$.*

5. *Solve* $ed \equiv 1 \mod \phi(n)$.

Then Bob has Public Key (n, e)*, Private Key* d*.*

Encryption: To send $m \in [0, n-1]$*. Compute* $c = m^e \mod n$*. Send* c *to Bob.*

Decryption: Compute $c^d \mod n = m'$

Claim. $m = m'$

Proof.

$$\begin{aligned} m' &\equiv c^d \mod n \\ &\equiv (m^e)^d \mod n \\ &\equiv m^{ed} \mod n \\ &\equiv m^{k\phi(n)+1} \mod n \\ &\equiv m \mod n \end{aligned}$$

Since $m \in [0, n-1]$, we have $m' = m$. □

38.7 Definition. We call A can be polynomial-time reduced to B, if we can solve A using polynomial time algorithm and we call the solver of B polynomially many times $A \leq B$

If $A \leq B, B \leq A$, we call A and B are polynomial-time equivalent $A \equiv B$

The adversary

P_1: Factor $n = pq$

P_2: Find $\phi(n)$

P_3: Find d

P_4: Given n, e and $m^e \mod n$, Find m. (called RSA Problem)

We have $P4 \leq P3 \leq P2 \leq P1$.
Claim. $P_2 \equiv P_1 \equiv P_3$

The security of RSA is based on the RSA Problem.