Public Key Cryptography.

Main limitations of symmetric key Cipher.

→ Secure Backchannel for key establishment
(need some way to share key)

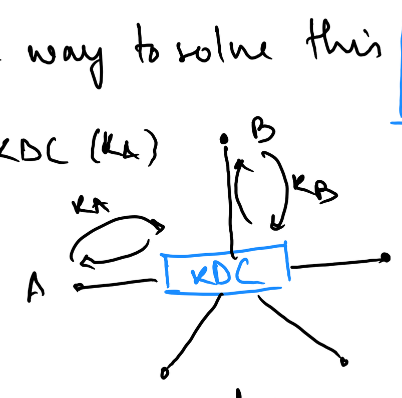
→ Too many keys (key management issues)

- Huge amount of storage
- secure Bits are costlier.

One possible way to solve this: Key Distribution Centre (KDC)

$A \to B$

- A sends to KDC (K_A)
- KDC sends to B (K_B)



n nodes.

→ only n keys req. (normally n_{C_2} req.)

Issues

→ need to trust the KDC (since KDC creates keys, it can eavesdrop)

→ does not work for Open systems (only people in list of KDC can communicate!)

→ single pt. of failure (if KDC breaks, communicaⁿ dies)

Beginnings of Public Key Cryptography.

Diffie Hellman Protocol

(A) —— (B)

Consider DLP problem.

q = size of group (for $\mathbb{Z}_p^*$ $q = p-1$)

A chooses $a \in \{0, \dots q\}$; sends g^a

B chooses $b \in \{0, \dots q\}$; sends g^b

For A

→ knows a

→ receives g^b
does $(g^b)^a = g^{ab}$

For B

→ knows b

→ receives g^a
does $(g^a)^b = g^{ab}$

eavesdropper: knows g^a & g^b; can't directly figure out g^{ab}

Proof that it's secure.

DDH Assumpⁿ → Differential Diffie-Hellman

Given all information about the group (g, q, elements)

$\forall$ PPTM A

$$\underbrace{\Pr\left[A(g^{ab}, g^a, g^b) = 1\right] - \Pr\left[A(g^k, g^a, g^b) = 1\right]}_{\text{Assume that we cannot differentiate betw. } g^{ab} \text{ \& any other element } g^k.} \leq 0$$

DDH is a universal assumpⁿ. (using DDH Assumpⁿ does not prove DHP only. Proves that every unsecure protocol can be converted to secure)

(SKE)

DDH ⇒ secure key exchange ⟺ OT ⟺ SMPC - - - -

another UA = RSA assumpⁿ

OWF ⟺ PRG ⟺ PRF ⟺ Str Cipher ⟺ Block Cipher

Hashing ⟺ MAC

Proving SKE ≤ One-way fⁿ. (OWF implies SKE exists) is harder than P = NP

** DDH true ⇒ DLP true.

DLP true ⇏ DDH true

General PKC.

(pk, sk)

pk: public server

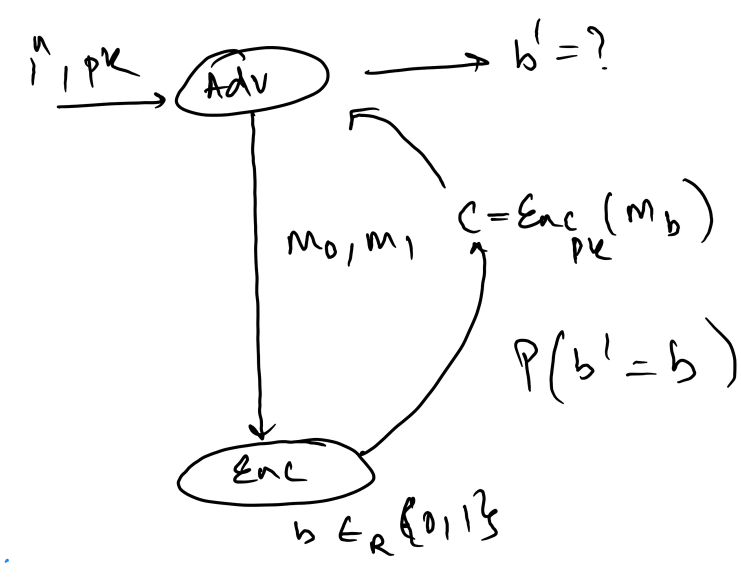
sk: secret key.

$c = Enc_{pk}(m)$

$Dec_{sk}(c) = m$

PKC has to atleast CPA-Secure.

Adversary has access to pk ⇒ can encrypt any plaintext.



$P(b' = b) \leq 1/2 + negl(n)$

El-Gamal Cryptosystem.

pk: $\langle G, q, g, g^x \rangle$

sk: x

Enc: choose r u.a.r $r \in_R [1, q]$

$c = \langle g^r, (g^x)^r \cdot m \rangle$

Dec: $\dfrac{(g^x)^r \cdot m}{(g^r)^x} = m$

known from ciphertext

from secret key.

Why El-Gamal is CPA-secure?

↳ non-deterministic

under DDH-assumpⁿ g^{xr} is not diff. from any other group element.

El-Gamal is not CCA-secure

$$\left.\begin{aligned} c_0 &= \langle g^{r_0}, g^{xr_0} m_0 \rangle \\ c_1 &= \langle g^{r_1}, g^{xr_1} m_1 \rangle \end{aligned}\right\} \therefore c_b = \langle g^{r_b}, g^{xr_b} m_b \rangle$$

$$c_b' = \langle g^{r'} \cdot g^{r_b}, g^{r'x} \cdot g^{r_b x} m' m_b \rangle$$

$$= \langle g^{(r'+r_b)}, g^{(r'+r_b)x} m' m_b \rangle$$

Apply Dec on c_b' (not challenge ciphertext)

output: $m' m_b$
↳ divide by m' = m_b

homomorphic property.

$Enc(AB) = Enc(A) \cdot Enc(B)$

normally would have to decrypt individual messages first, multiply and encrypt again

homomorphic says just multiply directly.

$\boxed{\text{homomorphic encrypⁿs are NOT CCA-secure.}}$