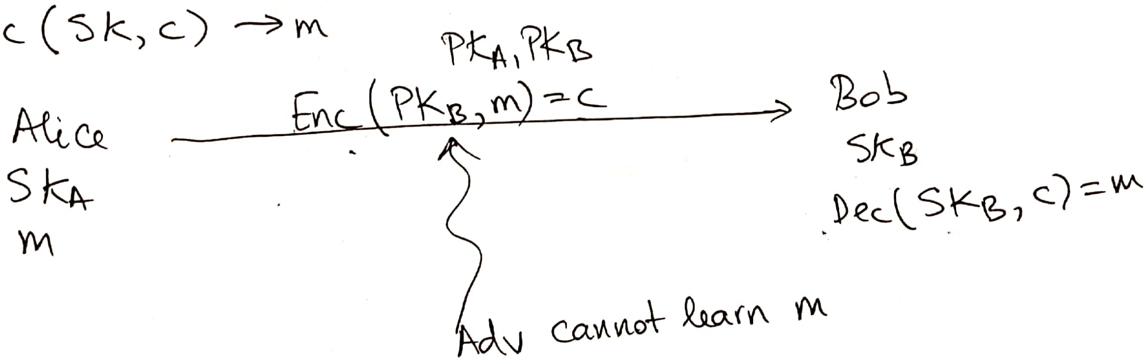
Public-key encryption

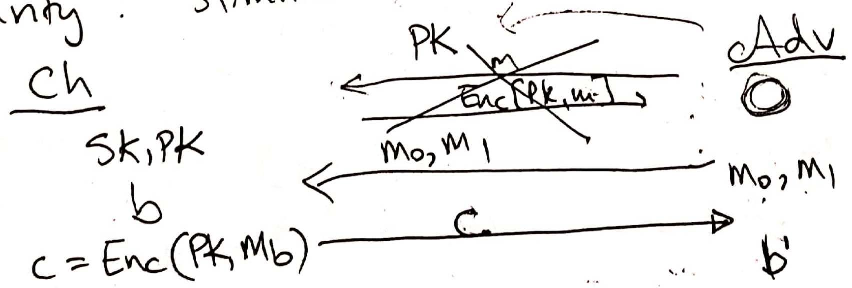
$Keygen() \to (SK, PK)$

$Enc(PK, m) \to c$

$Dec(SK, c) \to m$




1. Correctness : $Dec(SK, Enc(PK, \underline{m})) = \underline{m}$
2. Efficiency : Enc & Dec are fast to compute
3. Security : similar to IND-CPA = Semantic security




$\forall$ poly time Adv, $Pr[\text{Adv wins}] \leq \frac{1}{2} + negl$

El Gamal encryption (1985)

$[g^1, g^2 \ldots g^{p-1} \ldots [1, p-1]]$

Keygen():

- generate 2048-bit prime $p$
- generate random $g$ $\quad 1 < g < p-1$
- generate random $k$, $\quad 1 < k < p-1$

$sk = k$

- $PK = [g^k \bmod p; \; g; p]$ (public values: $g; p$)

output $(SK, PK)$

Enc(PK, m): $m \in [1 \ldots p-1]$

- pick random $r \in [1 \ldots p-1]$

$C = (g^r \bmod p; \; m \cdot PK^r \bmod p)$

$= (g^r \bmod p; \; m \cdot g^{rk} \bmod p)$

Dec(SK, C) $\quad SK = K$

$C = (R; S)$

$m = R^{-k} \cdot S \bmod p = \dfrac{S}{R^k} \bmod p$

Correctness

$(g^r \bmod p)^{-k} \cdot m \cdot g^{rk} \bmod p = g^{-rk+rk} \cdot m \bmod p \equiv m \bmod p$

$C = (g^r \bmod p \; ; \; m \cdot g^{rK} \bmod p)$

$g^{rK} \overset{c}{\approx} g^R$

Public: $PK = g^K \bmod p$

$g, p$

Discrete Log Assumption must hold

Sufficient: Decision Diffie Hellman Assumption

a, b, r randomly generated

$g^a, g^b, g^{ab} \bmod p$ indistinguishable to any Adv $g^a, g^b, g^r$

$\overset{c}{\approx}$ (computationally indistinguishables)

$g^{ab}$ is completely random

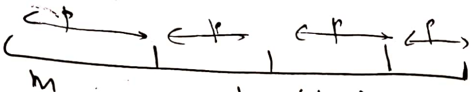
$g^{ab} \overset{c}{\approx} g^R$

Don't implement your own crypto, use tools

Scanned by CamScanner

Alice $\xrightarrow{PK_B}$ Bob

$SK_B$




1000 blocks

- slow because ~~pk~~ exponentiation is slow
- sym-key algs are much faster

~~$(PK_B)^m = g^{km} \bmod p$~~

- generates a random block cipher key $K_{BC}$
- $C_{key} = Enc(PK_B, K_{BC})$

  ↑ public-key enc (El Gamal)

$C_{key}, C_{msg} \rightarrow$ ($C_{key}$ = $Enc(PK_B, K_{BC})$)

$K_{BC} = Dec(SK_B, C_{key})$

↑ public-key dec (El Gamal)

$m = Dec(K_{BC}, C_{msg})$

$C_{msg} = Enc(K_{BC}, m)$ : can encrypt arbitrary length messages

↑ sym key enc, e.g. AES-CBC

**Hybrid encryption:**

- combines PK-enc & sym-key enc to send a long message without pre agreed upon sym key

---

PK Key exchange (DH key exchange) VS. (Agreement via) PK Encryption

- interactive ; ~~— leads to~~
- if service you are contacting is online, preferred because ~~key~~ sym key is generated locally & never sent on the network

- not interactive
- use for sending encrypted email

Scanned by CamScanner

Cryptographic hash functions

$H: \{0,1\}^* \rightarrow \{0,1\}^L$ ($*$ ← any length) ⟹ collisions exist but hard to find → no poly time Adv

SHA256 outputs 256 bits

Correctness: deterministic

$H(m)$: hash of m, digest of m, fingerprint of m

Efficiency: fast to compute $H(m)$

Security:

1. One way function:

$$\Pr[x \xleftarrow{\$} \text{random};\ y = H(x) : Adv(y) = x' \text{ s.t. } H(x') = y] = \text{negl}$$

2. Collision resistance (CR)

infeasible for any Adv to find $(x, x')$ s.t. $H(x') = H(x)$

e.g. SHA256 [past: SHA160] past hash functions: MD5, no longer (CR)

Scanned by CamScanner

Example of use:

Want to download F

Assumes that at most one of these services are not compromised.

File download service

F

$F' \neq F$

Server

$h = H(F)$ hash

User

$hash(F) \stackrel{?}{=} h$

$hash(F') \neq h$

relies on CR of hash

---

Another example:

| User | Cloud service |
|---|---|
| $hash(F) = h$ | Big file |
| store locally h (much smaller than F) ← | F |

Scanned by CamScanner