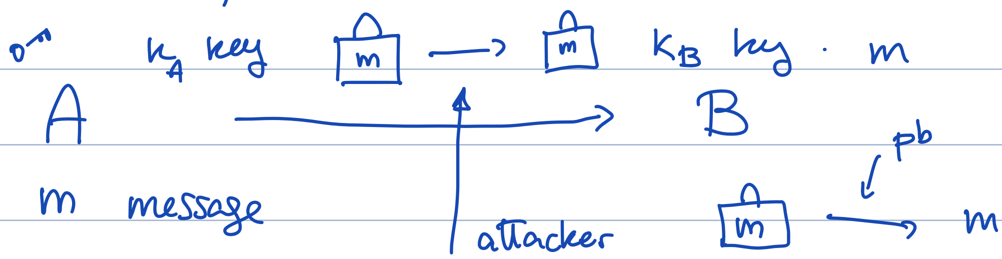
CRYPTOGRAPHY



goal: only A B read the message

ONE WAY FUNCTIONS

f : Domain $\longrightarrow$ Codomain

- Domain to be large
- $f(x)$ can be efficiently computed
- for "virtually all" y in range f
  HARD = comp. infeasible to compute x : $f(x) = y$
- f INJECTIVE

KEY SPACE

$K \rightsquigarrow \{ E_k : k \in K \}$ set of one way functions

$$E_k : \mathcal{M}_k \rightsquigarrow C_n$$

message space ($\mathcal{M}_k$) — cypher text space (C_n)

$$m \longmapsto \boxed{m}$$

we want the receiver of the cyphered text
to be able to recover m using extra info

ex (1) RSA $\quad E_k : \underset{\mathbb{Z}_n}{\overset{\mathcal{M}_k}{\shortparallel}} \longrightarrow \underset{\mathbb{Z}_n}{\overset{\mathcal{M}_k}{\shortparallel}}$

$k = (e, n) \qquad x \longmapsto x^e \bmod n$

usually n product of two large primes

$1 < e < \phi(n)$

E_K IS ONE WAY FUNCTION

efficient way to compute $x^e \bmod n$

REPETED SQUARING (SQUARE & MULTIPLY)

BUT is supposed very HARD to solve

given x^e and e find x

(2) problems in lattice theory

(3) Discrete Log Problem (DLP)

given $\alpha, y \in \mathbb{Z}_n$ find x : $\alpha^x \equiv y \bmod n$

or show that x does not exist

here $E_{(n,\alpha)} : \mathbb{Z}_n \to \mathbb{Z}_n$

$x \longmapsto \alpha^x \bmod n$

$x = \log_n \alpha$

DISCRETE LOG PROBLEM $(G, \cdot)$

Given a cyclic group $G = \langle g \rangle$

(every el. $b \in G$ can be written as $g^e = b$ for some $e \in \mathbb{Z}$)

pb : given $h \in G$ find x : $h = g^x$

main ex: $G = (\mathbb{Z}_p)^\times$ cyclic group of order $p-1$

α generator $\beta \in \mathbb{Z}_p$

Find x : $\beta = \alpha^x$ x Discrete log of β

$x = \log_\alpha \beta$

ex: $p = 19$ $(\mathbb{Z}_{19})^\times = \{ \bar{1}, \bar{2}, \bar{3} \dots, \overline{18} \}$

$\bar{2}$ is a generator

$\bar{2}^0 = \bar{1}$ $\bar{2}^1 = 2$ $\bar{2}^2 = \bar{4}$ $\bar{2}^3 = 8$

$\bar{2}^4 = \overline{16}$ $\bar{2}^5 = \overline{13}$ $\bar{2}^6 = 7$ $\bar{2}^7 = 14$ ...

$\log_2 14 = x \hookrightarrow \bar{2}^x = 14$ $x = 7$

$\log_2 7 = 6$ $\bar{2}^6 = 7$ (ex: $\log_2 12$)

⚠ not all $\alpha \in \mathbb{Z}_p^\times$ are generators

ex: $\bar{3} \in \mathbb{Z}_{11}^\times$ not a generator

- $p = 1999$ $\mathbb{Z}_{1999}^\times$ cyclic group of order 1998

$\alpha = \bar{3}$ is a generator

- compute $\beta = 3^{789} \bmod p$ (sol: 1452)
- find x s.t. $3^x \equiv 2 \bmod p$

change $p = 142 \cdot (10^{301} + 531) + 1$ find x : $3^x \equiv 2 \bmod p$

we know x exists but no exact value

Rmk : Whether DLP is hard depends on the group G

ex 1 : $G = (\mathbb{Z}_{101}, +)$

find x : $3 \cdot x \equiv 37 \mod 101$

linear Diophantine eq.

(- find inverse of 3 mod 37 $\rightsquigarrow$ 34

multiply by 37 $\rightsquigarrow$ 46

$3 \cdot 46 = 138 \equiv 37 \mod 101$

in $(\mathbb{Z}_{101} +)$ DLP is easy

<u>ex 2</u> : $G = (\mathbb{Z}_{101}^{\times}, \cdot)$ group of order $100 = 2^2 \cdot 5^2$

$3^x \equiv 37 \mod 101$ $x = 24$

$3^{24} \equiv 37 \mod 101$

we have algorithms to reduce the pb to DLP in smaller groups

usually pick $\mathbb{Z}_p^{\times}$ s.t. $p-1$ is NOT a product of powers of small primes

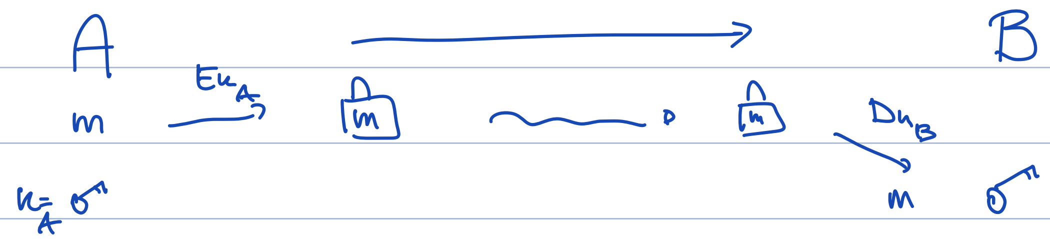
Rmk : DLP, as factoring, is an example of one way function but we don't have a proof that ANY one way function exist!

APPLICATIONS TO CRYPTO ALGORITHMS

1. key exchange DIFFIE - HELLMAN

② El - Gamal

③ Signature



① how to exchange keys

② E_{k_A} D_{k_B}

③ signatures to ensure the sender

DIFFIE-HELLMAN KEY EXCHANGE

pb : 2 parties have to agree on a common public key in a non secure channel

STEP 1 : A, B publicly agree on

p prime (SECURE) g primitive root of $\mathbb{Z}_p^\times$

STEP 2 : A secretely chooses exponent a

SENDS g^a to B

STEP 3 : B secretely chooses exponent b

SENDS g^b to A

PUBLIC KEY : $g^{ab} = \underset{A}{(g^b)^a} = (g^a)^b$

PRIVATE keys : a for A

b for B

SECURITY : attacker E wants to find private key

has to solve DLP

given g^a g^b g^{ab} find a or b

TODAY : ENCRYPTION SYSTEM

After DH A, B have

PUBLIC info : p prime , g generator $\mathbb{Z}_p^\times$

PUBLIC shared key : g^{ab}

PUBLIC keys : A : g^a B : g^b

PRIVATE keys : A : a B : b

ElGamal cryptosystem : exchange securely a message

B wants to send a message M to A

Assume : M encoded as an integer $M < p$

STEP 1 : B encrypt M as a pair $(g^b, g^{ab}M)$

and sends it to A

STEP 2 : A decrypt the message $(\underset{g^b}{\alpha}, \underset{g^{ab}\cdot M}{\beta})$ as follows

$$\beta \cdot \alpha^{-a} = g^{ab} \cdot M \cdot (g^b)^{-a} = \cancel{g^{ab}} \cdot M \cdot \cancel{g^{-ab}} = M$$

M message

SHARED KEY

$$B \longrightarrow (g^b, g^{ab} \cdot M) \longrightarrow A \qquad g^{ab} \cdot M \cdot (g^b)^{-a} = M$$

b private — B public key — a private key

SECURITY: E intercepts the communication

E sees $(g^b, g^{ab} \cdot M)$

to read M needs to compute g^{-ab}

which is available only with a private key

or solving a DLP

Rmk: if M message is long can be broken into blocks

but need a different b (private key) for each

ex: $M_1 M_2$ two blocks same private key b

$(\alpha_1, \beta_1) = (g^b, g^{ab} \cdot M_1) \qquad (\alpha_2, \beta_2) = (g^b, g^{ab} M_2)$

$$\beta_1 \cdot \beta_2^{-1} = g^{ab} \cdot M_1 \cdot (g^{ab} M_2)^{-1}$$
$$= g^{ab} \cdot M_1 \cdot M_2^{-1} \cdot g^{-ab} = M_1 \cdot M_2^{-1} \pmod p$$

So $M_2 = \beta_2 \cdot \beta_1^{-1} \cdot M_1$

Knowing M_1 one can recover M_2

EXAMPLE: public info $p = 107$ $\quad g = 2$ generator $\mathbb{Z}_{107}^{\times}$

private key A: $a = 67$

public key $2^{67} \equiv 94 \mod 107$

private key B: $b = 45$

public key $2^{45} \equiv 28 \mod 107$

M message 66 (= ASCII code for letter B)

B sends A $\left(\underset{g^b}{\overset{\shortparallel}{2^{45}}} , \underset{(g^a)^b = g^{ab}}{\overset{\shortparallel}{94^{45}}} \cdot \underset{M}{\overset{\shortparallel}{66}} \right) = \boxed{(28, 9)}$

$94^{45} \cdot 66 = 9 \mod 107$ $\mathbb{Z}_{107}^{\times}$

A reads $(\overset{\alpha}{\shortparallel}28, \overset{\beta}{\shortparallel}9)$ and computes

$\beta \cdot \alpha^{-a} = 9 \cdot 28^{\boxed{-67}} = 9 \cdot 28^{\boxed{\overset{p-1}{106} - 67}} = 9 \cdot 43 = 66$

$(\mathrm{mod}\ 107)$

A recovers $M = 66$ ✓

Implementation : usually to encode M one uses a HASH FUNCTION

$H : \underset{(\text{blocks of})}{\text{string}} \longrightarrow \mathbb{Z}_p$

ELGAMAL SIGNATURE ALGORITHM

Goal : attach data to a message M (signature)
so that the receive can verify the indentity of sender

PUBLIC DATA : p prime g generator of $\mathbb{Z}_p^{\times}$

A : public key g^a secret key a

B : g^b b

Assume message M is encoded as integer $M < p$

STEP 1: B choose random $k \in \{2, \ldots, p-2\}$
with $\gcd(k, p-1) = 1$
and computes
$$s = (M - b \cdot g^k) \cdot k^{-1} \mod p-1$$
(if $s = 0$ (unlikely) choose different k)

STEP 2: sends to A pair $(g^k, s) = (r, s)$

STEP 3: to verify identity of B
A has to verify that (A already has M)
$$g^M = g^{b \cdot r} \cdot r^s$$
$$r^s = (g^k)^s = g^{k(M - b \cdot g^k) \cdot k^{-1}} = g^{M - b \cdot g^k} = g^{M - b \cdot r}$$
$$g^{b \cdot r} \cdot r^s = g^{b \cdot r} \cdot g^{M - br} = g^M$$

SECURITY: to produce signature s
one needs B private key (DLP)
$$s = (M - b \cdot g^k) \cdot k^{-1}$$

example : $p = 11$ $g = 2$

B : public key : $2^8 \equiv 3$ private key $b = 8$

message $M = 5$

signature : choose random $k = 9$

ok since $\gcd(9, 11-1) = \gcd(9, 10) = 1$

compute $s = (M - bg^k)k^{-1} \mod p-1 = 10$

$r = g^k = 2^9 \equiv 6 \mod 11$

$s = (5 - 8 \cdot 6)k^{-1}$

$9s = (5 - 8 \cdot 6) \equiv 7 \mod 10$

solve $9 \cdot s \equiv 7 \mod 10$ ex : $s = 3$

signature $(r, s) = (6, 3)$

verification : $g^{br} \cdot r^s = \underset{\overset{\shortparallel}{3}}{2^{48}} \cdot \underset{\overset{\shortparallel}{7}}{6^3} \overset{?}{\equiv} \underset{\overset{|||}{10}}{2^5} \mod 11$ ✓