Defend of Course Works (CW) according to our curricular will be held on:
December 14 at 13:30 in 238 class.
December 19 at 15:30 in 103f and on
December 21 at 13:30 in 238 class.

Bob is thrown into the prison.
Jailor verifies all the messages Alice sends to Bob.
How Alice can tell Bob that she intends to release him by any means using her secret personal contacts?

Simmons G. J., 1985, 1994
https://scholar.google.com/scholar?start=0&q=Simmons+G.+J.&hl=en&as_sdt=0,5

Subliminal Channel - Steganography

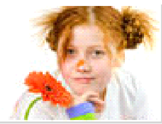
Graffiti: drawing in the walls

ElGamal Cryptosystem

1.Public Parameters generation

Gennerate strong prime number **p**.
Find a generator **g** in Z_p^*= {1, 2, 3, ..., **p**-1} using condition.
Strong prime $p=2q+1$, where q is prime, then g is a generator of Z_P^* iff
$g^q \neq 1 \bmod p$ and $g^2 \neq 1 \bmod p$.
Declare **Public Parameters** to the network **PP** = (**p**, **g**); **p** = **268435019**; **g**=**2**;
2^28=**268435456**

2.Private Keys Prk and public Public Keys PuK generation.



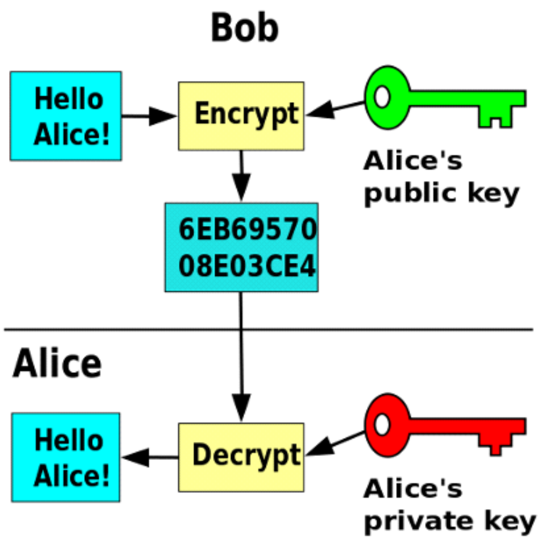
PrK = x = randi(p-1) → **x**

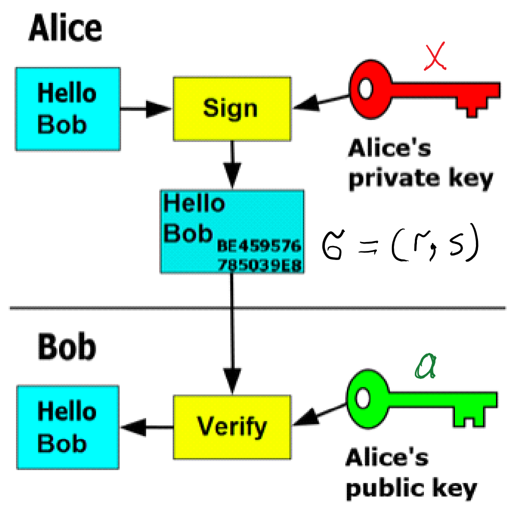
a = g^x mod **p**

PuK = a = mod_exp(g,x,p) → **a**

Asymmetric Encryption - Decryption
c=Enc(PuK_A, m)
m=Dec(PrK_A, c)

Asymmetric Signing - Verification
S=Sig(PrK_A, m)
V=Ver(PuK_A, S, m), V$\in$ {True, False} $\equiv$ {1, 0}





3.***Signature creation***

To sign any finite message ***M*** the signer performs the following steps using public parametres **PP**.

- Compute **h=H(*M*)**.
- Choose a random ***k*** such that **1 < *k* < *p* — 1** and **gcd**(***k***, ***p*** **— 1) = 1**.
- k^{-1} **mod (p-1)** computation: k^{-1} **mod (p-1) exists** if **gcd**(***k***, ***p*** **— 1) = 1**, i.e. **k** and **p-1** are relatively prime.

k^{-1} can be found using either Extended Euclidean algorithmt or Euler theorem or .....

>> k_m1=mulinv(k,p-1) % k^{-1}mod (p-1) computation.

- Compute $r=g^k \bmod p$
- Compute $s=(h-xr)k^{-1} \bmod (p-1)$ --> $h=xr+sk \bmod (p-1)$,

Signature $\sigma=(r,s)$

4.***Signature Verification***

A signature $\sigma=(r,s)$ on h-value *h* of message ***M*** is verified using Public Parameters **PP**=(**p**, **g**) and **PuK_A=a**.

1. Bob computes **h**=H(**M**).
2. Bob verifies if $1<r<p-1$ and $1<s<p-1$.
3. Bob verifies $V1=g^h \bmod p$, $V2=a^r r^s \bmod p$, and $V1=V2$.

The verifier Bob accepts a signature if all conditions are satisfied and rejects it otherwise.

5.**Correctness**

The algorithm is correct in the sense that a signature generated with the signing algorithm will always be accepted by the verifier.

The signature generation implies

$$h=xr+ks \bmod (p-1)$$

Hence Fermat's little theorem implies that all operations in the exponent are computed mod (**p**-1)

$g^h \bmod p = g^{(xr+ks) \bmod (p-1)} \bmod p = g^{xr} g^{ks} = (g^x)^r (g^k)^s = a^r r^s \bmod p$

M - message to be signed.

m - secret short message $|m| < |p-1|$ to be sent together with ElGamal signature $\sigma = (r, s)$ on M.

Using some known encoding method Encod( ) message m must be encoded to the number $z = \text{Encod}(m)$

It is assumed that $z = k$ in ElGamal signature scheme.

Then z must satisfy $\gcd(z, p-1) = 1$ (*)

If it is not the case, then m must be slightly modified to satisfy (*).

If p is strong prime, then $p = 2q+1$, when q - is prime.
Then $p-1 = 2q$ and $\gcd(z, p-1) = \gcd(z, 2q)$
Then to have $\gcd(z, 2q) = 1 \Longrightarrow z \neq iq$ & $z \neq$ even number.
If $\gcd(z, p-1) \neq 1$, then change $m \rightarrow$ recompute z in (*)
until $\gcd(z, p-1) = 1$.

Signature creation:

1. Compute $h = H(M)$!!!
2. Compute $r = g^z \bmod p$
3. Compute $z^{-1} \bmod (p-1)$
4. Compute $s = (h - x r) z^{-1} \bmod (p-1)$
5. Parameter s must satisfy $\gcd(s, p-1) = 1$ to exist $s^{-1} \bmod (p-1)$
   If p is strong prime, then $p = 2q+1$, when q - is prime.
   Then $p-1 = 2q$ and $\gcd(s, p-1) = \gcd(s, 2q)$
   Then to have $\gcd(s, 2q) = 1 \Longrightarrow s \neq iq$ & $s \neq$ even number.

If $\gcd(s, p-1) \neq 1$, then change $M \to$ recompute h in (1.) $\to$
$\to$ recompute s in (4.) until $\gcd(s, p-1) = 1$.

6. Signature is $\sigma = (r, s)$ on M.

A : M, $m \leftrightarrow z$; $\gcd(z, p-1) = 1$. $\xrightarrow[a]{M,\ \sigma = (r, s)}$ B:

$\sigma = (r, s)$; $\gcd(s, p-1) = 1$.

$h = H(M)$

$Ver(a, \sigma, h) = True$

$$s = (h - x \cdot r) \cdot z^{-1} \bmod (p-1) \qquad \cdot / z \bmod p$$

$$z \cdot s = (h - x \cdot r) \cdot \cancel{z^{-1}} \cdot \cancel{z} \bmod (p-1) \qquad \cdot / s^{-1} \bmod p$$

>> s_m1=mulinv(s,p-1)

$$z \cdot \cancel{s} \cdot \cancel{s^{-1}} = (h - x \cdot r) \cdot s^{-1} \bmod (p-1)$$

$$z = (h - x \cdot r) \cdot s^{-1} \bmod (p-1)$$

$Decod(z) = m$.

Bit commitment

Massey-Omura 3-pass Protocol https://en.wikipedia.org/wiki/Three-pass_protocol

3-pass protocol for sending messages is a framework which allows one party to securely send a message to a second party without the need to exchange or distribute encryption keys.

Bit Commitment.

Elon Musk bought a Bitcoin 25 000 USD --- 45 000 USD.

B : Should I sell my bitcoins?
A : Don't hurry, I know the price for next month.
B : Then tell me please.
A : I'll tell you next month, but if you want to know immediatly give me 3 BTC.
B : How I can know that you are not cheating?
A : We can use Bit Commitment scheme.

Public parameter PP = p = 268435019; g=2; p - may be strong prime

$A: K_A = (e_A, d_A)$ $B: K_B = (e_B, d_B)$

$e_A \cdot d_A = 1 \bmod (p-1)$ $e_B \cdot d_B = 1 \bmod (p-1)$

1) Choose e_A at random 1') -----.

$gcd(e_A, p-1) = 1$

2) $d_A = e_A^{-1} \bmod (p-1)$ 2') ----

>> dA = mulinv(eA,p-1)

M - message : Bitcoin price next month $|M| < |p-1|$.

A: Encrypts M with encryption function $\mathcal{E}$ similar to RSA encryption and finds ciphertext c_1

A: sends ciphertext c_1 to B.

$\mathcal{E}(e_A, M) = M^{e_A} \bmod p = c_1$

$\xrightarrow{c_1}$ $B: c_2 = \mathcal{E}(e_B, c_1) = c_1^{e_B} \bmod p$

$\xleftarrow{c_2}$

After 1 month: Elon Musk declared that that payments for electrocar Tesla can be made in bitcoins (BTC) 68 000 USD.

$A: c_3 = \mathcal{E}(d_A, c_2) = c_2^{d_A} \bmod p$

$\xrightarrow{c_3}$ $B: c_4 = c_3^{d_B} \bmod p =$

$= (c_2^{d_A})^{d_B} = ((c_1^{e_B})^{d_A})^{d_B} =$

$= (((M^{e_A})^{e_B})^{d_A})^{d_B} \bmod p =$

$= M^{(e_A d_A)(e_B d_B) \bmod (p-1)} \bmod p =$

$e_A \cdot d_A \bmod (p-1) = 1$ & $e_B \cdot d_B \bmod (p-1) = 1$

$= M^{1 \cdot 1} \bmod p = M = 65\,000$ \$

A: $3\ BTC \times 65\,000\,\$ = 195\,000\,\$$

B: $68\,000\,\$ - 25\,000\,\$ = 43\,000\,\$$

$10\ BTC \times 43\,000\,\$ = ---$

Compare qualitatively Massey-Omura 3-pass Protocol with bit commitment based on H-functions.

Let M be a forecasted bitcoin price

A: 1. $h = H(M)$ $\xrightarrow{\quad h \quad}$ B: waited for 1 monts

After 1 month $\xrightarrow{\quad M \quad}$ Verifies if

$h' = H(M) = h.$