Dear Students.
The Mid-Term Exam (MTE) will be held on Monday, 4-th of November, at 17:00.
Part of the students will be allowed to pass the MTE distantly through the Zoom:
https://liedm.zoom.us/j/9999112448
Passcode: 12345678
Those who will participate contactly must to bring their own computers to connect to the Zoom,
and to launch the Octave software with installed my .m files on them.
Otherwise you must to install and launch Octave in class computer together with installed my .m files on them.
During the MTE you must solve 2 problems:

1. Diffie-Hellman Key Agreement Protocol - DH KAP.
2. Man-in-the-Middle Attack (MiMA) for Diffie-Hellman Key Agreement Protocol - DH KAP.

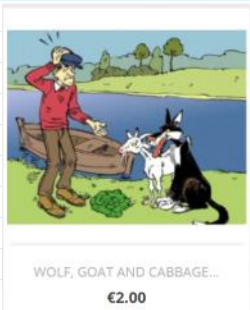
The problems are presented in the site:
imimsociety.net
In section 'Cryptography':
Cryptography (imimsociety.net)

Please register to the site and after that you receive 10 Eur virtual money to purchase the problems.
If the solution is successful then you are invited to press the green button [Get reward].
Then 'Knowledge bank' will pay you the sum twice you have payed.
So if the initial capital was 10 Eur of virtual money and you buy the problem of 2 Eur, then if the solution is correct your budget will increase up to 12 Eur.

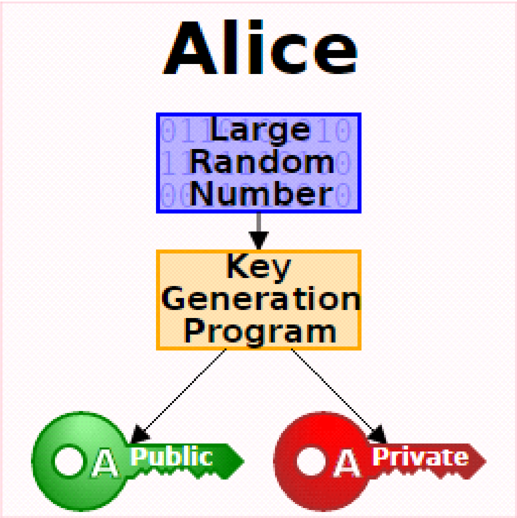
You can solve the problems in imimsociety as many time as you wish to better prepare for MTE.
I advise you to try at first to solve the problem in 'Intellect' section to exercise the brains.
It is named as 'WOLF, GOAT AND CABBAGE TRANSFER ACROSS THE RIVER ALGORITHM'.
<https://imimsociety.net/en/home/15-wolf-goat-and-cabbage-transfer-across-the-river-algorithm.html>

The more details I explain in may 5-th in 2024.10.07.

The pictures of problems listed above are the following.



Asymmetric - Public Key Cryptography



PrK and PuK are related

PuK = **F**(PrK)

F is **one-way function** (**OWF**)

Having PuK it is infeasible to find

PrK = $\mathbf{F}^{-1}$(PuK)

F(***x***)=***a*** is **OWF**, if:

1.It easy to compute ***a***, when ***F*** and ***x*** are given.

2.It is infeasible compute ***x*** when ***F*** and ***a*** are given.

PrK = *x* <-- randi ==> PuK = *a* = g^x mod p

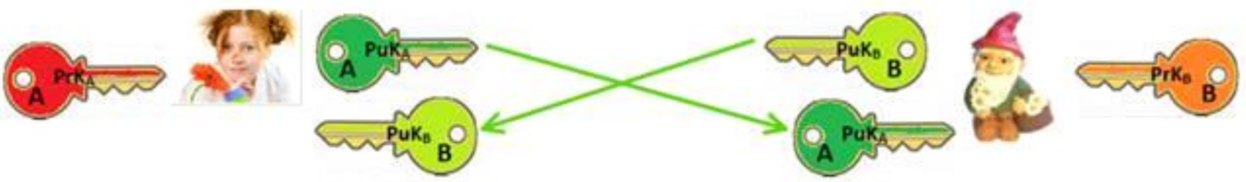
Public Parameters PP = (p, g)

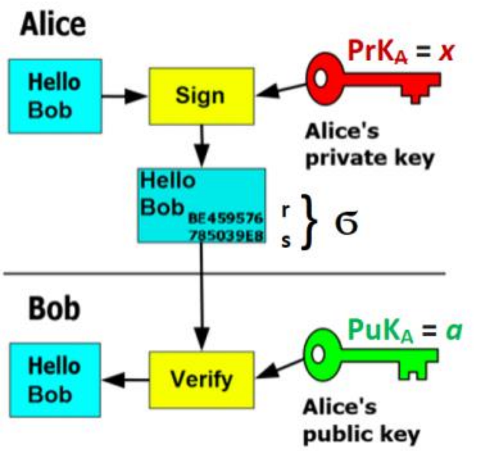
$p \sim 2^{2048} \rightarrow |p| \cong 2048$ bits

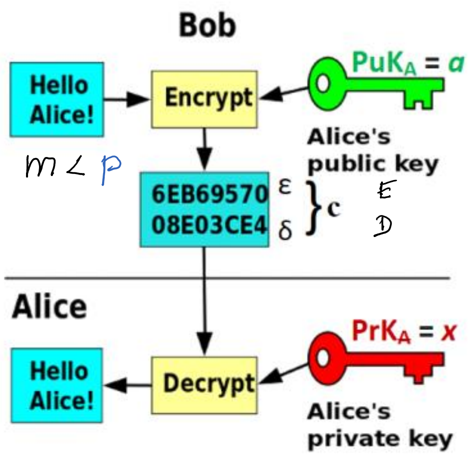
$p \sim 2^{28} \rightarrow |p| \cong 28$ bits

Threaths of insecure PrK generation

$\mathbb{Z}_p^* = \{1, 2, 3, \dots, p-1\}$; $*$ mod p







Asymmetric Encryption-Decryption: **El-Gamal Encryption-Decryption**

p=268435019; g=2;

Let message ***m~*** needs to be encrypted, then it must be encoded in decimal number ***m***: 1< ***m*** < ***p***.
E.g. ***m*** = 111222. Then ***m* mod *p*** = ***m***.

27 mod 54 = 27

27 mod 21 = 6 ≠ 27

A : $\xrightarrow{PuK_A = a}$ B : is able to encrypt m to A: $m < p$

Turing → ~~Enigma~~

~~Turing~~ $\to$ ~~Enigma~~

$\mathcal{B}$: $i \leftarrow randi(\mathcal{I}_{p-1})$ $\qquad \mathcal{I}_{p-1} = \{0, 1, 2, ..., p-2\}$ $^{*,+,-}$ $mod\ (p-1)$

$$\left.\begin{array}{l} E = m \cdot a^i \bmod p \\ D = g^i \bmod p \end{array}\right\} c = (E, D) \longrightarrow$$

$\mathcal{A}$: is able to decrypt $c = (E, D)$ using her $PrK_A = x$.

1. $D^{-x \bmod (p-1)} \bmod p$
2. $E \cdot D^{-x} \bmod p = m$

$(-x) \bmod (p-1) = (0 - x) \bmod (p-1) =$

$= (p - 1 - x) \bmod (p-1)$

$(p-1) \bmod (p-1) = 0$ since $\quad -\dfrac{p-1}{\dfrac{p-1}{0}} \Big| \dfrac{p-1}{1}$

$(-x) \bmod (p-1) = (p-1-x)$

$D^{-x} \bmod p = D^{p-1-x} \bmod p$

```
>> D_mx = mod_exp(D, p-1-x, p)
>> mx = mod(-x, p-1)
>> D_mx = mod_exp(D, mx, p)
```

Correctness

$Enc(PuK_A = a, i, m) = c = (E, D) = (E = m \cdot a^i \bmod p; D = g^i \bmod p)$

$Dec(PrK_A = x, c) = E \cdot D^{-x} \bmod p = m \cdot a^i \cdot (g^i)^{-x} \bmod p =$

$= m \cdot (\underbrace{g^x}_{a})^i \cdot g^{-ix} = m \cdot g^{xi} \cdot g^{-ix} = m \cdot g^{xi-ix} \bmod p = m \cdot g^0 \bmod p =$

$= m \cdot 1 \bmod p = m \bmod p = m = 111222$

since $m < p$

If $m > p \to m \bmod p \neq m$; $27 \bmod 5 = 2 \neq 27$. ASCII: 8 bits per char.

If $m < p \to m \bmod p = m$; $19 \bmod 31 = 19$. $\quad \dfrac{2048}{8} = 256$ char.

Decryption is correct if $m < p$.

Large file encryption $\longrightarrow$ Hybrid encryption

Authenticated Key Agreement Protocol using ElGamal Encryption and Signature.
Hybrid encryption for a large files combining asymmetric and symmetric encryption method.

Hybrid encryption. Let ***M*** be a large finite length file, e.g. of gigabytes length.
Then to encrypt this file using asymmetric encryption is extremely ineffective since we must split it into millions of parts having 2048 bit length and encrypt every part separately.
The solution can be found by using **asymmetric encryption** together with **symmetric encryption**, say AES-128.
It is named as **hybrid encryption method**.
For this purpose the **Key Agreement Protocol** (**KAP**) using **asymmetric encryption** for the same symmetric secret

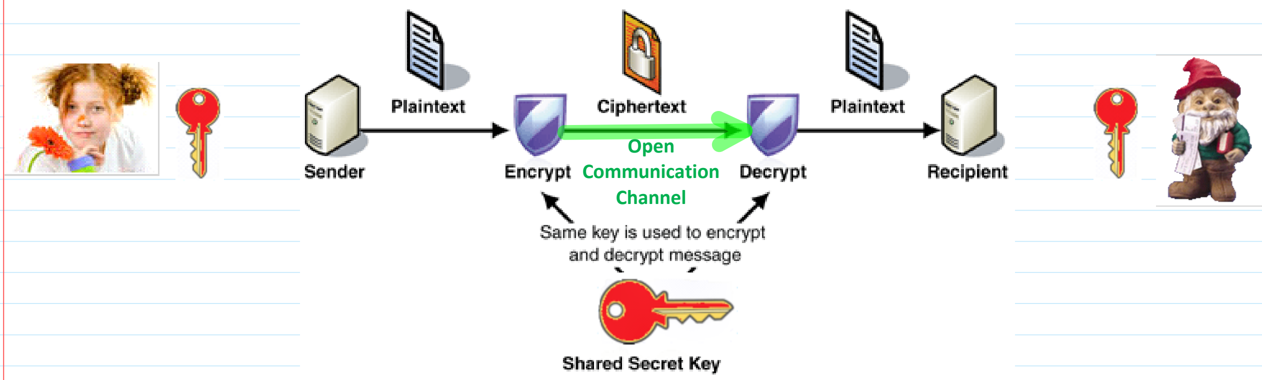
key ***k*** agreement must be realized and encryption of ***M*** realized by **symmetric encryption** method, say AES-128.

AKAP: Symmetric Enc & Asymmetric Enc & Digital Sign

Hybrid Encryption

How to encrypt large data file M: Hybrid enc-dec method.

1. Parties must agree on common symmetric secret key k.
   for symmetric block cipher, e.g. AES-128, 192, 256 bits.



A: $PrK_A = x$; $PuK_A = a$.
$PuK_B = b$.

B: $PrK_B = y$; $PuK_B = b$.
$PuK_A = a$.

To

1) $k \leftarrow randi(2^{128})$
$i_k \leftarrow randi(2^{128})$

$Enc(PuK_B = b, i_k, k) = c = (E, D)$

2) M - large file to be encrypted

$E_k(M) = AES_k(M) = G$

3) Signs ciphertext G

3.1) $h = H(G)$

3.2) $Sign(PrK_A = x, h) = \sigma = (r, s)$

$\xrightarrow{c,\, G,\, \sigma,\, PuK_A,\, Cert_A}$

1.1. Verify if PuK_A and $Cert_A$ are valid?

1.2. Verify if σ on $h = H(G)$ is valid?

$h' = H(G)$

$Ver(PuK_A, \sigma, h') = True$

2. $Dec(PrK_B, c) = k$

3. $D_k(G) = AES_k(G) = M$.

A was using so called encrypt-and-sign (E-&-S) paradigm.
(E-&-S) paradigm is recomended to prevent so called
Choosen Ciphertext Attacks - CCA: it is most strong attack
but most complex in realization.

ElGamal encryption is probabilistic: encryption of the
same message m two times yields the different cyphertexts

same message m two times yields the different ciphertexts c_1 and c_2.

1-st encryption:

$i_1 \leftarrow randi(\mathbb{Z}_{p-1})$

$\left.\begin{aligned} E_1 &= m \cdot a^{i_1} \bmod p \\ D_1 &= g^{i_1} \bmod p \end{aligned}\right\} c_1 = (E_1, D_1)$

2-nd encryption

$i_2 \leftarrow randi(\mathbb{Z}_{p-1})$

$\left.\begin{aligned} E_2 &= m \cdot a^{i_2} \bmod p \\ D_2 &= g^{i_2} \bmod p \end{aligned}\right\} c_2 = (E_2, D_2)$

$i_1 \neq i_2$

$c_1 \neq c_2$

Enigma

Necessity of probabilistic encryption.

Encrypting the same message with textbook RSA always yields the same ciphertext, and so we actually obtain that any deterministic scheme must be insecure for multiple encryptions.

Tavern episode

Enigma

Administrator

```
> p=int64(268435019)
p = 268435019
>> g=2
g = 2
>> m=111222
m = 111222
```

Alice

```
>> x=int64(randi(p-1))
x = 237315052
>> a=mod_exp(g,x,p)
a = 47080507
```

Bob

```
>> m=111222
m = 111222
>> i=int64(randi(p-1))
i = 177611802
>> a_i=mod_exp(a,i,p)
a_i = 105902610
>> E=mod(m*a_i,p)
E = 39890719
>> D=mod_exp(g,i,p)
D = 138689606
```

Alice

```
>> mx=mod(-x,p-1)
mx = 31119966
>>
>> D_mx=mod_exp(D,mx,p)
D_mx = 95323339
>> mm=mod(E*D_mx,p)
mm = 111222
```