

During the midterm exam you must solve 2 problems from

<https://imimsociety.net/en/14-cryptography>



The list of Course Work topics are presented in my Google drive:

<https://docs.google.com/document/d/1IFjPGzigO7EiMPFtwiBI28gfBrv8hnBo/edit?usp=sharing&oid=111502255533491874828&rtpof=true&sd=true>

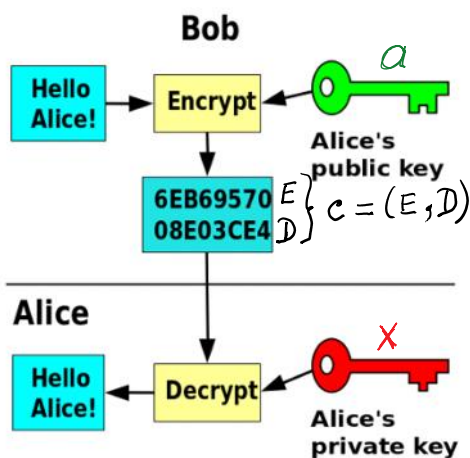
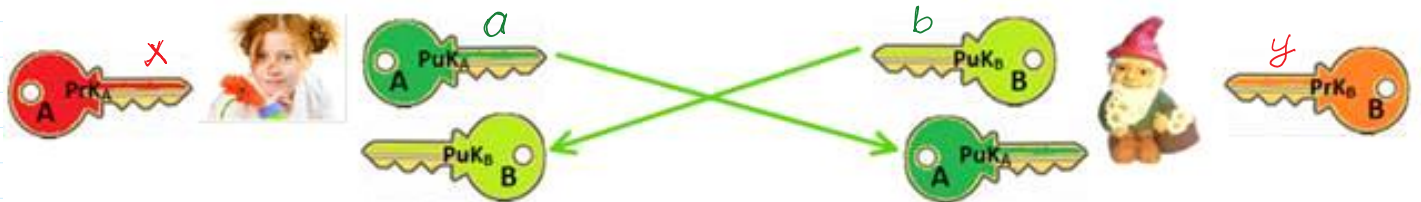
Please choose a topic and label it by S.Name, where S is your surmane.

## 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  $m \bmod p = m$ .



$$PrK_A = x = randi(p-1)$$

$$PuK_A = a = g^x \bmod p$$

A:  $PuK_A = a$  → B: is able to encrypt

$A$ :  $\text{PuK}_A = a \longrightarrow B$ : is able to encrypt  $m$  to  $A$ :  $m < p$

$B$ :  $i \leftarrow \text{randi}(\mathcal{I}_p^*)$

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

$A$ : is able to decrypt  $C = (E, D)$  using her  $\text{PrK}_A = x$ .

$$\begin{aligned} (-x) \bmod (p-1) &= (0-x) \bmod (p-1) = \\ &= (p-1-x) \bmod (p-1) \end{aligned}$$

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

$D^{-x} \bmod p$  computation using Fermat theorem:  
If  $p$  is prime, then for any integer  $a$  holds  $a^{p-1} = 1 \bmod p$ .

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

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

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

Correctness

$$\text{Enc}(\text{PuK}_A = a, i, m) = c = (E, D) = (E = m \cdot a^i \bmod p; D = g^i \bmod p)$$

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

$$\begin{aligned} &= m \cdot \underbrace{(g^x)^i}_{a^i} \cdot g^{-ix} = m \cdot g^{xi} \cdot g^{-ix} = m \cdot g^{xi - ix} \bmod p = m \cdot g^0 \bmod p = \\ &= m \cdot 1 \bmod p = m \bmod p = m = 111222 \end{aligned}$$

Since  $m < p$

If  $m > p \rightarrow m \bmod p \neq m$ ;  $27 \bmod 5 = 2 \neq 27$ .

If  $m < p \rightarrow m \bmod p = m$ ;  $19 \bmod 31 = 19$ .

Decryption is correct if  $m < p$ .

ASCII

$$\frac{2048}{8} =$$

= 256 char.

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

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

1-st encryption:

$$i_1 \leftarrow \text{rand}_i(\mathcal{Z}_p^*)$$

$$E_1 = m \cdot a^{i_1} \bmod p$$

$$D_1 = g^{i_1} \bmod p$$

$$C_1 = (E_1, D_1)$$

$i_1 \neq i_2$

$C_1 \neq C_2$

2-nd encryption

$$i_2 \leftarrow \text{rand}_i(\mathcal{Z}_p^*)$$

$$E_2 = m \cdot a^{i_2} \bmod p$$

$$D_2 = g^{i_2} \bmod p$$

$$C_2 = (E_2, D_2)$$

### 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

### Authenticated Key Agreement Protocol using ElGamal Encryption and Signature for a large files encryption using symmetric encryption method

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-256.

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-256.

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

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

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

$PuK_B = b$ .

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

$PuK_A = a$ .

1)  $k \leftarrow \text{rand}_i(2^{256})$

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

- 1.1. Verify if  $PuK_A$  is valid
- 1.2. Verify if  $c$  is valid

$$I_0 \quad \text{Enc}(\text{PrK}_B = b, l_k, k) = C = (E, D)$$

2) M - large file to be encrypted

$$E_k(M) = \text{AES}_k(M) = G$$

3) Signs ciphertext G

$$3.1) \quad h = H(G)$$

$$3.2) \quad \text{Sign}(\text{PrK}_A = x, h) = \tilde{\sigma} = (r, s)$$

$$\left. \begin{array}{l} C, G \\ \tilde{\sigma}, \text{PrK}_A \\ \text{Cert}_A \end{array} \right\}$$

1.1. verify if  $\text{PrK}_A$  is valid

1.2. Verify if  $\tilde{\sigma}$  is valid

$$h' = H(G)$$

$$\text{Ver}(\text{PrK}_A, \tilde{\sigma}, h') = \text{True}$$

$$2. \text{Dec}(\text{PrK}_B, c) = k$$

$$3. D_k(G) = \text{AES}_k(G) = M.$$

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

#### AKAP: DH-KAP & Asym. Enc

### Homomorphic property of ElGamal encryption

Let we have 2 messages  $m_1, m_2$  to be encrypted

$$i_1 \leftarrow \text{randi}(\mathbb{Z}_p^*)$$

$$E_1 = m_1 \cdot a^{i_1} \bmod p$$

$$D_1 = g^{i_1} \bmod p$$

$$i_2 \leftarrow \text{randi}(\mathbb{Z}_p^*)$$

$$E_2 = m_2 \cdot a^{i_2} \bmod p$$

$$D_2 = g^{i_2} \bmod p$$

$$\text{Enc}(a, (i_1 + i_2) \bmod (p-1), m_1 \cdot m_2 \bmod p) = C_{12} = (E_{12}, D_{12})$$

$$E_{12} = m_1 \cdot m_2 \cdot a^{i_1 + i_2 \bmod (p-1)} \bmod p = \underbrace{(m_1 \cdot a^{i_1} \bmod p)}_{E_1} \cdot \underbrace{(m_2 \cdot a^{i_2} \bmod p)}_{E_2} \bmod p$$

$$E_{12} = E_1 \cdot E_2 \bmod p$$

$$D_{12} = g^{i_1 + i_2} \bmod p = \underbrace{(g^{i_1} \bmod p)}_{D_1} \cdot \underbrace{(g^{i_2} \bmod p)}_{D_2} \bmod p$$

$$D_{12} = g^{e_1 + e_2} \bmod p = \underbrace{(g^{e_1} \bmod p)}_{D_1} \cdot \underbrace{(g^{e_2} \bmod p)}_{D_2} \bmod p$$

$$D_{12} = D_1 \cdot D_2 \bmod p$$

$$\begin{aligned} \text{Enc}(a, (i_1 + i_2) \bmod (p-1), m_1 \cdot m_2 \bmod p) &= c_{12} = \\ &= (E_{12}, D_{12}) = \\ &= (E_1 \cdot E_2 \bmod p, D_1 \cdot D_2 \bmod p) = c_1 \cdot c_2 \end{aligned}$$

Multiplicative isomorphism

Encryption function of product  $m_1 \cdot m_2$  of two plaintexts  $m_1$  and  $m_2$  maps to ciphertext  $c_1 \cdot c_2 = c$  of two ciphertexts  $c_1$  and  $c_2$ , when  $c_1 = \text{Enc}(a, i_1, m_1)$  and  $c_2 = \text{Enc}(a, i_2, m_2)$ .

Till this place

Multiplicatively additive isomorphism

$$\text{Enc}(m_1 + m_2) = c_1 \cdot c_2 \quad \Leftarrow \text{Pascal Paillier encryption.}$$

Application in eVoting and Blockchain systems.

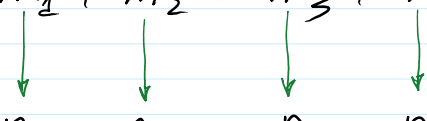
One special case of ElGamal encryption

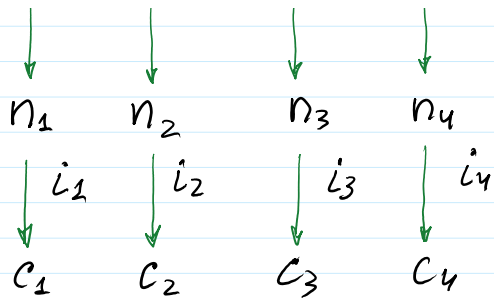
is instead of  $m_1, m_2$  encryption to encrypt messages  $n_1 = g^{m_1}, n_2 = g^{m_2}; n_3 = g^{m_3}, n_4 = g^{m_4};$

$$\text{Enc}(a, i_1 + i_2, n_1 \cdot n_2) = \text{Enc}(a, i_1, n_1) \cdot \text{Enc}(a, i_2, n_2)$$

$$\begin{aligned} E_{12} &= E_1 \cdot E_2 \bmod p = n_1 a^{i_1} \bmod p \cdot n_2 a^{i_2} \bmod p = \\ &= g^{m_1} a^{i_1} \bmod p \cdot g^{m_2} a^{i_2} \bmod p = \\ &= g^{m_1 + m_2} \cdot a^{i_1 + i_2} \bmod p. \end{aligned}$$

$$\text{Let } m_1 + m_2 = m_3 + m_4$$





If  $m_1 + m_2 = m_3 + m_4 \pmod{p-1} \Rightarrow c_1 \cdot c_2 = c_3 \cdot c_4$  ;

**Homomorphic encryption: cloud computation with encrypted data.**

Paillier encryption scheme is additively-multiplicative homomorphic and has a potentially nice applications in blockchain, public procurement, auctions, gamblings and etc.

$$\text{Enc}(\text{Puk}, m_1 + m_2) = c_1 \cdot c_2.$$