

During the exam you must solve the following 5 problems.



And answer to 2 questions from the declared list.

DEF-Homomorphism. EIG-Enc

Compatibility relations of modular arithmetic:

$$(a + b) \bmod p = (a \bmod p + b \bmod p) \bmod p.$$

$$(a * b) \bmod p = ((a \bmod p) * (b \bmod p)) \bmod p.$$

$$a^z \bmod p = (a \bmod p)^z \bmod p.$$

Fermat little theorem: If p is prime, then for any integer a holds $a^p = a \bmod p$.

1. We may assume that a is in the range $0 \leq a \leq p - 1$.

This is a simple consequence of the laws of [modular arithmetic](#); we are simply saying that we may first reduce a modulo p since

1. It suffices to prove that for a in the range $1 \leq a \leq p$

$$a^p = a \bmod p \quad | \quad a^{-1} \bmod p$$

$$a^p \cdot a^{-1} = a \cdot a^{-1} \bmod p$$

$$0 < a < p. \quad a^{p-1} = 1 \bmod p \quad a^0 = a^{p-1} = 1 \bmod p.$$

Indeed, if the previous assertion holds for such a , multiplying both sides by a yields the original form of the theorem.

Computation of exponents mod $(p-1)$:

$$s = xh + r \rightarrow g^s \bmod (p-1) \bmod p$$

$$g^{p-1} = 1 \bmod p \quad \& \quad g^0 = 1 \bmod p \quad \rightarrow \quad 0 \equiv p-1$$

$$a^z \bmod p = a^{z \bmod (p-1)} \bmod p.$$

```
>> p=genstrongprime(28) // strong prime number generation
p = 215393099           // p - is strong prime iff p=2*q+1, when q is also prime: q=(p-1)/2
>> q=(p-1)/2
q = 107696549

// finding a generator g: g - is a generator in  $Z_p^* = \{1, 2, 3, \dots, p-1\} \bmod p$ 
// iff  $g^q \neq 1$  &  $g^2 \neq 1$ 

>> g=2
g = 2
>> mod_exp(g,q,p)
ans = 215393098
>> mod_exp(g,2,p)
ans = 4                // g=2 is a generator
```

$$s = (xh + r) \bmod (p-1); \quad g^s \bmod p = g^{s \bmod (p-1)} \bmod p.$$

```
>> x=int64(randi(p-1))
x = 169506978
>> a=mod_exp(g,x,p)
a = 99428799
>> con=concat('hello bob',a)
con = hello bob99428799
>> h=hd28(con)
h = 252819993
>> r=int64(randi(p-1))
r = 96513791

>> s=int64(x*h+r)
s = 42854753087924945
>> smodpm1=mod(s,p-1)
smodpm1 = 150400265
>> g_s=mod_exp(g,s,p)
g_s = 162111969
>> g_smodpm1=mod_exp(g,smodpm1,p)
g_smodpm1 = 162111969
```

DEF homomorphism

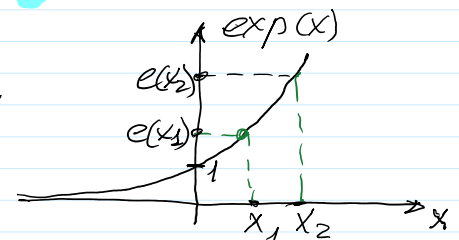
$$\exp(x) = e^x; \quad \exp: \mathbb{R} \rightarrow \mathbb{R}^+; \quad e = 2,71828\dots$$

$$x \in \mathbb{R} \rightarrow e^x \in \mathbb{R}^+.$$

$$\exp(x_1 + x_2) = e^{(x_1 + x_2)} = e^{x_1} \cdot e^{x_2} = \exp(x_1) \cdot \exp(x_2)$$

Additively - multiplicative homomorphism.

Since it is 1-to-1, then it is isomorphism.



$$\text{DEF } (x) = g^x \bmod p; \quad p \text{ - (strong) prime}$$

g - generator in $\mathbb{Z}_p^* = \{1, 2, 3, \dots, p-1\}$

$x \in \mathbb{Z}_{p-1} = \{0, 1, 2, 3, \dots, p-2\}; + \text{mod}(p-1), * \text{mod}(p-1), - \text{mod}(p-1)$
 $|\mathbb{Z}_{p-1}| = p-1$ $/ \text{mod}(p-1)$

DEF(x) = $a \in \mathbb{Z}_p^* = \{1, 2, 3, \dots, p-1\}; * \text{mod } p, / \text{mod } p$.
 $|\mathbb{Z}_p^*| = p-1 = |\mathbb{Z}_{p-1}|$

DEF realizes the following mapping DEF: $\mathbb{Z}_{p-1} \rightarrow \mathbb{Z}_p^*$; DEF is defined by Public Parameters: **PP** = (p, g)

DEF_{g,p}(x) = $g^x \text{mod } p = a$.

p = 11;

g = 7.

	$x \in \mathbb{Z}_{10}$	$a \in \mathbb{Z}_{11}^*$
$7^0 = 1 \text{ mod } 11$	0	1
$7^1 = 7 \text{ mod } 11$	1	2
$7^2 = 5 \text{ mod } 11$	2	3
$7^3 = 2 \text{ mod } 11$	3	4
$7^4 = 3 \text{ mod } 11$	4	5
$7^5 = 10 \text{ mod } 11$	5	6
$7^6 = 4 \text{ mod } 11$	6	7
$7^7 = 6 \text{ mod } 11$	7	8
$7^8 = 9 \text{ mod } 11$	8	9
$7^9 = 8 \text{ mod } 11$	9	10

$$\text{DEF}(x_1 + x_2) = g^{(x_1 + x_2) \text{ mod } (p-1)} \text{ mod } p = g^{x_1} \cdot g^{x_2} \text{ mod } p =$$

$$(a * b) \text{ mod } p = ((a \text{ mod } p) * (b \text{ mod } p)) \text{ mod } p.$$

$$= ((g^{x_1} \text{ mod } p) \cdot (g^{x_2} \text{ mod } p)) \text{ mod } p = \text{DEF}(x_1) \cdot \text{DEF}(x_2)$$

Additively - multiplicative homomorphism.

Since it is 1-to-1, then it is isomorphism.

ElGamal Encryption-Decryption

Public Parameters generation **PP** = (p, g).

Asymmetric Signing - Verification

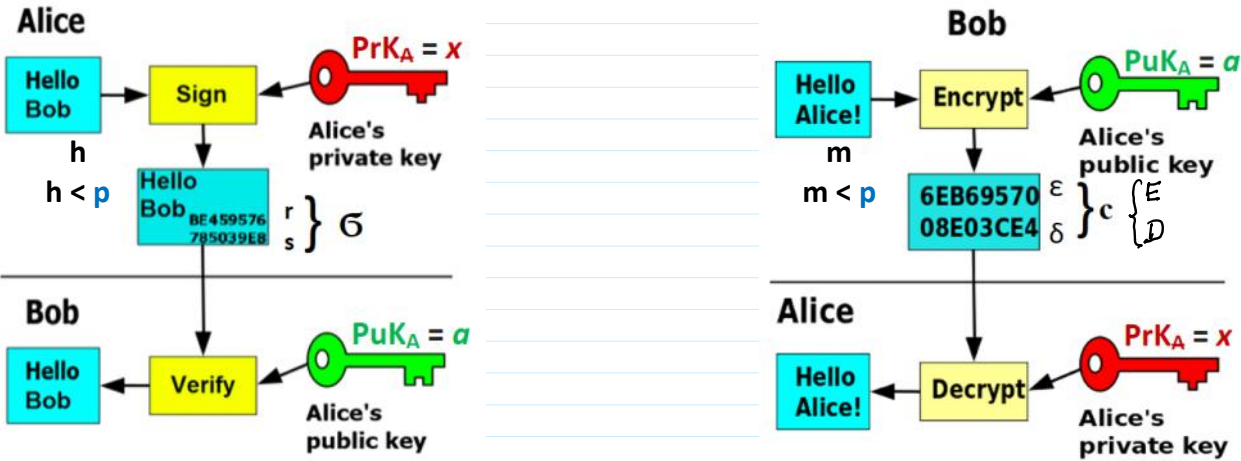
Sign(**PrK_A**, h) = σ = (r, s)

V=Ver(**PuK_A**, h, σ), V ∈ {True, False} ≡ {1, 0}

Asymmetric Encryption - Decryption

c=Enc(**PuK_A**, m)

m=Dec(**PrK_A**, c)



ElGamal Cryptosystem

1. Public Parameters generation $PP = (p, g)$.

Generate strong prime number p : `>> p=genstrongprime(28)` % strong prime of 28 bit length

Find a generator g in $Z_p^* = \{1, 2, 3, \dots, p-1\}$ using condition.

Strong prime $p=2q+1$, where q is prime, then g is a generator of Z_p^* iff

$g^q \neq 1 \pmod p$ and $g^2 \neq 1 \pmod p$.

`>> 2^28-1`
`ans = 2.6844e+08`
`>> int64(2^28-1)`
`ans = 268435455`

Declare Public Parameters to the network $PP = (p, g)$;

$p = 268435019$; $g = 2$;

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

$2^{28}-1 = 268,435,455$

$PrK = x \leftarrow \text{randi}(Z_p^*) \implies PuK = a = g^x \pmod p$

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 $m \pmod p = m$.

$$\mathcal{I}_p^* = \{1, 2, 3, \dots, p-1\}; * \pmod p$$

A: $PuK_A = a \longrightarrow$ B: is able to encrypt m to c : $m < p$

B: $i \leftarrow \text{randi}(\mathcal{I}_{p-1})$
 $E = m \cdot a^i \pmod p$
 $D = g^i \pmod p$
 $c = (E, D) \longrightarrow$ A: is able to decrypt $c = (E, D)$ using her $PrK_A = x$.

$(-x) \pmod{(p-1)} = (0-x) \pmod{(p-1)} = (p-1-x) \pmod{(p-1)}$
 1. $D^{-x \pmod{(p-1)}} \pmod p$ `> x=123`
`x = 123`
 2. $E \cdot D^{-x \pmod p} = m$ `>> pp=127`
`pp = 127`

$$-(p^{-1-x}) \text{ mod } (p-1)$$

$$(p-1) \text{ mod } (p-1) = 0 \text{ since}$$

$$(-x) \text{ mod } (p-1) = (p-1-x)$$

$$D^{-x} \text{ mod } (p-1) = D^{p-1-x} \text{ mod } (p-1)$$

$$\gg D_{-mx} = \text{mod_exp}(D, p-1-x, p)$$

$$\frac{p-1}{p-1} \mid \frac{p-1}{1}$$

```
pp = 127
>> isprime(pp)
ans = 1
>> mx = mod(-x, pp-1)
mx = 3
>> mod(x+mx, pp-1)
ans = 0
```

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

$D^{-x} \text{ mod } p$ computation using Fermat theorem:

If p is prime, then for any integer a in \mathbb{Z}_p^* holds $a^{p-1} = 1 \text{ mod } p$.

$$D^{p-1} = 1 \text{ mod } p \quad / \cdot D^{-x \text{ mod } (p-1)} \text{ mod } p$$

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

$$D^{-x} \text{ mod } p = D^{p-1-x} \text{ mod } p$$

Correctness

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

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

$$= m \cdot \underbrace{(g^x)^i}_a \cdot g^{-ix} = m \cdot g^{xi} \cdot g^{-ix} = m \cdot g^{xi-ix} \text{ mod } p = m \cdot g^0 \text{ mod } p =$$

$$= m \cdot 1 \text{ mod } p = m \text{ mod } p = m = 111222$$

Since $m < p$

$$\begin{array}{r} 27 \overline{) 15} \\ 25 \quad \overline{) 5} \\ \hline 2 \end{array}$$

If $m > p \rightarrow m \text{ mod } p \neq m$; $27 \text{ mod } 5 = 2 \neq 27$.

If $m < p \rightarrow m \text{ mod } p = m$; $19 \text{ mod } 31 = 19$.

ASCII: 8 bits per char.
 $\frac{2048}{8} = 256 \text{ char.}$

Decryption is correct if $m < p$.

Homomorphic Encryption

Let m_1 and m_2 have to be encrypted.

$$\text{Enc}(PK_A = a, i_1, m_1) = c_1 = (E_1, D_1) = (E_1 = m_1 a^{i_1} \text{ mod } p, D_1 = g^{i_1} \text{ mod } p)$$

$$\text{Enc}(PK_A = a, i_2, m_2) = c_2 = (E_2, D_2) = (E_2 = m_2 a^{i_2} \text{ mod } p, D_2 = g^{i_2} \text{ mod } p)$$

$$c_{12} = c_1 \cdot c_2 = (E_1, D_1) \cdot (E_2, D_2) = (E_1 \cdot E_2, D_1 \cdot D_2) = (E_{12}, D_{12}) \text{ mod } p$$

$$C_{12} = C_1 \cdot C_2 = (E_1, D_1) \cdot (E_2, D_2) = (E_1 \cdot E_2, D_1 \cdot D_2) = (E_{12}, D_{12}) \pmod{p}$$

$$E_{12} = m_1 \cdot m_2 \cdot a^{i_1} a^{i_2} \pmod{p} = m_1 \cdot m_2 a^{(i_1+i_2)} \pmod{(p-1)} \pmod{p} =$$

$$= m_1 \cdot m_2 a^{i_{12}} \pmod{p} = m_{12} a^{i_{12}} \pmod{(p-1)} \pmod{p}$$

$$D_{12} = D_1 \cdot D_2 = g^{i_1} \cdot g^{i_2} \pmod{p} = g^{i_1+i_2} \pmod{p} = g^{i_{12}} \pmod{p}$$

$$\begin{cases} m_{12} = m_1 \cdot m_2 \pmod{p} \\ i_{12} = (i_1 + i_2) \pmod{(p-1)} \end{cases}$$

$$\text{Enc}(\text{PK}_A = a, i_{12}, m_1 \cdot m_2) = C_{12} = C_1 \cdot C_2 = (E_1 \cdot E_2, D_1 \cdot D_2) = (E_{12}, D_{12})$$

Multiplicatively homomorphic encryption.

We need an additively-multiplicative encryption.

$$n_1 = g^{m_1} \pmod{p} \rightarrow \text{Enc}(\text{PK}_A = a, i_1, n_1) = (E_1, D_1) =$$

$$= (E_1 = n_1 \cdot a^{i_1} \pmod{p}, D_1 = g^{i_1} \pmod{p}) = C_1$$

$$n_2 = g^{m_2} \pmod{p} \rightarrow \text{Enc}(\text{PK}_A = a, i_2, n_2) = (E_2, D_2) =$$

$$= (E_2 = n_2 \cdot a^{i_2} \pmod{p}, D_2 = g^{i_2} \pmod{p}) = C_2$$

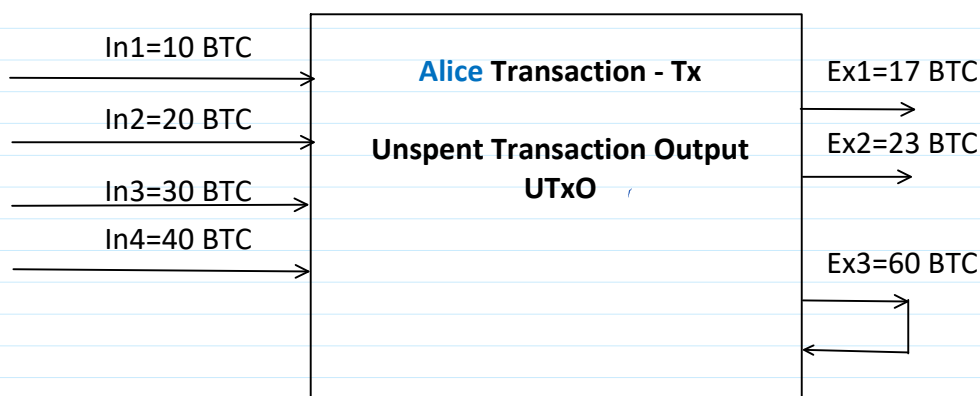
By transforming m_1, m_2 to n_1, n_2 we obtain additively-multiplicative homomorphic encryption with respect to m_1, m_2 , by encryption n_1, n_2 in a standard ElGamal encryption.

Since

$$n_1 \cdot n_2 \pmod{p} = g^{m_1} \cdot g^{m_2} \pmod{p} = g^{m_1+m_2} \pmod{(p-1)} \pmod{p},$$

Then

$$\text{Enc}(a, i_1+i_2 \pmod{(p-1)}, n_1 \cdot n_2 \pmod{p}) = \text{Enc}(a, i_1+i_2 \pmod{(p-1)}, g^{m_1+m_2} \pmod{(p-1)} \pmod{p}) = C_{12} = C_1 \cdot C_2 \pmod{p}.$$



$$\text{Balance verification: } \text{In1} + \text{In2} + \text{In3} + \text{In4} = \text{Ex1} + \text{Ex2} + \text{Ex3} =$$

$$10 + 20 + 30 + 40 = 17 + 23 + 60 = 100$$

If Incomes are encrypted

$$In_1 = g^{in_1} \text{ mod } p \rightarrow \text{Enc}(a, i_1, n_1) = cIn_1$$

$$In_2 = g^{in_2} \text{ mod } p \rightarrow \text{Enc}(a, i_2, n_1) = cIn_2$$

$$In_3 = g^{in_3} \text{ mod } p \rightarrow \text{Enc}(a, i_3, n_1) = cIn_3$$

$$In_4 = g^{in_4} \text{ mod } p \rightarrow \text{Enc}(a, i_4, n_1) = cIn_4$$

If Expenses are encrypted

$$Ex_1 = g^{ex_1} \text{ mod } p \rightarrow \text{Enc}(a, e_1, n_1) = cEx_1$$

$$Ex_2 = g^{ex_2} \text{ mod } p \rightarrow \text{Enc}(a, e_2, n_1) = cEx_2$$

$$Ex_3 = g^{ex_3} \text{ mod } p \rightarrow \text{Enc}(a, e_3, n_1) = cEx_3$$

If balance is is valid, then:

$$cIn_1 * cIn_2 * cIn_3 * cIn_4 = cEx_1 * cEx_2 * cEx_3$$

