027_018 Homom-Enc-ElGam-Ex

$$p=11 \;;\; g=2 \;;\quad \left.\begin{array}{l} m_1=1 \;;\; g^{m_1} \bmod p \\ m_2=2 \;;\; g^{m_2} \bmod p \end{array}\right\} \begin{array}{l} \to c_1 \to Dec(x, g^{m_1}) = m_1 \\ \to c_2 \to Dec\; x, g^{m_2}) = m_2 \end{array}$$

$$(m_1+m_2) \underset{\bmod 10}{\bmod (p-1)} = (1+2) \underset{\bmod 10}{\bmod (p-1)} = 3$$

$$\left.\begin{array}{l} m_1 = 4 \\ m_2 = 9 \end{array}\right\} (m_1+m_2) \bmod 10 = (4+9) \bmod 10 = 13 \bmod 10 = 3$$

Prevention: $m_1 + m_2 < (p-1)/2$

$\mathcal{Z}_{p-1}$	0	1	2	3	4	5	6	7	8	9
						± 5	-4	-3	-2	-1

```
>> p=int64(268435019)
p = 268435019
>> g=2;
>>
>> x=int64(randi(p-1))
x = 141076898
>>  a=mod_exp(g,x,p)
a = 116336486
```

```
>> m1=2000;
>> m2=3000;
>> m12=mod(m1+m2,p-1)
m12 = 5000
>> n1=mod_exp(g,m1,p)
n1 = 28125784
>> n2=mod_exp(g,m2,p)
n2 = 222979214
```

m_1 : $i_1 \leftarrow randi(\mathcal{Z}_{p-1})$

$$\left.\begin{array}{l} \mathcal{E}_1 = m_1 * a^{i_1} \bmod p \\ \delta_1 = g^{i_1} \bmod p \end{array}\right\}$$

```
>> i1=int64(randi(p-1))
i1 = 256575903
>> a_i1=mod_exp(a,i1,p)
a_i1 = 177744290
>> E1=mod(n1*a_i1,p)
E1 = 78907012
>> D1=mod_exp(g,i1,p)
D1 = 71219017
```

m_2 : $i_2 \leftarrow randi(\mathcal{Z}_{p-1})$

$$\left.\begin{array}{l} \mathcal{E}_2 = m_2 * a^{i_2} \bmod p \\ \delta_2 = g^{i_2} \bmod p \end{array}\right\}$$

```
>> i2=int64(randi(p-1))
i2 = 148753825
>> a_i2=mod_exp(a,i2,p)
a_i2 = 206019372
>> E2=mod(n2*a_i2,p)
E2 = 144070332
>> D2=mod_exp(g,i2,p)
D2 = 20873398
```

Computations in Cloud data base

```
>> E12=mod(E1*E2,p)
E12 = 248852506
>> D12=mod(D1*D2,p)
D12 = 220753507

c12=(E12, D12)
```

Alice decrypts c12=(E12, D12)

Verification

```
>> mx=mod(-x,p-1)
mx = 127358120
>> mod(mx+x,p-1)
ans = 0
>> D12_mx=mod_exp(D12,mx,p)
D12_mx = 21824811
>> nnn12=mod(E12*D12_mx,p)
nnn12 = 143845522
```

```
>> nn12=mod(n1*n2,p)
nn12 = 143845522
>> n12=mod_exp(g,m12,p)
n12 = 143845522
```

```
% Finds discrete logarithm value corresponding to exponent value i
% by total scan of i from start by step until fin
% p - is a strong prime (Public Parameter)
% g - is a generator (Public Parameter)
% def - is a discrete exponent function value computed by mod_exp(g,i,p)
%       where dl=i is a searchable value of exponent
%
function dl = dlog(p, g, def, start, step, fin)
  dl=0;
  i=start;
  while i<fin
      ee=mod_exp(g,i,p);
      if ee==def
        dl=i;
        return;
      endif
      i+=step;
  endwhile
  disp('Exponent is not found!');
 end
```

```
>> def=nnn12
def = 143845522
>> start=0
start = 0
>> step=100
step = 100
>> fin=9900
fin = 9900
>>
>> dl = dlog(p, g, def, start, step, fin)
dl = 5000
```