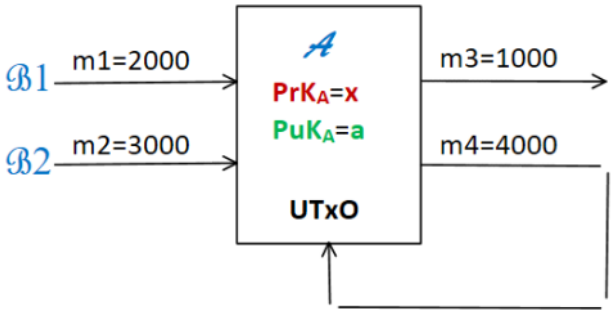
Application in UTxO blockchain to provide confidentiality and verifiability of transferred money amounts.

Public Parameters PP = (p, g); p=268435019; g=2;



How to provide anonymity of transaction amounts and to verify the **balance**: **m1+m2 = m3+m4** ?

n1= g^{m1} mod **p** **n3**= g^{m3} mod **p**

n2= g^{m2} mod **p** **n4**= g^{m4} mod **p**

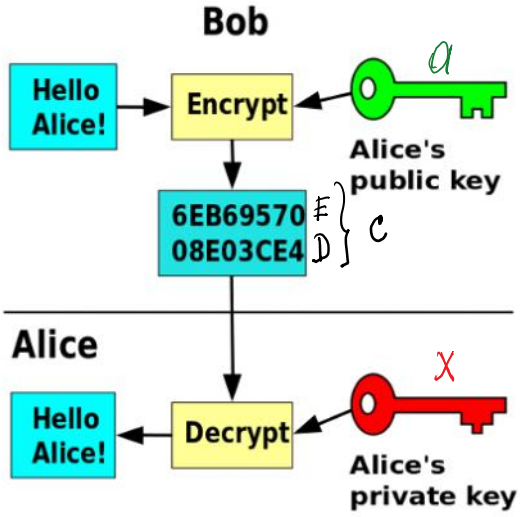
If **m1+m2 = m3+m4**,
Then **n1*n2 = n3*n4**.

$n_1 \cdot n_2 = g^{m_1} g^{m_2} \bmod p = g^{m_1+m_2} \bmod p$

ElGamal Encryption and ZKP based on **Schnorr Identification** are used.

Public Parameters PP = (p, g); p=268435019; g=2;

ElPublic and Private keys generation
PrK_A = *x* = randi(***p***-1).
PuK_A = *a* = g^x mod ***p***.



ElGamal Encryption - Decryption
c=Enc(PuK_A, m)=(E,D)
m=Dec(PrK_A, c)

ℬ1: Enc(**a**, n1) = c1
i1 = randi(**p**-1)
E1= n1*a^{i1} mod **p**
D1= g^{i1} mod **p**
c1=(E1,D1)

Enc(**a**, i1) = ci1
j1 = randi(**p**-1)
Ei1= i1*a^{j1} mod **p**
Di1= g^{j1} mod **p**
ci1=(Ei1,Di1)

ℬ2: Enc(**a**, n2) = c2
i2 = randi(**p**-1)
E2= n2*a^{i2} mod **p**
D2= g^{i2} mod **p**
c2=(E2,D2)

Enc(**a**, i2) = ci2
j2 = randi(**p**-1)
Ei2= i2*a^{j2} mod **p**
Di2= g^{j2} mod **p**
ci2=(Ei2,Di2)

ℬ1: c1=(E1,D1), ci1=(Ei1,Di1) → 𝒜: Dec(**x**, c1) = n1 & verifies if n1=g^{m1} mod **p**

Dec(**x**, ci1) = i1

$\mathcal{B}$2: c2=(E2,D2), ci2=(Ei2,Di2) → $\mathcal{A}$: Dec(**x**, c2) = n2 & verifies if n2=g^{m2} mod **p**
Dec(**x**, ci2) = i2

$\mathcal{A}$: Computes: i12=i1+i2 mod (**p**-1)
Gemerates: i3 = randi(**p**-1)
Computes: i4=i12-i3 mod (**p**-1)
Computes: i34=i3+i4 mod (**p**-1)
Verifies if: i12=i34=***i***

n3= g^{m3} mod **p**
n4= g^{m4} mod **p**

To prove the honesty of transaction $\mathcal{A}$ encrypts number n_3, n_4 with her $PuK_A = a$.

c1=(E1,D1)

c2=(E2,D2)

E3= n3*a^{i3} mod **p**
D3= g^{i3} mod **p**
c3=(E3,D3)

E4= n4*a^{i4} mod **p**
D4= g^{i4} mod **p**
c4=(E4,D4)

Nodes has an information of transaction
c_1, c_2 of input ciphertexts & c_3, c_4 of output ciphertexts.
If $m_1 + m_2 = m_{12}$ ⟺ $m_3 + m_4 = m_{34}$ $(\text{mod } p-1)$
Then $n_1 \cdot n_2 = n_{12}$ ⟺ $n_3 \circ n_4 = n_{34}$ $(\text{mod } p)$

$\mathcal{A}$: Enc(**e**, n3) = ce3=(E3,D3) → $\mathcal{E}$: Dec(**z**, ce3) = n3 & verifies if n1=g^{m1} mod **p**
E3= n3*e^{i3} mod **p**
D3= g^{i3} mod **p**

Nodes knows c_3, ce_3 : how to know that c_3 & ce_3 encrypts the same plaintext - number $n_3 = g^{m_3} \text{ mod } p$?

$c_3 \neq ce_3$

$c_3 = (E3, D3)$
$ce3 = (Ee3, De3)$
$\Rightarrow \left(\frac{E3}{Ee3}, \frac{D3}{De3}\right)$

$n_3 a^{i_3} \text{ mod } p$ $\quad a^{i_3}$ $\quad i_3$

$$\frac{n_3\, a^{i_3} \bmod p}{n_3\, e^{i_3} \bmod p} = \frac{a^{i_3}}{e^{i_3}} \bmod p = \left(\frac{a}{e}\right)^{i_3} \bmod p$$

```
>> p=int64(268435019);
>> g=2;
>> x=int64(randi(p-1))
x = 88637397
>> a=mod_exp(g,x,p)
a = 71787401

>> m1=2000;
>> m2=3000;
>> m3=1000;
>> m4=4000;
>>
>> n1=mod_exp(g,m1,p)
n1 = 28125784
>> n2=mod_exp(g,m2,p)
n2 = 222979214
>> n3=mod_exp(g,m3,p)
n3 = 260099963
>> n4=mod_exp(g,m4,p)
n4 = 246637967
```

B1:

```
>>  i1=int64(randi(p-1))
i1 = 35336707
>> a_i1=mod_exp(a,i1,p)
a_i1 = 215116696
>> E1=mod(n1*a_i1,p)
E1 = 113883800
>> D1=mod_exp(g,i1,p)
D1 = 175198005

>> j1=int64(randi(p-1))
j1 = 176083525
>> a_j1=mod_exp(a,j1,p)
a_j1 = 186263712
>> Ej1=mod(i1*a_j1,p)
Ej1 = 80302084
>>  Dj1=mod_exp(g,j1,p)
Dj1 = 204302127
```

B2:

```
>>  i2=int64(randi(p-1))
i2 = 56554409
>> a_i2=mod_exp(a,i2,p)
a_i2 = 141556987
>> E2=mod(n2*a_i2,p)
E2 = 55189582
>> D2=mod_exp(g,i2,p)
D2 = 215063534

>> j2=int64(randi(p-1))
j2 = 255282413
>> a_j2=mod_exp(a,j2,p)
a_j2 = 114387504
>> Ej2=mod(i2*a_j2,p)
Ej2 = 157516916
>> Dj2=mod_exp(g,j2,p)
Dj2 = 170451614
```

$$Enc(a, n_1) = C_1 = (E_1, D_1)$$
$$= (n_1 \cdot a^{i_1} \bmod p,\ g^{i_1} \bmod p)$$
$$Dec(x, e_1) = E_1 \cdot D_1^{(-x) \bmod p-1} \bmod p$$

$$>> mx = addinv(x, p-1)$$?

$$>> mx = mod(p-1-x, p-1)$$

```
>> mx=addinv(x,p-1)
mx = 179797621
>> mod(x+mx,p-1)
ans = 0
>> mx=mod(p-1-x,p-1)
mx = 179797621
>> mod(x+mx,p-1)
ans = 0
```

Till this place

Zero Knowledge Proof (ZKP) of equivalence of 2 ciphertexts *c*1, *c*2 corresponding to the same plaintext *m* obtained by encryption with different Puks

$\mathcal{A}$: Enc(**e**, n3) = c3=(E3,D3)
E3= n3*$\mathbf{e}^{i3}$ mod **p**
D3= $\mathbf{g}^{i3}$ mod **p**
Enc(**e**, i3) = ci3
j3 <-- randi(**p**-1)
Ei3= i3*$\mathbf{e}^{j3}$ mod **p**
Di3= $\mathbf{g}^{j3}$ mod **p**

c3=(E3,D3)
ci3=(Ei3,Di3)

→ $\mathcal{E}$: PrK=z; PuK=e.

c4=(E4,D4)
ci4=(Ei4,Di4)

Enc(**e**, n4) = c4=(E4,D4)
E4= n4*$\mathbf{a}^{i4}$ mod **p**
D4= $\mathbf{g}^{i4}$ mod **p**
Enc(**e**, i4) = ci4
j4 <-- randi(**p**-1)
Ei4= i4*$\mathbf{e}^{j4}$ mod **p**
Di4= $\mathbf{g}^{j4}$ mod **p**

$\mathcal{E}$: Dec(**x**, c3) = n3 & verifies if n3=$\mathbf{g}^{m3}$ mod **p**
Dec(**x**, ci3) = i3

$\mathcal{E}$: Dec(**x**, c4) = n4 & verifies if n4=$\mathbf{g}^{m4}$ mod **p**
Dec(**x**, ci4) = i4

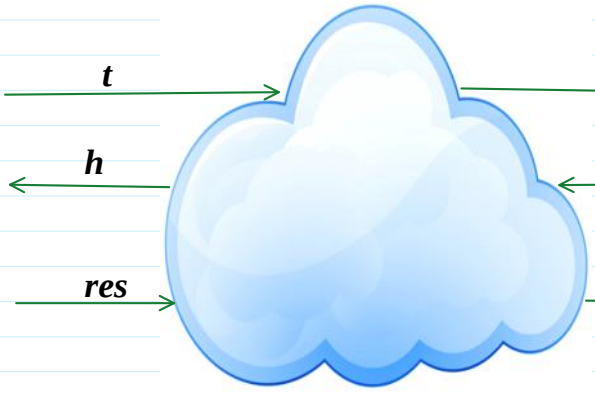
```
>> p=268435019;
>> g=2;
>> x=int64(randi(p-1))
x = 88637397
>> a=mod_exp(g,x,p)
a = 71787401

>> m1=1000;
>> m2=3000;
>> m3=3000;
>> m4=4000;
>>
>> n1=mod_exp(g,m1,p)
n1 = 28125784
>> n2=mod_exp(g,m2,p)
n2 = 222979214
>> n3=mod_exp(g,m3,p)
n3 = 260099963
>> n4=mod_exp(g,m4,p)
n4 = 246637967
```

Net:
c1, c2, c3, c4. Verification: c12=c1*c2 & c34=c3*c4 --> c12==c34

Schnorr Identification: Zero Knowledge Proof - ZKP

Schnorr Id Scenario: **Alice** wants to prove **Bank** that she knows her Private Key - $PrK_A = x$ which corresponds to her Public Key - $PuK_A = a = g^x \bmod p$ not revealing $PrK_A = x$.

𝒜: ZKP of knowledge x:
$PrK_A = x =$ randi(p-1)
$PuK_A = a = g^x \bmod p$
1.Computes commitment ***t*** for ***i***:
i=randi(***p***-1)
$t = g^i \bmod p$
3.Computes response ***res***:
$res = i + xh \bmod p-1$



ℬ: $PuK_A = a = g^x \bmod p$

2.Generates challenge ***h***:
h=randi(***p***-2)

Verifies:
$g^{res} = ta^h \bmod p$

Correctness:
$g^{res} \bmod p = g^{i+xh} \bmod p = g^i g^{xh} \bmod p = t(g^x)^h \bmod p = ta^h \bmod p$.

A:

Net