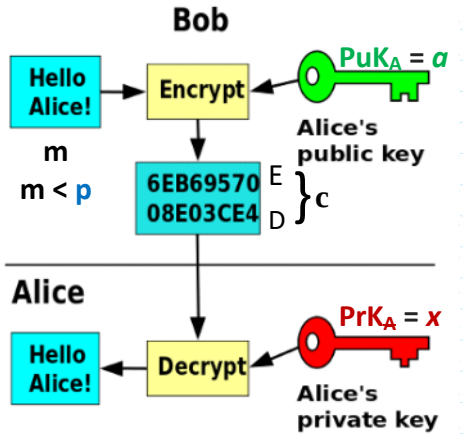
Koliokviumas vyks kontaktiniu būdu per paskaitą, Balandžio 17 d., 17:30, 140 a.
Atsineškite savo asmeninius kompiuterius.
Kolioviumo tvarka pateikta Moodle.

ElGamal encryption

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, …, **p**-1} using certain conditions.
Strong prime $p=2q+1$, where q is prime, then g is a generator of Z_P* **iff**
$g^q \neq 1 \bmod p$ and $g^2 \neq 1 \bmod p$.
Declare **Public Parameters** to the network PP = (**p**, **g**); **p= 268435019; g=2;**
2^28-1= 268,435,455

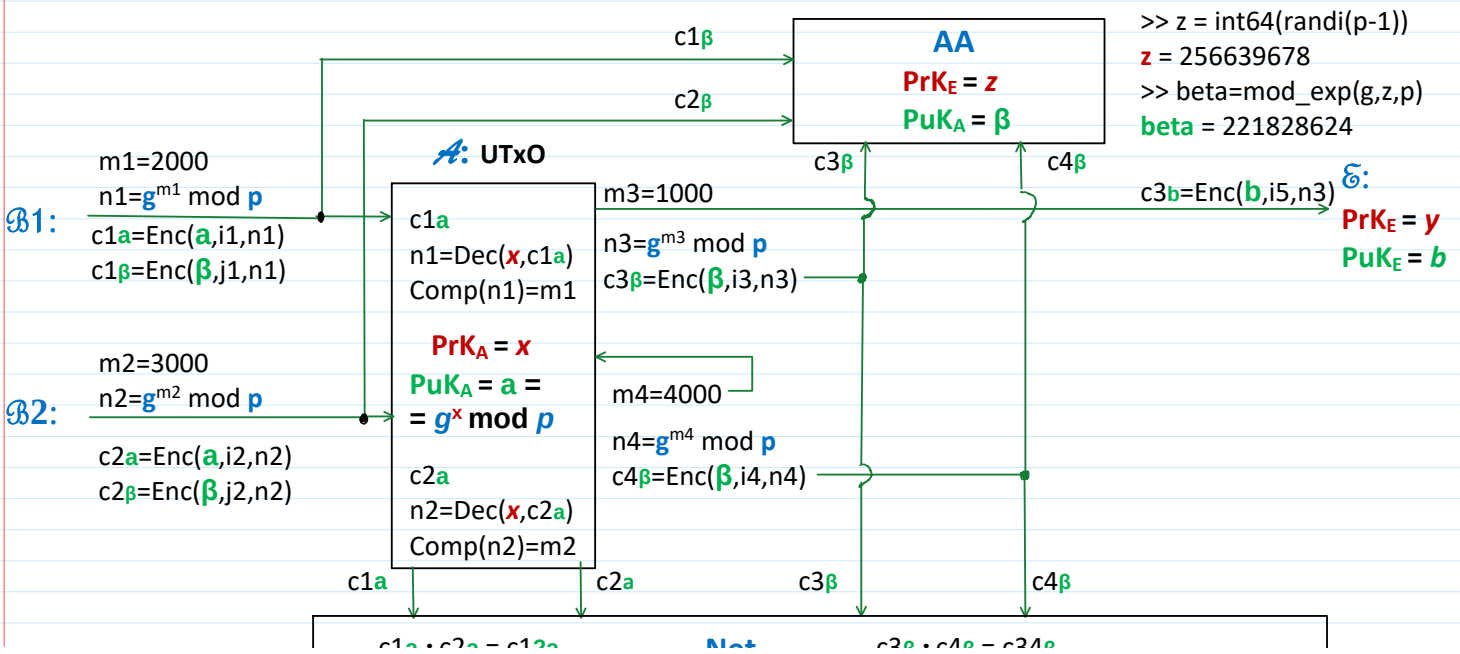
PrK = *x* <-- randi ==> PuK = *a* = g^x mod p

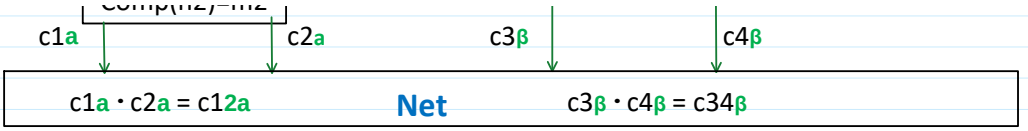


Encryption.
r <-- randi(*p*-1);
$E = m \cdot a^r \bmod p$;
$D = g^r \bmod p$;
c = (*E, D*);
The number of exponent operations is 2.

Decryption.
$D^{-x} \bmod p$;
$m = E \cdot D^{-x} \bmod p$;
The number of exponent operations is 1.

Confidential Verifiable Transactions - 2 $PP = (p, g)$.





c12a = Enc(a,i1+i2,n1·n2)
= Enc(a,i1+i2,**n12**)
n12 = n1.n2 mod p
= $g^{(m1+m2) \bmod (p-1)} \bmod p$

c34β = Enc(β,i3+i4,n3·n4)
= Enc(β,i3+i4,**n34**)
n34 = n3.n4 mod p
= $g^{(m3+m4) \bmod (p-1)} \bmod p$

If m1+m2 = m3+m4 then **n12** = n1·n2 = **n34** = n2·n4

Fig. 1. Private and verifiable transactions using ElGamal Cryptosystem.

Let us compare the number of most resources consuming operations required to realize the private and verifiable transactions using ElGamal Cryptosystem (EGC) and Elliptic Curve Cryptosystem (ECC) schemes for only 1 sender **Bob1** and 1 receiver-sender **Alice**.
In ElGamal Cryptosystem such operation is Discrete Exponent Function (DEF), e.g. of the form **a = g^x mod p** or exponentiation.
In Elliptic Curve Cryptosystem (ECC) such operation is Elliptic Curve point ***G*** multiplying by integer ***z***, e.g. ***A*** = ***z**G*
Which we name as EC exponentiation.

We assume that these operations are almost equivalent.

1. **EGC operations**.
1.1. Bob1 performs 2 encryptions: 1 for Alice and 1 for AA. Hence the number of exponentiations is 4.
We do not take into account the exponentiation for computing n1 since the number m1 is considerable small, and Comp(n1)=m1 since Alice knows the approximate sum of m1.
1.2. Alice performs 1 decryption for income from Bob1 and 2 encryptions for expenses: 1 for Ema and 1 for AA.
Hence Alice performs 5 exponentiations.
1.3. To proof the equivalence of ciphertexts c12a and c34β it is required to perform 4 exponentiations.

In total it is required to perform 13 exponentiations for Alice.

ECC: G generator; $t \leftarrow randi$
$H = t * G$; H is the other generator. $PP = (G, H)$ + other params.

$\mathcal{B}$: $m_\mathcal{J} = 2000$
k_{BA} Ped. comm. sch.
k_{BAA} $PCS_\mathcal{J}$
$PCS_\mathcal{J}$: $u \leftarrow randi$
$C_\mathcal{J} = u * G \boxplus m_\mathcal{J} * H$

$C_\mathcal{J}$, C_{BAA} → $\mathcal{AA}$: $k_{BAA} \rightarrow D(k_{BAA}, C_{BAA}) = u \,||\, m$
Verifies validity of $C_\mathcal{J}$.
$C_\mathcal{J}$, C_{BA} → $\mathcal{A}$: $k_{BA} \rightarrow D(k_{BA}, C_{BA}) = u \,||\, m$
Verifies validity of $C_\mathcal{J}$

$\mathcal{B}$ realizes Key Agreement Protocol (KAP) with $\mathcal{A}$ and agrees on common symmetric

$\mathcal{A}$: sends the sum $m_E = 2000$ to $\mathcal{E}$ so me is her expenses
$m_\mathcal{J} = m_E$

with A and agrees on common symmetric secret key k_{BA}.

B using symmetric encryption with key k_{BA} encrypts u and m_J

$E(k_{BA}, u \| m_J) = c_{BA}$

$m_J = m_E$

PCS_E : $v \leftarrow rand_i$

$C_E = v * G \boxplus m_E * H$

A realizes KAP with $\mathcal{E} \rightarrow k_{AE}$

$E(k_{AE}, v \| m_E) = c_{AE} \rightarrow \mathcal{E}$

B realizes KAP with AA and agrees on common symmetric secret key k_{BAA}.

$E(k_{BAA}, u \| m_J) = c_{BAA}$

A realizes KAP with $AA \rightarrow k_{AAA}$

$E(k_{AAA}, v \| m_E) = c_{AAA} \rightarrow AA$

A: computes commitment $C_{JE} = C_J - C_E =$

$= u * G \boxplus m_J * H \boxminus v * G \boxminus m_E * H =$

Let ***u***, ***v*** are integers < ***p***.

Property 1: $(u + v)*P = u*P \boxplus v*P$ in literature it is replaced to --> $\underline{(u + v)P = uP + vP}$

Property 2: $(u)*(P \boxplus Q) = u*P \boxplus u*Q$ in literature it is replaced to --> $u(P + Q) = uP + uQ$

$= (u - v) * G + \underbrace{(m_J - m_E)}_{0} * H =$

$= (u - v) * G = N$

A: $N = (u - v) * G \longrightarrow Net$

Till this place

1. **The list of operations for private and verifiable transactions**. To realize private and verifiable transactions based on Pedersen Commitment Scheme (PCS) beside EC exponentiation the additional operations are required. They are the following:

2.1. PCS requiring 2 EC exponentiations.
2.2. Key Agreement Protocol (KAP) requiring 2 EC exponentiations for both parties.
2.3. Symmetric encryption: we assume it is equivalent at least to 2 EC exponentiations.
2.4. Symmetric decryption: we assume it is equivalent at least to 2 EC exponentiations.
2.5. PCS opening requiring 2 EC exponentiations.

1. The total number of operations for private and verifiable transactions.

3.1. Bob performs 2 KAPs with Alice and AA requiring 2 exponentiations.
3.2. Bob performs 2 symmetric encryptions for Alice and AA for masking and amount data requiring at least 4 equivalent exponentiations.
3.3. Bob realizes 1 PCS for Alice and 1 PCS for AA requiring 4 EC exponentiations.
3.4. Alice decrypts blinding factor and amount requiring 2 equivalent exponentiations.
3.5. Alice verifies PCS requiring 2 equivalent exponentiations.

3.6. Alice realizes 1 PCS for Ema and 1 PCS for AA requiring 4 EC exponentiations.
3.7. Alice computes blinding factor for the Net to prove the validity of transaction requiring 1 EC exponentiation.

The total number of equivalent operations is 21.

Net: Computes $c12\alpha = c1\alpha * c2\alpha = (E1\alpha, D1\alpha) * (E2\alpha, D2\alpha) =$

$= (E1\alpha * E2\alpha \bmod p, D1\alpha * D2\alpha \bmod p) = (E12\alpha, D12\alpha)$

$c34\beta = c3\beta * c4\beta = (E3\beta, D3\beta) * (E4\beta, D4\beta) =$

$= (E3\beta * E4\beta \bmod p, D3\beta * D4\beta \bmod p) = (E34\beta, D34\beta)$

$E3\beta = n_3 \cdot \beta^{i_3} \bmod p$; $D3\beta = g^{i_3} \bmod p$;
$E4\beta = n_4 \cdot \beta^{i_4} \bmod p$; $D4\beta = g^{i_4} \bmod p$;
$\Rightarrow E34\beta = n_3 \cdot n_4 \cdot \beta^{(i_3+i_4) \bmod (p-1)} \bmod p$
$\Rightarrow D34\beta = g^{(i_3+i_4) \bmod (p-1)} \bmod p$

$c_{34\beta} = (E34\beta = n_{34} \cdot \beta^{i} \bmod p, D34\beta = g^{i} \bmod p)$

Taking in mind that:

1) If $m_{12} = m_1 + m_2 \bmod (p-1) = m_{34} = m_3 + m_4 \bmod (p-1)$

$n_{12} = n_1 \cdot n_2 \bmod p \Longleftrightarrow n_{34} = n_3 \cdot n_4 \bmod p$

2) Then $Dec(PrK=x, c12\alpha) = n_{12} = g^{(m_1+m_2) \bmod (p-1)} \bmod p$

$Dec(PrK=z, c34\beta) = n_{34} = g^{(m_3+m_4) \bmod (p-1)} \bmod p$

Then $c12\alpha$ and $c34\beta$ encrypts the same number $n_{12} = n_{34} = n$.

But since $\alpha \neq \beta \rightarrow c12\alpha \neq c34\beta$ in any way!

A: Must prove that ciphertexts $c12\alpha$ and $c34\beta$ encrypted the

$\mathcal{A}$: Must prove that ciphertexts $c12_\alpha$ and $c34_\beta$ encrypted the same number $n = n_{12} = n_{34}$ ⟺ balance $= (m_1 + m_2) \bmod (p-1) = (m_3 + m_4) \bmod (p-1) = 5000$.

This is named as ciphertexts equivalency problem.

Proof. $i = i_{34} = (i_3 + i_4) \bmod (p-1)$

```
>> i34=mod(i3+i4,p-1)
i34 = 115795473
```

1) $\mathcal{A}$ proves to the Net that she knows her $PrK_A = x$ by declaring her $PuK_A = a$ using NIZKP.

2) $\mathcal{A}$ proves to the Net that she knows her random parameter $i = i_{34}$ $(i_3 + i_4) \bmod (p-1)$ for $n_{34} = n_3 * n_4 \bmod p$ encryption. Random parameters i_3 and i_4 must be secret otherwise encrypted values n_3 and n_4 can be decrypted without a knowledge of her $PrK = x$.

3) $\mathcal{A}$ referencing to these proofs provides a ciphertexts equivalency proof.

Non-Interactive Zero Knowledge Proof - NIZKP $\mathbf{PP} = (p, g)$.

$\mathcal{A}$: NIZKP of knowledge x:
$PrK_A = x$ = randi(p-1)
$PuK_A = a = g^x \bmod p$
1. Computes r for random number u:
   u=randi(p-1)
   $r = g^u \bmod p$
2. Generates h:
   h=randi(p-1)
3. Computes:
   $s = u + xh \bmod (p-1)$

$PuK_A = a$ →
(r, s)

$\mathcal{B}$: $PuK_A = a$
Verifies:
$g^s = ra^h \bmod p$

$PrK_A = x$ is called witness for a statement $PuK_A = a$.

Let $\mathcal{A}$ wants to prove the knowledge of x and $i = i_{34}$.

Then the statement $st = \{a = g^x \bmod p,\ D_{34\beta} = g^i \bmod p\}$

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

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

Commitments t_1 and t_2 are generated:

$$\left.\begin{aligned} t_1 &= g^u \bmod p \\ t_2 &= g^v \bmod p \end{aligned}\right\} h = H(a \| D_{34\beta} \| t_1 \| t_2)$$

Net: $\{a, D_{34\beta}, t_1, t_2\}$

$h = H(a \| D_{34\beta} \| t_1 \| t_2)$

$$\left.\begin{aligned} r &= x \cdot h + u \bmod (p-1) \\ s &= i \cdot h + v \bmod (p-1) \end{aligned}\right\} \xrightarrow{\text{Net verifies}} \begin{aligned} g^r &= t_1 \cdot a^h \bmod p \\ g^s &= t_2 \cdot (D_{34\beta})^h \bmod p \end{aligned}$$

Correctnes:

$$g^r = g^{(x \cdot h + u) \bmod (p-1)} \bmod p = g^{xh} \cdot g^u = (g^x)^h \cdot g^u = a^h \cdot t_1$$

$$g^s = g^{(i \cdot h + v) \bmod (p-1)} \bmod p = g^{ih} \cdot g^v = (g^i)^h \cdot g^v = (D_{34\beta})^h \cdot t_2$$

$$Enc(a, i_1, n_1) = c_{1a} = (E_{1a}, D_{1a}) = (n_1 \cdot a^{i_1}, g^{i_1}) \bmod p$$

$$Enc(a, i_2, n_2) = c_{2a} = (E_{2a}, D_{2a}) = (n_2 \cdot a^{i_2}, g^{i_2}) \bmod p$$

$$c_{1a} \cdot c_{2a} = c_{12a} = Enc(a, i_{12}, n_{12}) = (E_{12a}, D_{12a}) = (n_{12} \cdot a^{i_{12}}, g^{i_{12}}) = c_{12a}$$

$$i_{12} = (i_1 + i_2) \bmod (p-1)$$

$$n_{12} = n_1 \cdot n_2 \bmod p$$

$$c_{3\beta} \cdot c_{4\beta} = c_{34\beta} = Enc(\beta, i_{34}, n_{34}) = (E_{34\beta}, D_{34\beta}) = (n_{34} \cdot \beta^{i_{34}}, g^{i_{34}}) = c_{34\beta}$$

$$i_{34} = (i_3 + i_4) \bmod (p-1)$$

$$n_{34} = n_3 \cdot n_4 \bmod p$$

If transaction balance is valid: $m_1 + m_2 = 2000 + 3000 = 1000 + 4000 = m_3 + m_4$

then since:

$$\left.\begin{aligned} n_{12} &= n_1 \cdot n_2 = g^{m_1} \cdot g^{m_2} \bmod p = g^{m_1 + m_2} \bmod p \\ n_{34} &= n_3 \cdot n_4 = g^{m_3} \cdot g^{m_4} \bmod p = g^{m_3 + m_4} \bmod p. \end{aligned}\right\} n_{12} = n_{34} = n$$

A: must prove to the net, that c_{12a} and $c_{34\beta}$ encrypted the same value $n_{12} = n_{34} = n$; ⟶ Ciphertexts Equivalency Proof.

The statement St for this proof is the following:

$St = \{C_{12}\alpha, C_{34}\beta, \alpha, \beta\}$; For example: $\alpha = g^x \bmod p$

$PuK = \alpha$ is a statement for x.

For proof A randomly generates integers u, v and $(-v) \bmod (p-1)$

$u \leftarrow randi(\mathcal{I}_{p-1})$; $\mathcal{I}_{p-1} = \{0, 1, 2, \ldots, p-2\}$

$v \leftarrow randi(\mathcal{I}_{p-1})$

$-v \bmod (p-1) \longrightarrow >> mv = mod(-v, p-1)$

1. The following commitments $\{t_1, t_2, t_3\}$ are computed:

$t_1 = g^u \bmod p$

$t_2 = g^v \bmod p$

$t_3 = (D_{12}\alpha)^u \cdot \beta^{-v} \bmod p$

2. The following h-value is computed using secure h-function H:

$h = H(\alpha \| \beta \| t_1 \| t_2 \| t_3)$

3. A having her PrK x and $i_{34} = (i_3 + i_4) \bmod (p-1)$ computes r and s

$r = (x \cdot h + u) \bmod (p-1)$

$s = (i_{34} \cdot h + v) \bmod (p-1)$

and declares the following set of data to the Net

$\{C_{1\alpha}, C_{2\alpha}, C_{3\beta}, C_{4\beta}\} \cup \{\alpha, \beta, t_1, t_2, t_3, r, s\} \longrightarrow$ Net

Net computes h-value defined above

$h = H(\alpha \| \beta \| t_1 \| t_2 \| t_3)$

Net verifies transaction correctness by verifying the following identities

$g^r = \alpha^h \cdot t_1 \bmod p$

$g^s = (D_{34}\beta)^h \cdot t_2 \bmod p$

$(E_{34}\beta)^h \cdot (E_{12}\alpha)^{-h} \cdot (D_{12}\alpha)^r \cdot \beta^{-s} = t_3 \bmod p$

Declare **Public Parameters** to the network **PP** = (**p**, **g**); **p= 268435019**; **g=2;**

PrK_A = *x* <-- randi ==> PuK_A = *a* = g^x mod p **PrK_A = *x* <-- randi ==> PuK_A = *a* = g^x mod p**

```
>> p=int64(268435019)
p = 268435019
>> g=2;
```

```
>> x=int64(randi(p-1))
x = int64(220099152)
>> a=mod_exp(g,x,p)
a = 174059961
```

```
>> z=int64(randi(p-1))
z = int64(49750938)
>> beta=mod_exp(g,z,p)
beta = int64(213338364)
```

Incomes

```
>> m1=2000;
>> n1=mod_exp(g,m1,p)
n1 = 28125784
>> i1=int64(randi(p-1))
i1 = int64(207414820)
>> a_i1=mod_exp(a,i1,p)
a_i1 = 192148999
>> E1a=mod(n1*a_i1,p)
E1a = 207347548
>> D1a=mod_exp(g,i1,p)
D1a = 202537833
```

c1*a* = (E1*a*, D1*a*)

Verification: Dec(*x*, c1*a*) = nn1

```
>> mx=mod(-x,p-1)
mx = 48335866
>> D1a_mx=mod_exp(D1a,mx,p)
D1a_mx = 75547583
>> nn1=mod(E1a*D1a_mx,p)
nn1 = 28125784
```

```
>> m2=3000;
>> n2=mod_exp(g,m2,p)
n2 = 222979214
>> i2=int64(randi(p-1))
i2 = int64(67446699)
>> a_i2=mod_exp(a,i2,p)
a_i2 = 211790072
>> E2a=mod(n2*a_i2,p)
E2a = 77938423
>> D2a=mod_exp(g,i2,p)
D2a = 82080815
```

C2*a* = (E2*a*, D2*a*)

Verification: Dec(*x*, c2*a*) = nn2

```
>> mx=mod(-x,p-1)
mx = 48335866
>> D2a_mx=mod_exp(D2a,mx,p)
D2a_mx = 57701660
>> nn2=mod(E2a*D2a_mx,p)
nn2 = 222979214
```

```
>> E12a=mod(E1a*E2a,p)
E12a = 52532683
>> D12a=mod(D1a*D2a,p)
D12a = 32918394
```

C12a = (E12a, D12a)

Expenses

```
>> m3=1000;
>> n3=mod_exp(g,m3,p)
n3 = 260099963
>> i3=int64(randi(p-1))
i3 = int64(137379932)
>> beta_i3=mod_exp(beta,i3,p)
beta_i3 = 14259017
>>  E3beta=mod(n3*beta_i3,p)
```

```
>> m4=4000;
>> n4=mod_exp(g,m4,p)
n4 = 246637967
>> i4 = int64(randi(p-1))
i4 = int64(225960178)
>> beta_i4=mod_exp(beta,i4,p)
beta_i4 = 159771180
>> E4beta=mod(n4*beta_i4,p)
```

```
>> E34beta=mod(E3beta*E4beta,p)
E34beta = 57420210
>> D34beta=mod(D3beta*D4beta,p)
D34beta = 107062668
```

C34beta = (E34beta, D34beta)

>> beta_i3=mod_exp(beta,i3,p)
beta_i3 = 14259017
>> E3beta=mod(n3*beta_i3,p)
E3beta = 167897317
>> D3beta=mod_exp(g,i3,p)
D3beta = 65145889

Verification: Dec(***z***, c3**beta**) = nn3
>> mz=mod(-z,p-1)
mz = 218684080
>> D3beta_mz=mod_exp(D3beta,mz,p)
D3beta_mz = 258869169
>> nn3=mod(E3beta*D3beta_mz,p)
nn3 = 260099963

>> beta_i4=mod_exp(beta,i4,p)
beta_i4 = 159771180
>> E4beta=mod(n4*beta_i4,p)
E4beta = 195130083
>> D4beta=mod_exp(g,i4,p)
D4beta = 229603826

Verification: Dec(***z***, c3**beta**) = nn3
>> mz=mod(-z,p-1)
mz = 218684080
>> D4beta_mz=mod_exp(D4beta,mz,p)
D4beta_mz = 218460911
>> nn4=mod(E4beta*D4beta_mz,p)
nn4 = 246637967

C34beta = (E34beta, D34beta)

The modification of the existing NIZKP to prove the equivalency of two cophertexts.
Namely c_{12a} and Ca,I in (18), (19), and $C\beta,E$ in (20), (21). Recall that Ca,I is a ciphertext of plaintext I encryption with Alice's PuK=a and $C\beta,E$ is a ciphertext of plaintext E encryption with the AA's PuK=β. The statement St of our proposed NIZKP consists of the following:

$$St = \{(\varepsilon_{a,I}, \delta_{a,I}), (\varepsilon_{\beta,E}, \delta_{\beta,E}), a, \beta\}. \quad (22)$$

The random integers $u \leftarrow randi(Z_q)$ and $v \leftarrow randi(Z_q)$ are generated by Alice, and the value $(-v) \bmod q$ is computed. The proof of ciphertext equivalence is computed using three computation steps:

1. The following commitments are computed:
$$t_1 = g^u \bmod p; \quad (23)$$
$$t_2 = g^v \bmod p; \quad (24)$$
$$t_3 = (\delta_{a,I})^u \cdot \beta^{-v} \bmod p. \quad (25)$$
2. The following h-value is computed using the cryptographically secure h-function H:
$$h = H(a||\beta||t_1||t_2||t_3||). \quad (26)$$
3. Alice, having her $PrK_A=x$ randomly generates the secret number l for E encryption and computes the following two values:
$$r = x \cdot h + u \bmod q; \quad (27)$$
$$s = l \cdot h + v \bmod q. \quad (28)$$
Then Alice declares the following set of data to the Net:
$$\{a, \beta, t_1, t_2, t_3, r, s\} \rightarrow \text{Net}. \quad (29)$$
To verify the transaction's validity, the Net computes the h-value according to (26) and then verifies three identities:
$$g^r = a^h \cdot t_1; \quad (30)$$
$$g^s = (\delta_{\beta,E})^h \cdot t_2; \quad (31)$$
$$(\varepsilon_{\beta,E})^h \cdot (\varepsilon_{a,I})^{-h} \cdot (\delta_{a,I})^r \cdot \beta^{-s} = t_3. \quad (32)$$

To verify the transaction's validity, the Net computes the h-value according to (26) and then verifies three identities:
$$g^r = a^h \cdot t_1; \quad (30)$$
$$g^s = (\delta_{\beta,E})^h \cdot t_2; \quad (31)$$
$$(\varepsilon_{\beta,E})^h \cdot (\varepsilon_{a,I})^{-h} \cdot (\delta_{a,I})^r \cdot \beta^{-s} = t_3. \quad (32)$$