

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**

n3•n4= g^{m1+m2} **mod p** **n3•n4** = g^{m3+m2} mod **p**

[Dan Boneh]:

If **m1+m2 = m3+m4 mod (p-1)**,

Then **n1*n2 = n3*n4 mod p**.

Public Parameters: >> **p** = int64(268435019); **g** = **2**; βεδ

𝒜: PrK$_A$ = *x*; PuK$_A$ = *a* = g^x mod *p*.

>> x=int64(randi(p-1))

x = 88637397

>> a=mod_exp(g,x,p)

a = 71787401

𝒜𝒜: PrK=z; PuK = β = g^z mod *p*.

>> z=int64(randi(p-1))

z = 240828179

>> β=mod_exp(g,z,p)

β = 170660741

ℰ: PrK=z; PuK=e = g^z mod *p*.

>> z=int64(randi(p-1))

y = 240828179

>> e=mod_exp(g,z,p)

e = 170660741

ElGamal Encryption of message $n = g^m$ mod **p**:

with **PuK$_A$ = *a***:

i ← int64(randi(*p*-1))

E = *n*•a^i **mod** ***p***.

D = g^i **mod** ***p***.

c = Enc(***a***,***i***,*n*) = (E, D)

ε = *n*•a^i **mod** ***p*** = *E* - alternative notation

δ = g^i **mod** ***p*** = *D* - alternative notation

ElGamal Decryption of ciphetrtext *c* = (ε, δ):

with **PrK$_A$ = *x***:

Compute D^{-x} **mod** ***p***.

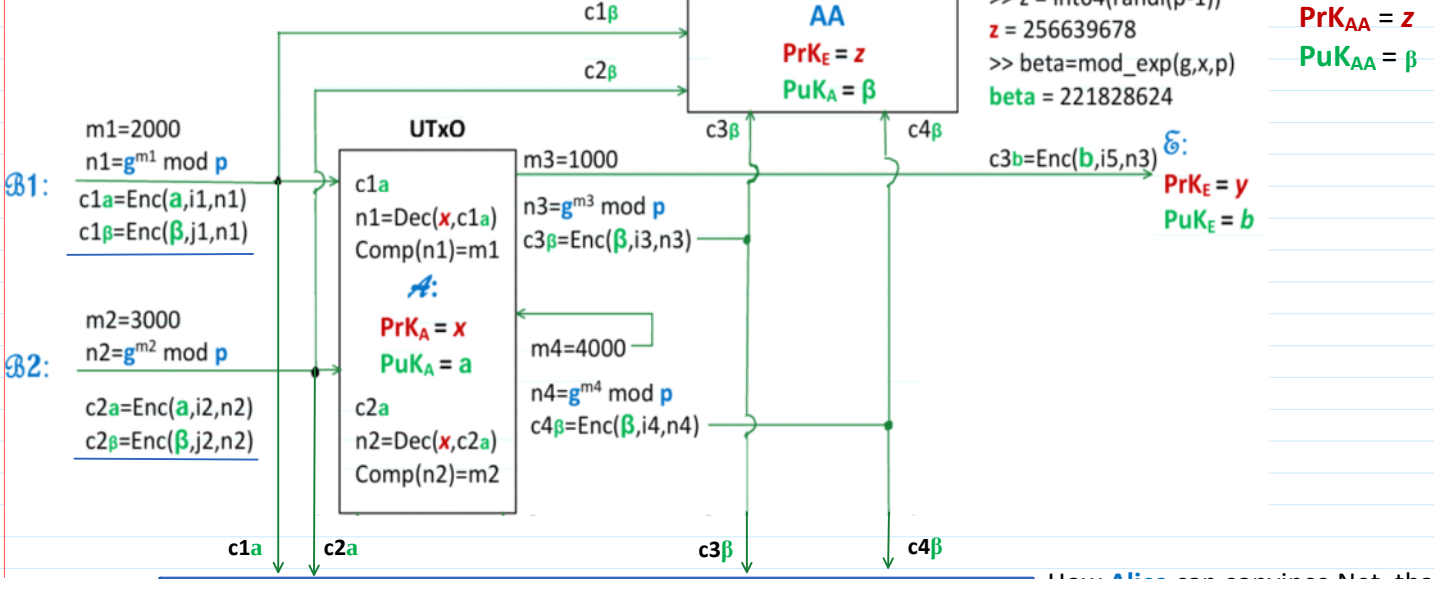
Compute E•D^{-x} **mod** ***p*** = *n*.

Alternative notations: *c* = (ε, δ)

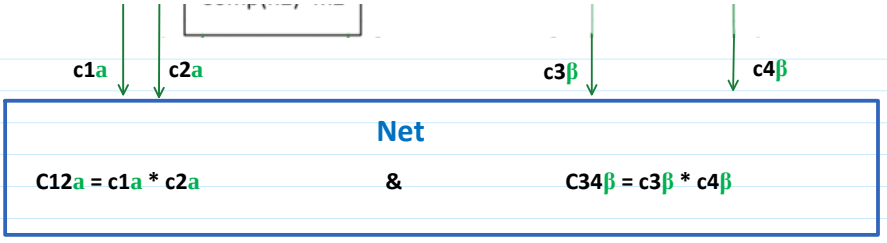
Compute δ^{-x} **mod** ***p***.

Compute ε•δ^{-x} **mod** ***p*** = *n*.

Confidential Verifiable Transactions



How Alice can convince Not that



How **Alice** can convince Net, that C12**a** and C34**β** encrypts the same sum Of total Incomes and Expenses?:
2000 + 3000 = 5000 = 1000 + 4000

To achieve that it is required for **Bob1**, **Bob2** to encrypts *i*1 and *i*2 with **Alice** $PuK_A = a$ obtaining ciphertexts c1*a*1 and c2*a*2 and sending these ciphertexts to **Alice in addition**.

Bob1: *t*1 ← int64(randi(*p*-1))	**Bob2**: *t*2 ← int64(randi(*p*-1))
Enc(*a*,*t*1,*i*1) = c1*a*1	Enc(*a*,*t*2,*i*2) = c2*a*2

In this case only **Alice** is **responsible** to prove that his transaction is honest against the **Net**.
We leave no actions to **AA** to provide the honesty of transactions to the **Net**.

Proof. **C12*a*** and **C34β** encrypts the same sum of total Incomes and Expenses: 2000 + 3000 = 5000 = 1000 + 4000.

1. **Alice** decrypts **c1*a*** and **c2*a*** and obtains **n1** and **n2**.
2. **Alice** finds the transfered sums **m1** and **m2** by total scan of relation **n1**= g^{m1} mod **p** and **n2**= g^{m2} mod **p**.
3. **Alice** decrypts **c1*a*1** and **c2*a*2** with her $PuK_A = x$ and finds Dec(*x*,**c1*a*1**) = **i1** and Dec(*x*,**c2*a*2**) = **i2**.
4. **Alice** computes **i3** and **i4** satisfying the equation:
   i1 + **i2** = **i3** + **i4** = **i** mod (**p**-1), where **i3** is choosen at random and **i4** is computed according to this relation.
5. **Alice** encrypts **n3** and **n4** with **AA** $PuK_{AA} = \beta$ using random generated integers **i3** and **i4**:

Enc(β,*i*1,*n*3) = c3β	Enc(β,*i*4,*n*4) = c4β

Notice that:	c1*a* = (E_1, D_1)	c3β = (E_3, D_3)
	c2*a* = (E_2, D_2)	c4β = (E_4, D_4)

$E_1 = n_1 \bullet a^{i1} \bmod p$ & $E_2 = n_2 \bullet a^{i2} \bmod p$ $\qquad$ $E_3 = n_3 \bullet \beta^{i3} \bmod p$ & $E_4 = n_4 \bullet \beta^{i4} \bmod p$

$E_{12} = n_1 \bullet n_2 \bullet a^{i1+i2} \bmod p$ $\qquad$ $E_{34} = n_3 \bullet n_4 \bullet \beta^{i3+i4} \bmod p$

$E_{12} = n_{12} \bullet a^{i12} \bmod p$ $\qquad$ $E_{34} = n_{34} \bullet \beta^{i34} \bmod p$

$n_{12} == n == n_{34} \bmod p$

$i_{12} == i == i_{34} \bmod (p-1)$

$E_{12} = n \bullet a^{i} \bmod p$ $\qquad$ $E_{34} = n \bullet \beta^{i} \bmod p$

Property:

$$\frac{E_{12}}{E_{34}} \;===\; \frac{n_{12} \bullet a^{i12} \bmod p}{n_{34} \bullet \beta^{i34} \bmod p} \;===\; \frac{\cancel{n} \bullet a^{i} \bmod p}{\cancel{n} \bullet \beta^{i} \bmod p} \;===\; \frac{a^{i} \bmod p}{\beta^{i} \bmod p} \;===\; (a/\beta)^{i} \bmod p = d^{i} \bmod p.$$

Public keys ratio is denoted by **d**=(**a**/**e**) that is known to all **Net**.

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 $PrK_A = x$.

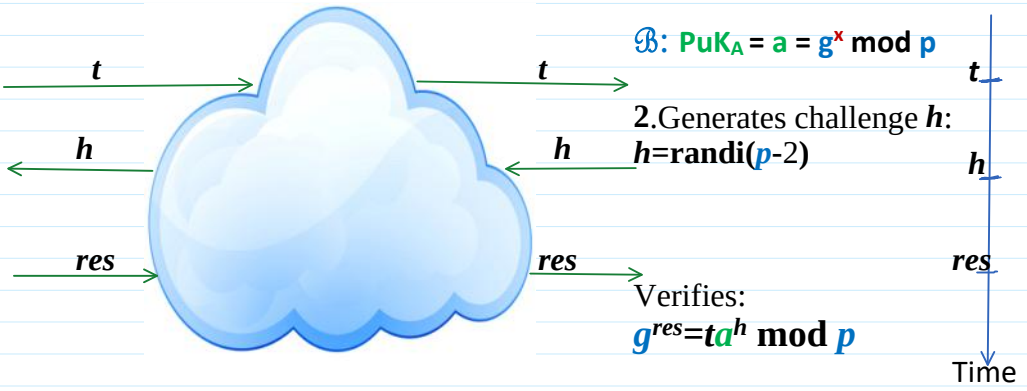
𝒜: ZKP of knowledge x:

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



ℬ: $PuK_A = a = g^x \bmod p$

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



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

Instead of using **ZKP**, **Alice** can use Schnorr signature to proof the knowlegce of ***x***.

Alice proof that **C12a** and **C34β** encrypts the same sum of Incomes and Expenses:
2000 + 3000 = 5000 = 1000 + 4000

1.Computes commitment ***t*** for ***i***:
$i=i_{12}=i_{34}$
$t=g^i \bmod p$
2.Computes commitment ***u*** for ***l***:
l=randi(***p***-1)
$u=g^l \bmod p$
3.Computes commitment ***v*** for ***l*** for ratio d=(a/β):
$v=(a/\beta)^l \bmod p = d^l \bmod p$.
4.Computes h-value:
$h=H(u||v)$
5.Signs ***h*** by ***i*** and ***l*** by computing ***s***:
$s=l+ih \bmod (p-1)$

t, u, v, s
𝒜: →

c1***a*** = (E_1, D_1); c4β = (E_4, D_4)
c2***a*** = (E_2, D_2); c3β = (E_3, D_3)

Net: $h=H(u||v)$

Net: $PuK_A = a$; $PuK_{AA} = \beta$; $(a/\beta)^i = d^i \bmod p$.
Verifies: **1)** If $g^s = ut^h \bmod p$
2) If $d^s = v(E_{12}/E_{34})^h \bmod p$
If (1) and (2) verifications passes then **Net** verifies if:
E_{12}
----- === $(a/e)^i \bmod p = d^i$.
E_{34}

Correctness:

1. $g^s = g^{l+ih} \bmod p = g^l g^{ih} \bmod p = u(g^i)^h = ut^h \bmod p$.

2. $d^s = d^{l+ih} \bmod p = d^l d^{ih} \bmod p = v(d^i)^h == v(E_{12}/E_{34})^h \bmod p$.

Till this place

c12=c1*c2

$C_{12} = (E_1, D_1) * (E_2, D_2) =$

c34=c3*c4

$C_{34} = (E_3, D_3) * (E_4, D_4) =$

$= (E_1 * E_2 \bmod p, D_1 * D_2 \bmod p)$ $= (E_3 * E_4 \bmod p, D_3 * D_4 \bmod p)$

Encrypted with $\begin{cases} E_1 * E_2 \bmod p \neq E_3 * E_4 \bmod p \\ D_1 * D_2 \bmod p = D_3 * D_4 \bmod p \end{cases}$ Encrypted with

a e

Since $i_{12} = i_{34}$

Confidential Transaction Balance Verification Scheme. This section presents first integrated cryptographic schemes designed for a private blockchain using the Unspent Transaction Output (UTXO) paradigm, focusing on transaction confidentiality and user anonymity with controlled deanonymization.

The proposed scheme in fig. 1 ensures confidential transaction balance verification in a private blockchain using the UTXO paradigm. It integrates additively-multiplicative homomorphic ElGamal encryption with a modified Schnorr-based non-interactive zero-knowledge proof (NIZKP), enabling the network (Net) to verify the balance equation $I = E$ (income = expenses) without revealing actual values. Alice proves that ciphertexts

$$C_{12a} = (\varepsilon_{a,I}, \delta_{a,I}) \tag{7.1}$$

and

$$C_{34\beta} = (\varepsilon_{\beta,I}, \delta_{\beta,I}), \tag{7.2}$$

encrypted with different public keys (a and β), correspond to the same value.

ElGamal Encryption: Operates in the the subgroup G_q of cyclic group Z_p^* = {1, 2, 3, ..., p-1} where p and q are primes and q is a divider of p. Then all elements of G_q except 1 are the generators since group G_q is of prime order (the number of elements of G_q is prime). Let g be a generator of G_q. The exponents of elements of G_q are in the field $_q$ = {0, 1, 2, ..., q-1} with addition and multiplication operations performed *mod q*. Public parameters are:

$$PP = (p, g). \tag{7.3}$$

– *Key Generation*: For Alice: private key

$$x \leftarrow randi(Z_q), \tag{7.4}$$

public key

$$a = g^x \bmod p; \tag{7.5}$$

for AA: private key $\leftarrow randi(Z_q)$, public key

$$\beta = g_z \bmod p \tag{7.6}$$

– *Encryption*: Given message d, random

$$l \leftarrow randi(Z_q), \tag{7.7}$$

ciphertext:

$$c_\beta = (\varepsilon_\beta, \delta_\beta) = (d \cdot \beta^l, g^l) \bmod p \tag{7.8}$$

– *Decryption*:

$$= \varepsilon_\beta \cdot (\delta_\beta)^{-z} \bmod p. \tag{7.9}$$

– *Additive Homomorphism*: Incomes i_m and expenses e_n are transformed:

$$I_m{}^{(i)} = g^{i_m{}^{(i)}} \bmod p, E_n{}^{(j)} = g^{e_n{}^{(j)}} \bmod p. \tag{7.10}$$

Total encrypted income $C_{a,I}$ and expenses $C_{\beta,E}$:

$$C_{a,I} = (\varepsilon_{a,I}, \delta_{a,I}) = (I \cdot a^k, g^k), \tag{7.11}$$

$$C_{\beta,E} = \left(\varepsilon_{\beta,E}, \delta_{\beta,E}\right) = \left(E \cdot \beta^{l}, g^{l}\right) \quad (7.12)$$

Where

$$= I_1 \cdot \ldots \cdot I_M, E = E_1 \cdot \ldots \cdot E_{N+1}, k = k_1 + \ldots + k_M \, mod \, q \quad (7.13)$$

NZKP for Ciphertext Equivalency: Alice proves that

$$C_{12a} = C_{a,I} \quad (7.14)$$

and

$$C_{34\beta} = C_{\beta,E} \quad (7.15)$$

encrypt $I = E$.

- *Commitments*: Random

$$u, v \leftarrow randi(\mathbb{Z}_p), \quad (7.16)$$

$$t_1 = g^{u} mod \, p; t_2 = g^{v} mod \, p; t_3 = \left(delta_{a,I}\right)^{u} \cdot \beta^{-v} mod \, p. \quad (7.17)$$

- *Challenge*:

$$h = H\,(a \,||\, \beta \,||\, t_1 \,||\, t_2 \,||\, t_3), \quad (7.18)$$

where H is a cryptographically secure hash function.

- *Responses*:

$$= x \cdot h + u \, mod \, p \quad (7.19)$$

$$= l \cdot h + v \, mod \, p \quad (7.20)$$

- Verification: The Net verifies:

$$= l \cdot h + v \, mod \, p \quad (7.21)$$

$$g^{r} = a^{h} \cdot t_1; \; g^{s} = (\delta_{\beta,E})^{h} \cdot t_2; \; (\varepsilon_{\beta,E})^{h} \cdot (\varepsilon_{a,I} I)^{-h} \cdot (\delta_{a,I})^{r} \cdot \beta^{-s} = t_3 \quad (7.22)$$

If $I = E$, the equations hold, proving the balance.

Security and Efficiency: The scheme is secure against eavesdropping attacks (based on ElGamal semantic security and DDH asumption). NIZKP reduces the number of encryptions/decryptions from 2M to 4 modular exponentiations, improving efficiency.