

Schnorr Signature

Public parameters: $PP = (p, g)$; $p = \text{int64}(268435019)$; $g = 2$;

In the case of Schnorr cryptosystem our simulation is performed with Public Parameters:

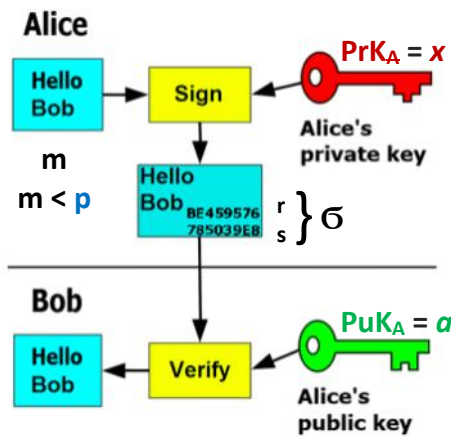
$PP = (p, g)$; $p = 268435019$; $g = 2$;

$p = \text{int64}(268435019)$

By having PP private key PrK and public key PuK are generated:

$PrK = x \leftarrow \text{randi}(p-1)$

$PuK = a = g^x \bmod p$.



$u \leftarrow \text{randi}(p-1)$.

$r = g^u \bmod p$.

$h = H(M || r)$.

$\gg \text{con} = \text{concat}(M, r)$

$\gg h = \text{hd28}(\text{con})$

$s = u + xh \bmod (p-1)$. (*) $\gg s = \text{mod}(u + xh, p-1)$

Alice's signature on h is $\sigma = (r, s)$.

Notice that it is infeasible to find x from (*), when s and h are given, since there is 1 equation (*) and 2 unknowns u and x .

Signature is valid if: $\underset{V1}{g^s \bmod p} = \underset{V2}{r a^h \bmod p}$. (Eq.1)

Schnorr multisignature

$A : PrK_A = x ; PuK_A = a$.

$B : PrK_B = y ; PuK_B = b$.

m - message to be signed by A and B

$h = H(m)$

$\text{Sign}_A(x, h) = (r_A, s_A) = \sigma_1$

$\text{Sign}_B(y, h) = (r_B, s_B) = \sigma_2$

$k_A \leftarrow \text{rand}$

$k_B \leftarrow \text{rand}$

$r_A = g^{k_A} \bmod p$

$r_B = g^{k_B} \bmod p$

$s_A = k_A + xh \bmod (p-1)$

$s_B = k_B + yh \bmod (p-1)$

Schnorr multisignature σ_{12} :

$$\tilde{G}_{12} = \tilde{G}_1 \cdot \tilde{G}_2 = (r_A, s_A) \cdot (r_B, s_B) = (\underbrace{r_A \cdot r_B \bmod p}_{r_{12}}, \underbrace{s_A + s_B \bmod (p-1)}_{s_{12}})$$

Verification

$$g^{s_{12}} = r_{12} (ab)^h \bmod p$$

$$g^{s_A} \cdot g^{s_B} = r_A \cdot r_B \cdot a^h b^h \bmod p = r_A r_B (ab)^h \bmod p$$

In 2024.11 Donald Trump declared America to be a Bitcoin country.
Possibly inspired by Elon Musk.

Anonymity in private Blockchain

Let **Alice** opened her Bitcoin account with Bitcoin Address by generating her private key **PrK=x** and public key **PuK=a**.
We assume that **PuK=a** are linked to **Alice** Address in Bitcoin.

In Bitcoin and other Blockchains the Address is computed as a function of user's public key:
Addr_A = F(PuK) and consist of several dozens of decimal numbers.

Cryptocurrency transaction

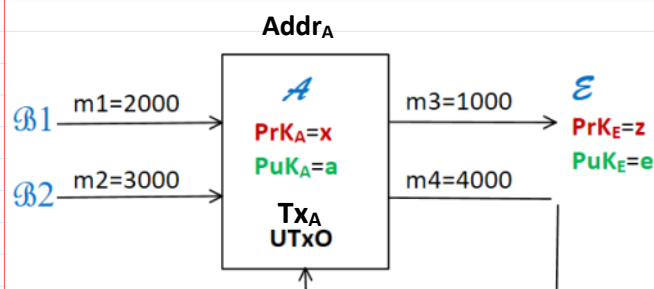
No.	Pajamos-Incomes	Išlaidos-Expenses	Likutis-Balance
In1.	Client1: 1000 Sat		1000 Sat
In2.	Client2: 2000 Sat	Out1. Firm 5: 1700 Sat	1300 Sat
In3.	Client3: 3000 Sat	Out2. Firm 6: 2300 Sat	2000 Sat
In4.	Client4: 4000 Sat	Out3. Firm 7:	6000 Sat
Total	10 000 Sat	4000 Sat	6000 Sat

Sum of Inputs =
= Sum of Outputs
Divisibility

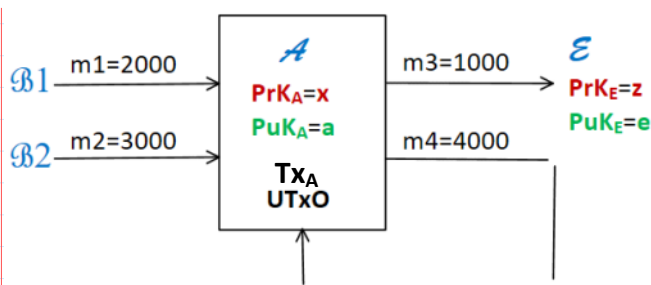
Unspent Transaction Output - **UTxO** paradigm

Transaction (Tx) information in simplified form consist of the following information:

1. The address of Tx creator.
2. The sums of Incomes and addresses of senders.
3. The sums of Expenses and addresses of receivers.



Alice has a certificate **Cert_A** for her **PuK_A = a = g^x mod p**.



Alice has a certificate Cert_A for her $\text{PuK}_A = a = g^x \bmod p$.

Schnorr-Aggregated-Signature Anonymization in BlockChain

But **Alice** do not want that all her incomes belonging to her Address were known and therefore and she prefers to be anonymous to the **Net**.

Then she creates a set of Addresses by generating a set of private keys $\{\text{PrK}_i = x_i\}$ and a set of public keys $\{\text{PuK}_i = a_i\}$, where $i=1, 2, \dots, N$.

But! There are the situations when **Alice** must prove some subjects that she possesses some amount of money distributed among a lot of her accounts and transactions with different addresses.

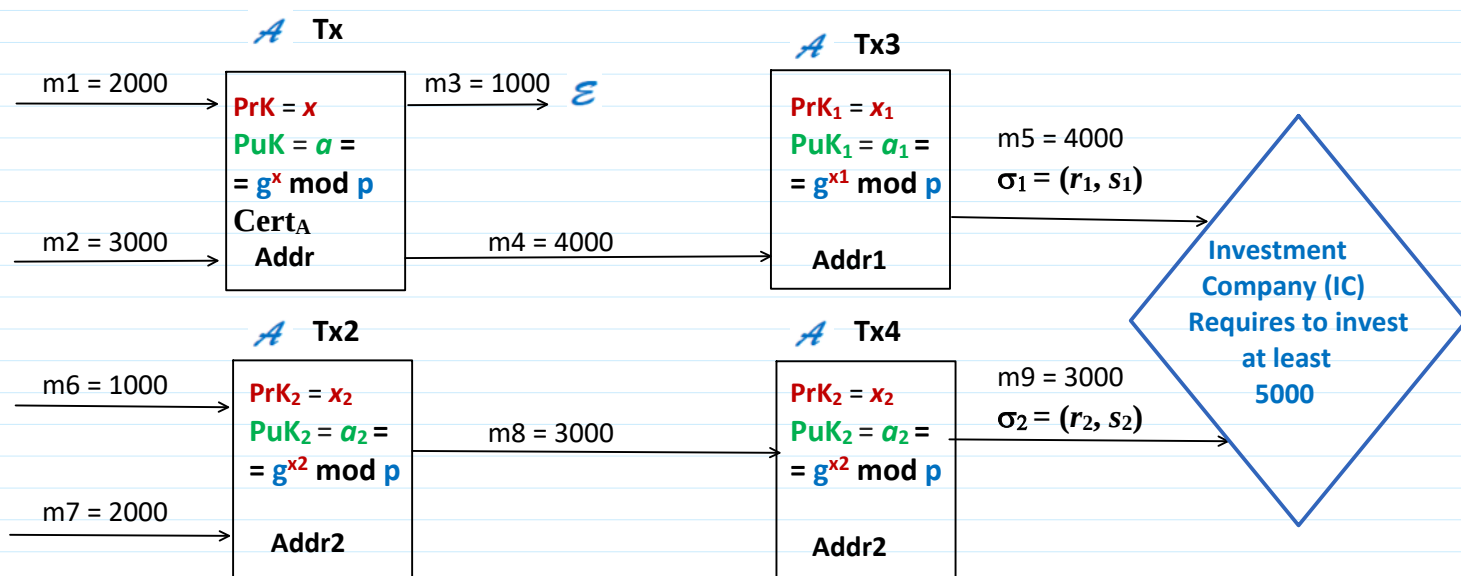
For example, she could pretend to tax concessions - mokesčiu lengvatos (according to the law) and she must prove to certain Investment Company that she possesses sufficient amount of money.

In this case she must prove that she controls some accounts with this sufficient amount of money for investment.

In this case **Alice** can prove that her transactions are authentic (i.e. are created by her) by proving that $\text{PuK}=a$ belongs to her, e.g. using Certificate issued by Certificate Authority for $\text{PuK}=a$, but at the same time she remains **anonymous** for other part of the **Net**.

For example.

Alice generated two private keys x_1 and x_2 satisfying relation $x = x_1 + x_2 \bmod (p-1)$



Alice must sign on different transactions **Tx3** and **Tx4**.

In this case the group consist of 2 anonymous addresses **Addr1** and **Addr2** belonging to **Alice**.

Alice creates the following signatures σ_1, σ_2 for Addr1 and Addr2.

Addr1	Addr2
$\text{PrK}_1 = x_1, \text{PuK}_1 = a_1;$	$\text{PrK}_2 = x_2, \text{PuK}_2 = a_2;$
$u_1 \leftarrow \text{randi}(p-1);$	$u_2 \leftarrow \text{randi}(p-1);$
$r_1 = g^{u_1} \bmod p;$	$r_2 = g^{u_2} \bmod p;$
$h_1 = H(\text{Tx3} r_1);$	$h_2 = H(\text{Tx4} r_2);$
$s_1 = u_1 + x_1 h_1 \bmod (p-1);$	$s_2 = u_2 + x_2 h_2 \bmod (p-1);$

$\sigma_1 = (r_1, s_1).$ $\sigma_2 = (r_2, s_2).$

Alice sends signatures σ_1, σ_2 to IC.

After receiving σ_1 and σ_2 , and sums 4000 and 3000, IC performs the verification of these signatures according to (Eq.1).

IC has no idea that sums 4000 and 3000 transferred by Tx3 and Tx4 belongs to Alice represented by Addr.

IC accounts the sums 4000 and 3000 to Addr1 and Addr2.

Anonymity disclosing for Investment Company (IC) Schnorr-Aggregated-Signature Deanonymization in BlockChain

Agregated signature is used when signers are signing different messages.

It is realized by the special multiplication of all signatures.

Then the verification of aggregated signature is realized by having signers public keys,

2 components of aggregated and h-vaues of signed messages.

Alice has private key $\text{PrK}_A = x$ and public key $\text{PuK}_A = a = g^x \bmod p$, corresponding to Addr.

Alice has a certificate Cert_A issued by Certificate Authority (CA) on her $\text{PuK}_A = a$.

Alice must prove to IC that $\text{PuK}_1 = a_1 = g^{x_1} \bmod p$ and $\text{PuK}_2 = a_2 = g^{x_2} \bmod p$ together with Addr1 and Addr2 belongs to her.

This means that she must prove that she knows x_1 and x_2 corresponding to

$\text{PuK}_1 = a_1 = g^{x_1} \bmod p$ and $\text{PuK}_2 = a_2 = g^{x_2} \bmod p$.

To save the computation resources Alice does not proves the knowledge of every x_1 and x_2 .

Alice proves that she knows $x = x_1 + x_2 \bmod (p-1)$ instead since to guess x without the knowledge of x_1 and x_2 is infeasible.

According to the Birthday Paradox the probability Prob_2 to guess x_{12} when x_{12} is a sum mod p of two secret numbers x_1 and x_2 having $n=2048$ bits is negligible, i.e.

$$\text{Prob}_2 < 2^{-n/2}.$$

In the case of sum of k private keys the probability Prob_k satisfies inequality

$$\text{Prob}_k < 2^{-n/k}.$$

Alice realizes the Non-Interactive Zero knowledge Proof (NIZKP).

Alice computes aggregated signature σ_{12} of signatures σ_1, σ_2 by in a special way multiplying signatures σ_1, σ_2 .



$$\sigma_{12} = \sigma_1 * \sigma_2 = s_2 = (r_1, s_1) * (r_2, s_2) = (R_{12}, S_{12}).$$

$$R_{12} = r_1 * r_2 \bmod p = g^{u_1} * g^{u_2} \bmod p.$$

$$S_{12} = s_1 + s_2 \bmod (p-1) = [(s_1 = u_1 + x_1 h_1) + (s_2 = u_2 + x_2 h_2)] \bmod (p-1).$$

Alice signs aggregated signature σ_{12} with her private key x using Schnorr signature.

She computes h-value: $h' = H(a_1 || Addr1 || Tx3 || a_2 || Addr2 || Tx4 || a || Addr || Cert_A || \sigma_{12})$

$$u \leftarrow \text{randi}(p-1).$$

$$r_p = g^u \bmod p.$$

$$h'' = H(h' || r_p).$$

$$s_p = u + x h'' \bmod (p-1).$$

Alice signature on σ_{12} is $\sigma_P = (r_p, s_p)$.

Alice sends to IC: Tx3 and Tx4,
Addr1 and Addr2,
 a, a_1, a_2 and $Cert_A$,
 $\sigma_P = (r_p, s_p)$.

IC: computes aggregated signature σ_{12} .

IC: verifies aggregated signature σ_{12} using the following identity:

$$g^{S_{12}} \bmod p = R_{12} * \underbrace{(a_1)^{h_1}}_{V1} * \underbrace{(a_2)^{h_2}}_{V2} \bmod p. \quad (\text{Eq.2})$$

Compare it with a single Schnorr signature verification in (Eq. 1)

$$\underbrace{g^s}_{V1} \bmod p = \underbrace{r a^h}_{V2} \bmod p. \quad (\text{Eq.1})$$

Correctness:

$$g^{S_{12}} \bmod p = g^{(s_1 + s_2) \bmod (p-1)} \bmod p = g^{s_1 \bmod (p-1)} * g^{s_2 \bmod (p-1)} \bmod p = g^{(u_1 + x_1 h_1) \bmod (p-1)} * g^{(u_2 + x_2 h_2) \bmod (p-1)} \bmod p$$

$$\begin{aligned} &= r_1 * (a_1)^{h_1} * r_2 * (a_2)^{h_2} \bmod p = \\ &= r_1 * r_2 * (a_1)^{h_1} * (a_2)^{h_2} \bmod p = \\ &= R_{12} * a_1^{h_1} * a_2^{h_2} \bmod p. \end{aligned}$$

IC: computes h-value: $h' = H(a_1 || Addr1 || Tx3 || a_2 || Addr2 || Tx4 || a || Addr || Cert_A || \sigma_{12})$;
computes h-value: $h'' = H(h' || r_p)$.

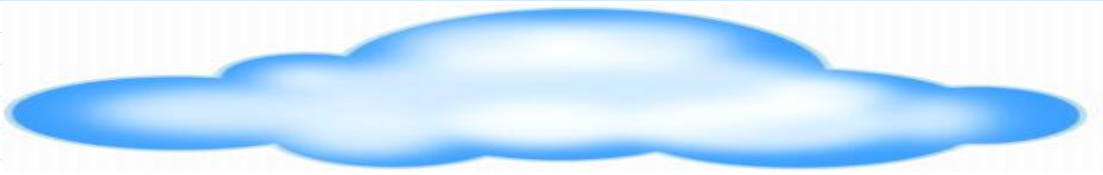
IC: verifies signature $\sigma_P = (r_p, s_p)$ on aggregated signature σ_{12} using the following identity:

IC: verifies signature $\sigma_P = (r_p, s_p)$ on aggregated signature σ_{12} using Alice public key a and the following identity:

$$\underbrace{g^{s_p}}_{V1} \bmod p = \underbrace{r_p}_{V2} a^{h''} \bmod p. \quad (\text{Analog of Eq.1})$$

If verification passes then IC registrates Alice funds, received from transactions Tx3 and Tx4 as Alice investment of the sum 7000 = 4000 + 3000.

Cryptography, Internet of Things (IoT), Blockchain, Cryptocurrency Business Value Chain



Producer



IoT



Wholesaler

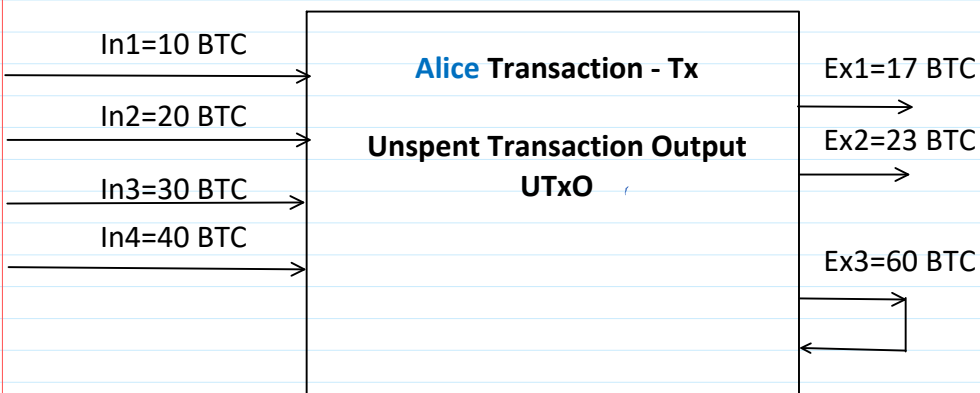


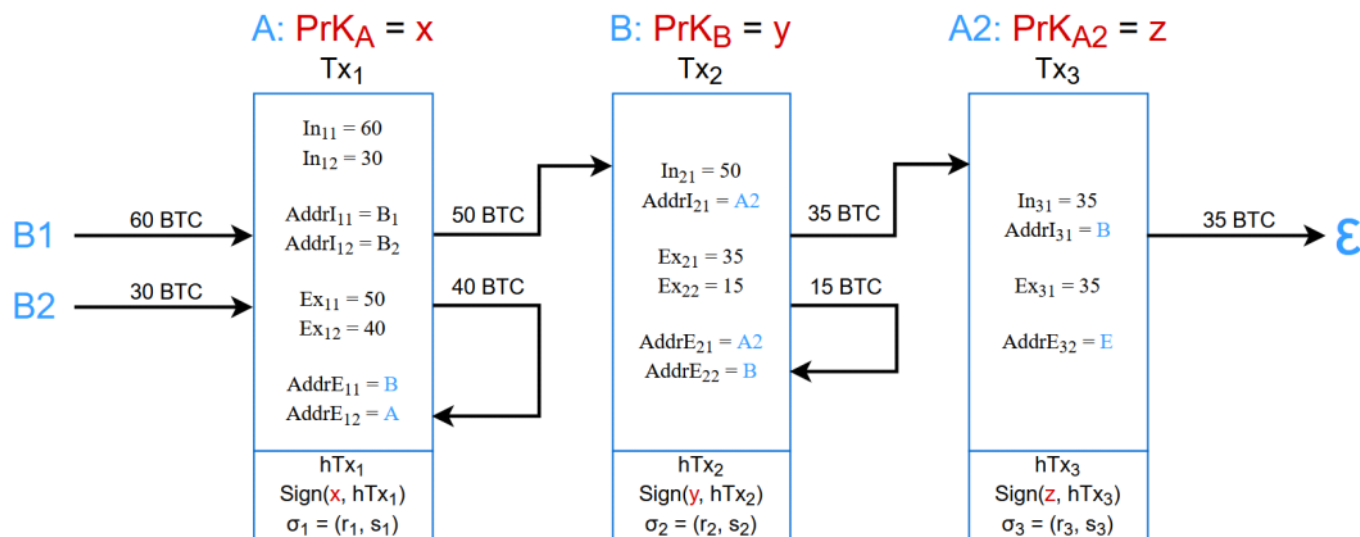
Retailer



Blockchain

Transactions and Tokens in Ethereum blockchain are controlled by Smart Contracts. Smart Contracts are controlled by *Events* that are initialized by Oracles created by IoT. Smart Contracts after receiving Oracles initiates Transactions.





Till this place