ElGamal Cryptosystem (CS)	Elliptic Curve Cryptosystem (ECC)
PP=(strongprime p, generator g); p=255996887; g=22; **Cyclic multiplicative group**: Z_p^*={1, 2, 3, …, p-1}; $\bullet_{\mathbf{mod}\ p}$, $:_{\mathbf{mod}\ p}$.	**PP**=(**EC secp256k1**; **BasePoint** or **Generator** G; prime p; param. a, b); Parameters a, b defines **EC** equation $y^2=x^3+ax+b$ **mod** p over **finite field** F_p={0, 1, 2, 3, …, p-1}; $+_{\mathbf{mod}\ p}$, $-_{\mathbf{mod}\ p}$, $\bullet_{\mathbf{mod}\ p}$, $:_{\mathbf{mod}\ p}$, (except division by 0)
PrK=x; >> x=randi(p-1); $p \sim 2^{2048}$; x~ 2^{2048}	**PrK**$_{ECC}$=z; >> z=randi(p-1). $p \sim 2^{256}$; z ~ 2^{256};
PuK=$a=g^x \bmod p$.	**PuK**$_{ECC}$=A=z$*G$.
Alice A: x=**1975596**; a=**210649132**;	**Alice** A: z=**…..**; $A=(x_A, y_A)$;

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

Anonymity in Blockchain
Ring signatures in ECDSA - Monero

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** Aaddress in Bitcoin.

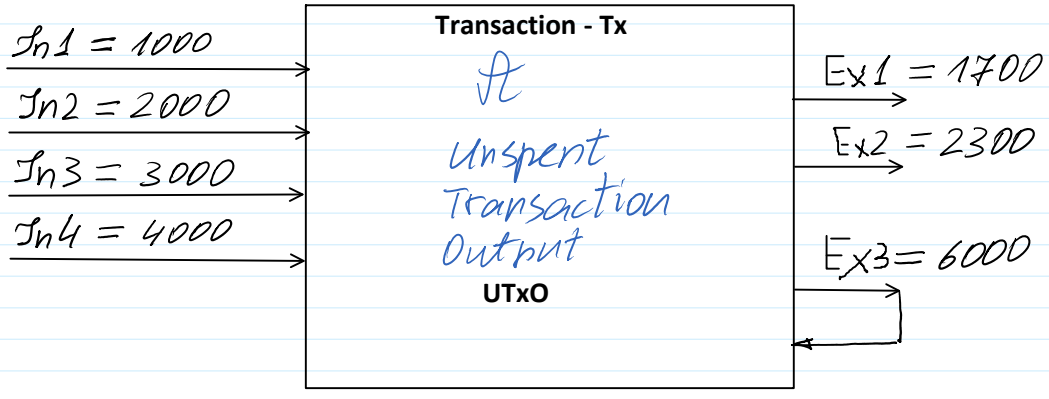
In Bitcoin and other Blockchains the Address is computed as a function of user's public key:
Addr$_A$ = **F**(**PuK**) and consist of several dozens of decimal numbers.

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.t Firm 6: 2300 Sa	2000 Sat
In4.	Client4: 4000 Sat	Out3. Firm 7:	6000 Sat
Total	**10 000 Sat**	**4000 Sat**	**6000 Sat**

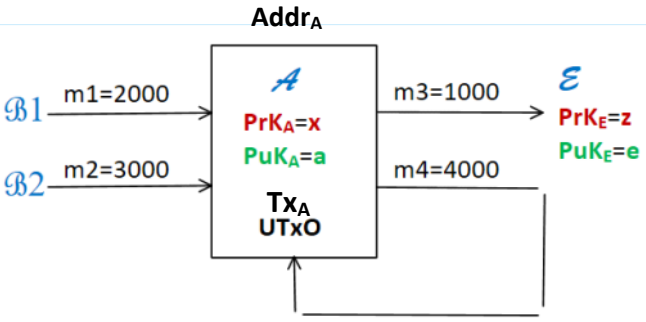
Sum of Inputs =
= Sum of Outputs
Divisibility

Unspent Transaction Output - **UTxO** paradigm



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

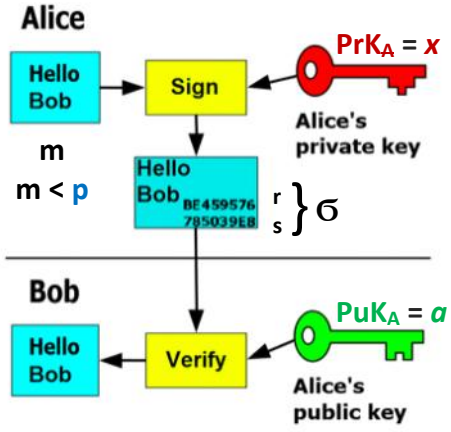
1. The address of Tx creator.
2. The sums of Incomes and addresses of senders.
3. The sums of Expenses and addresses of receivers.



Alice has a certificate **Cert$_A$** for her **PuK$_A$** = ***a*** = g^x **mod p**.

Schnorr Signature

In the case of Schnorr cryptosystem our simulation is performed with Public Parameters:
PP = (***p***, ***g***); **p=268435019; g=2**; p=int64(268435019)
By having **PP** private key **PrK** and public key **PuK** are generated:
PrK = ***x*** <-- randi(p-1)
PuK = ***a*** = g^x **mod** ***p***.



u <-- randi(***p***-1).
r = g^u **mod** ***p***.
h = **H(*M*||*r*)**. >> con=concat(M,r)
>> h=hd28(con)
s = ***u***+***xh*** **mod (*p*-1)**. (*) >> s=mod(u+x*h,p-1)
Alice's signature on ***h*** is **σ = (*r*, *s*)**.

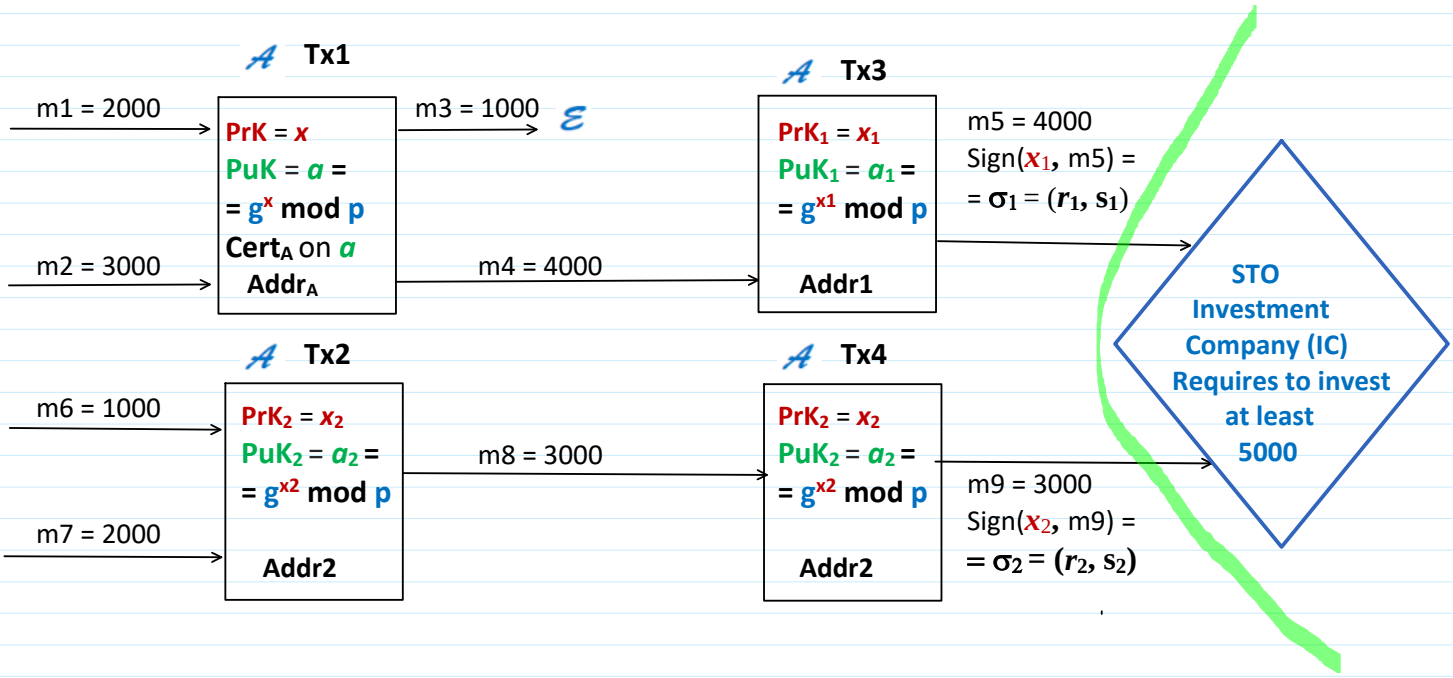
Notice that it is infeasible to find ***x*** from (*), when
s and ***h*** are given, since there is 1 equation (*) and 2 unknowns ***u*** and ***x***.

Signature is valid if: g^s **mod** ***p*** = ra^h **mod** ***p***. (Eq.1)
V1 V2

Anonymization

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 {**PrK$_i$** = ***x$_i$***} and a set of public keys {**PuK$_i$** = ***a$_i$***}, where *i*=1, 2, …, 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** must prove that her transactions are authentic (i.e. are created by herself) by proving that **PuK**=***a*** belongs to her, e.g. using Certificate issued by Certificate Authority for **PuK**=***a***, but at the same time she remains **anonymous** for other part of the **Net**.



In Monero blockchain for anonymization Alice is using **Ring Signature**, instead procedure presented above.
It is interresting to compare the realization effectivity of procedure presented above and procedure based on Ring Signature.

Compare realization effectivity of DEF Schnorr multisignature with ECC ring signature computing the number of
Discrete Exponent Function Operations - DEFO: $a = g^u \bmod p$
Elliptic Curve Cryptography Operations - ECCO: EC point multiplication by integer $z*G = P$.

Schnorr-Multi-Signature
De-anonymization in BlockChain

Anonymous Group of Signers (**GoS**) must sign on different transactions with different private keys.
In this case the group consist of 2 anonymous addresses **Addr1** and **Addr2** belonging to **Alice**.
Let the **GoS** is: $\{S_1; S_2\}$.
All members of **GoS** have their private and public keys:

S_1;	S_2;
$PrK_1=x_1$, $PuK_1=a_1$;	$PrK_2=x_2$, $PuK_2=a_2$;
u_1 <-- randi(p-1);	u_2 <-- randi(p-1);
$r_1=g^{u1} \bmod p$;	$r_2=g^{u2} \bmod p$;
$h_1=H(Tx3 \| r_1)$;	$h_2=H(Tx4 \| r_2)$;
$s_1=u_1 + x_1h_1 \bmod (p-1)$;	$s_2=u_2 + x_2h_2 \bmod (p-1)$;
$\sigma_1 = (r_1, s_1)$.	$\sigma_2 = (r_2, s_2)$.

How to join signatures $\sigma_1 = (r_1, s_1)$ and $\sigma_2 = (r_2, s_2)$ to the one signature $\sigma_P = (r_P, s_P)$.
Schnorr multisignature solves this problem.
Individual Schnorr signatures are multiplied by the special multiplication operation.

$\sigma_{12} = \sigma_1 * \sigma_2 = (r_1, s_1) * (r_2, s_2) = (R_{12}, S_{12})$.
$R_{12} = r_1 * r_2 \bmod p = g^{u1} * g^{u2} \bmod p = g^{u1+u2 \bmod (p-1)} \bmod p$.
$S_{12} = s_1 + s_2 \bmod (p-1) = [(s_1=u_1 + x_1h_1) + (s_2=u_2 + x_2h_2)] \bmod (p-1) = [\, u_1 + x_1h_1 + u_2 + x_2h_2 \,] \bmod (p-1)$.

GoS signature verification:

$g^{S_{12}} \bmod p = R_{12} * (a_1)^{h1} * (a_2)^{h2} \bmod p$. (Eq.2)

V1 V2

Correctness:

$g^{S_{12}} \bmod p = g^{(s1 + s2) \bmod (p-1)} \bmod p = g^{s1 \bmod (p-1)} * g^{s2 \bmod (p-1)} \bmod p = g^{(u1 + x1*h1) \bmod (p-1)} * g^{(u2 + x2*h2) \bmod (p-1)} \bmod p =$

$= r_1 * (a_1)^{h1} * r_2 * (a_2)^{h2} \bmod p =$

$= r_1 * r_2 * (a_1)^{h1} * (a_2)^{h2} \bmod p =$

$= R12 * a_1^{h1} * a_2^{h2} \bmod p.$

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

$g^s \bmod p = r a^h \bmod p$. (Eq.1)

V1 V2

But to form the Schnorr multisignature may everyone since $\sigma_1 = (r_1, s_1)$ and $\sigma_2 = (r_2, s_2)$ are available to anybody.

Alice to prove her identity against Investment Company (IC) it is required to sign the signature $\sigma_{12} = \sigma_1 * \sigma_2 = (r_1, s_1) * (r_2, s_2) = (R_{12}, S_{12})$ with her **PrK**=x:

u <-- randi(p-1);

$r = g^u \bmod p$;

$h = H(\sigma 12 \,||r)$;

$s = u + xh \bmod (p-1)$;

By obtaining signature $\sigma = (r, s)$.

The signature $\sigma = (r, s)$ is verified by **PuK1**=a using the verification equation from above:

$g^s \bmod p = r a^h \bmod p$.

V1 V2