Exam is in 6 November at 9:00
In 29-th of November the Oral presentation of Course work must be presented.
During the exam the remining 4 problems must be solved in www.imimsociety.net and Oral presentation of Course work must be presented fo those who did not present in 29-th of November.

The bank is entirely eliminated from the blockchai cryptocurrency, namely bitcoin.
This changes the nature of the currency considerably.
It means that there is no longer any single organization in charge of the currency.
And when you think about the enormous power a central bank has – control over the money supply.

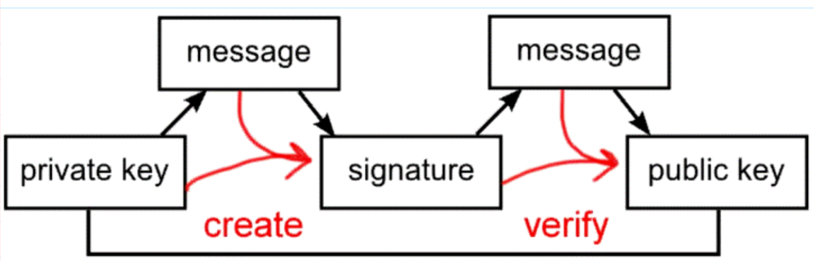
1 mBTC = 10^{-3} BTC
1 μBTC = 10^{-6} BTC
1 Satoshi = 10^{-8} BTC
1 Sat

For eSignature realization bitcoin uses ECDSA key pair.

A 256-bit **private key** - **PrK** consisting of . The private key is needed to sign a transaction and thus transfer (spend) bitcoins. ;,\.;\/'

A 512-bit **public key** - **PuK** computed from the **PrK**. private key.
This public key is used to verify the signature on a transaction.
Inconveniently, the Bitcoin protocol adds a prefix of 04 to the public key.
The public key is not revealed until a transaction is signed, unlike most systems where the public key is made public.

From <http://www.righto.com/2014/02/bitcoins-hard-way-using-raw-bitcoin.html>



A : PrK , PuK , m - message to be signed

$$\mathcal{S} = (r, s) = Sig(PrK, m) \Rightarrow Ver(PuK, \mathcal{S}, m) = \begin{cases} 1, & Ok \\ 0, & No \end{cases}$$

A : $\xrightarrow[PuK]{m,\ \mathcal{S}}$ B

Signing and verification math.

The elliptic curve E over a finite field Fp, with most popular choice being prime fields GF(p) where all arithmetic is performed modulo a prime p, is set of all pairs (x, y) ∈ Fp which fulfill

The elliptic curve E over a finite field Fp, with most popular choice being prime fields GF(p) where all arithmetic is performed modulo a prime p, is set of all pairs (x, y) ε Fp which fulfill E:

$$y^2 \equiv x^3 + a.x + b \bmod p$$

together with an imaginary point of infinity O , where p > 3 is prime, and a, b ε Fp.
The cryptographic signatures used in Bitcoin are ECDSA signatures and use the curve **secp256k1** defined over Fp where p = $2^{256} - 2^{32} - 977$ which has a 256-bit prime order.
This choice deviates from **NIST** recommended **FIPS 186-4** standard [30] in that the curve coefficients are different from the **NIST** recommended standard to speed up scalar multiplication as well as Pollard's rho algorithm for computing discrete logarithms [31].

Omited

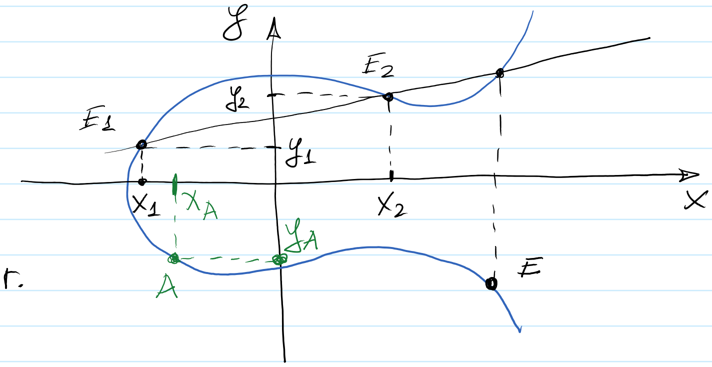
EC group : additive

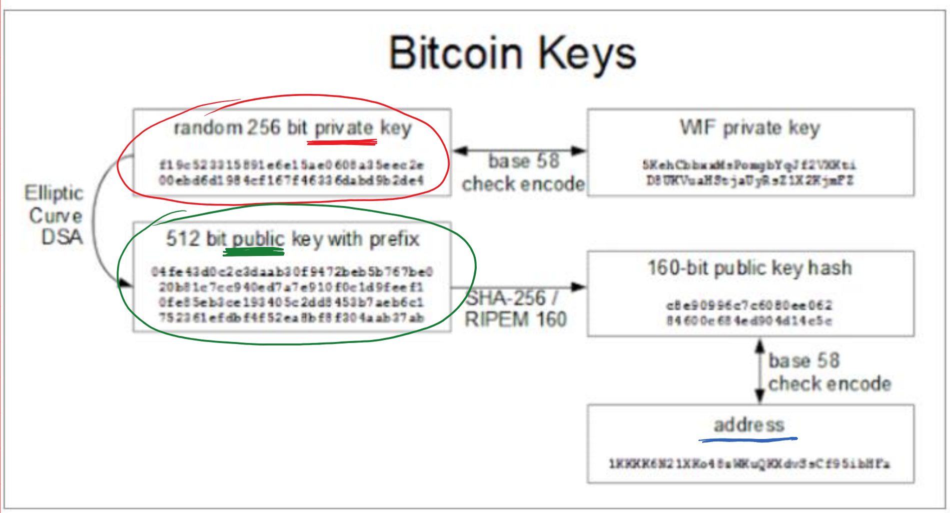
$E = E_1 + E_2$

$E_1 = (x_1, y_1)$, $E_2 = (x_2, y_2)$

Mult. Gr. Z_p^* EC Gr.

$a = g^x \bmod p$ $A = x G$





1) Anonymous CC
2) IoT CC

Bitcoin users address

The next step is to generate the Bitcoin address that is shared with others.
Since the 512-bit public key is inconveniently large, it is hashed down to 160 bits using the SHA-256 and RIPEMD hash algorithms.[9]
The key is then encoded in ASCII using Bitcoin's custom Base58Check encoding.[10] yielding Bitcoin address.
The resulting address, such as 1KKKK6N21XKo48zWKuQKXdvSsCf95ibHFa, is the address people publish in order to receive bitcoins.
Note that you cannot determine the public key or the private key from the address.
If you lose your private key (for instance by throwing out your hard drive), your bitcoins are lost forever.

Omited

Note that you cannot determine the public key or the private key from the address.
If you lose your private key (for instance by throwing out your hard drive), your bitcoins are lost forever.
Finally, the Wallet Interchange Format key (WIF) is used to add a private key to your client wallet software.
This is simply a Base58Check encoding of the private key into ASCII, which is easily reversed to obtain the 256-bit private key.

Omitted

To summarize, there are three types of keys: the private key, the public key, and the Bitcoin address the latter you see published.
Bitcoin address is the hash of the public key, and they are represented externally in ASCII using Base58Check encoding.
The private key is the important key, since it is required to access the bitcoins and the other keys can be generated from it.

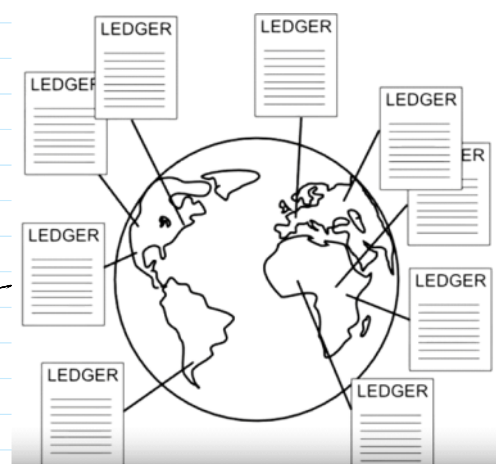
Given ECDSA public-key K, Bitcoin address is generated using the cryptographic hash functions SHA-256 and RIPEMD-160 [32]: HASH160 = RIPEMD-160(SHA-256(K)). Bitcoin address is computed directly from the HASH160 value as illustrated below in Figure 3, where base58 is a binary-to-text encoding scheme [33]:

base58 (0x00 ‖ HASH160 ‖ SHA-256(SHA-256(0x00 ‖ HASH160))/2 ^{224} ⌋)

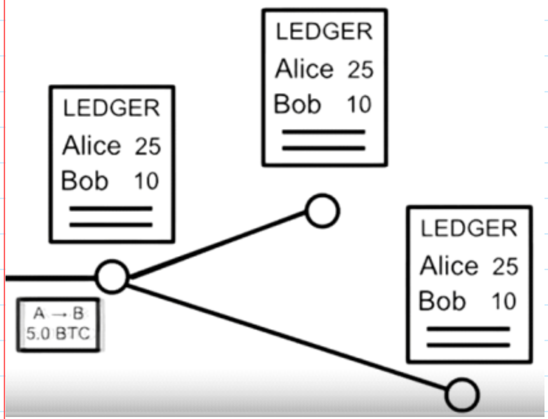
Figure3. How Bitcoin Address is Computed Using ECDSA Algorithm.

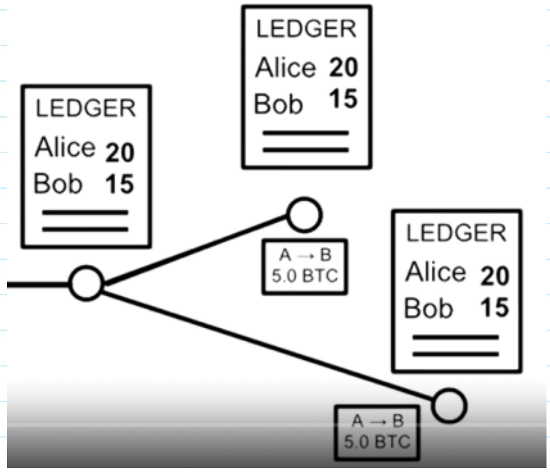
30 http://csrc.nist.gov/groups/STM/cavp/documents/dss2/dsa2vs.pdf
31 http://research.microsoft.com/apps/pubs/default.aspx?id=204914
32 http://homes.esat.kuleuven.be/~bosselae/ripemd160.html
33 https://en.bitcoin.it/wiki/Base58Check_encoding

Everyone using bitcoincoin keeps a complete record of which bitcoins belong to which person.
You can think of this as a shared public ledger showing all bitcoin transactions.
We’ll call this ledger the ***block chain***, since that’s what the complete record will be called in bitcoin, once we get to it.

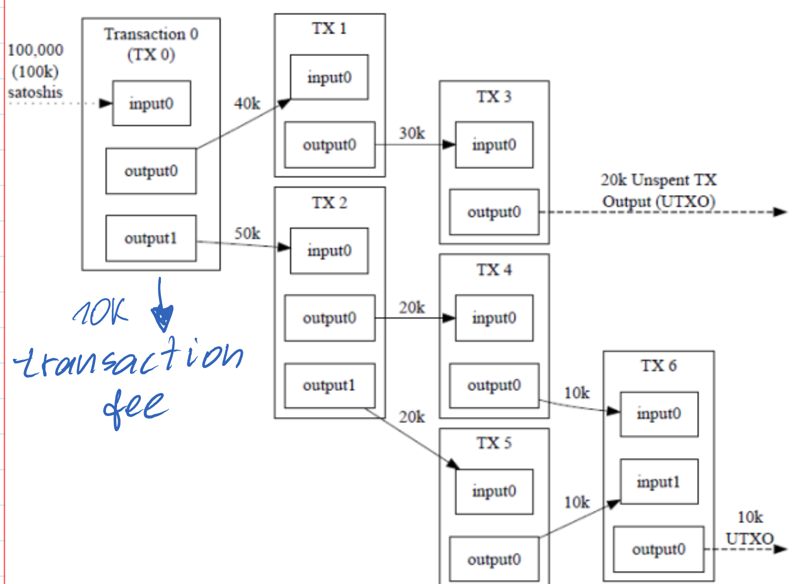


Distributed Ledger Technology – DLT





A ***transaction*** is the basic operation in the Bitcoin system.
You might expect that a transaction simply moves some bitcoins from one address to another address, but it's more complicated than that.
A Bitcoin transaction moves bitcoins between one or more ***inputs*** and ***outputs***.
Each input is a transaction and address supplying bitcoins.
Each output is an address receiving bitcoin, along with the amount of bitcoins going to that address.

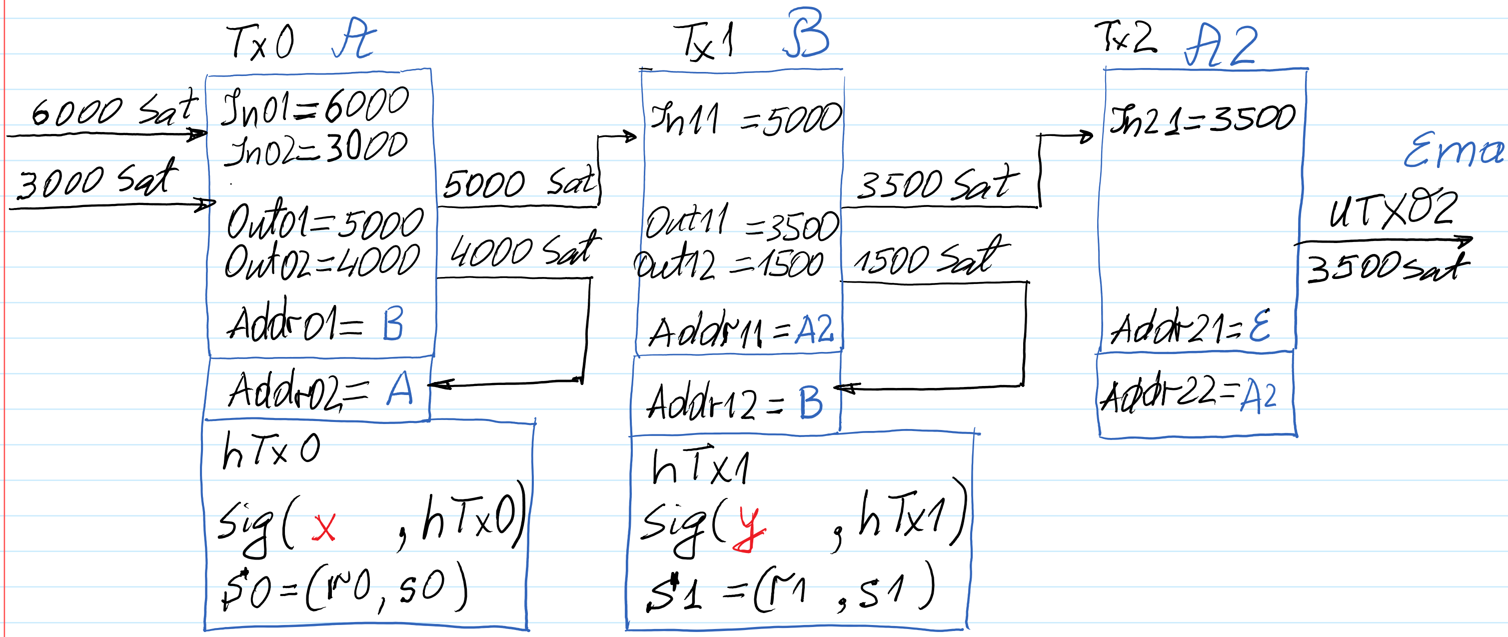


10K transaction fee

Unspent Transactions Output
UTXO

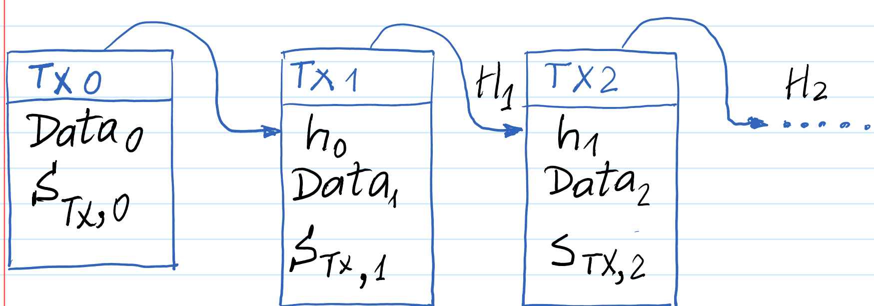
if the inputs exceed the value of the outputs, any difference in value may be claimed as a transaction fee by the Bitcoin miner who creates the block containing that transaction.
For example, in the illustration above, each transaction spends 10,000 satoshis fewer than it receives from its combined inputs, effectively paying a 10,000 satoshi transaction fee.

Tx0 A   Tx1 B   Tx2 A2



Transaction template:
TxN = 'TxN:InN1=...||InN2=...||OutN1=...||OutN2=...||Rec1=...||Rec2=...'
Transactions:
Tx0 = 'Tx0:In01=6000||In02=3000||Out01=5000||Out02=4000||Rec1=B||Rec2=A'
Tx1='Tx1:In11=5000||Out11=3500||Out12=1500||Rec1=A2||Rec2=B'
Tx2='Tx2:In21=3500||Out21=3500||Out22=0||Rec1=E||Rec2=0'

TSA fraud --> Prevention using Blockchain



$$h_0 = H(Data_0 \| S_{Tx,0})$$
$$h_1 = H(H_0 \| Data_1 \| S_{Tx,1})$$
$$h_2 = H(H_2 \| Data_2 \| S_{Tx,2})$$

1) The h-value for signature creation we denote by letter h:
e.g. to sign a transaction Tx1 it is required to compute
$h_1 = H(Data_1) \Longrightarrow S_{Tx,1} = Sig(PrK_1, h_1) = (r_1, s_1)$

2) The h-value used for the next transaction Tx2 to be included in blockchain is denoted by capital letter H_2
e.g. to create a new transaction after Tx1 it is required

in Blockchain is denoted by capital letter H_2

e.g. to create a new transaction after Tx1 it is required to include h-value of previous transaction Tx1

$H_1 = H(H_1 \| Data_1 \| S_{Tx_1})$.

A: **$Tx, n+1$**

H_n

$Data_{n+1}$

$H(H_n \| DT_{n+1}) = h_{n+1}$

$Sig(PrK_A, h_{n+1}) = S_{n+1}$

B:

$Ver(PuK_A, S, h_{n+1}) = True$

For B's transac. creation of No n+2:

$h_{n+2} = H(Data_{n+2})$

$S_{n+2} = Sig(PrK_B, h_{n+2})$

Block mining: requires to compute h-value of the Block containing N leading zeroes in this h-value.

Let SHA-256 h-value is used ⇒ 64 hexadecimal numbers.

Then if Difficulty Target is equal to $N=18$ hex numbers

$$h_0 = H(\text{BlocData} \,||\, \text{nonce}) \neq \underbrace{000000000000000000}_{18}\text{A9DE}\ldots 2_h.$$

$$h_1 = H(\text{BlocData} \,||\, \text{nonce} := \text{nonce}+1) =$$

— — — — — — — —

$$h_N = \underbrace{00\ldots\ldots0}_{18}\,\text{D}\ldots\ldots 5. \qquad N \sim 2^{64}$$

Till this place