Summer Project 2018-19 Blockchain Technology

-Neha Jain
3rd year cse

Contents

- Blockchain
  - -Bitcoin blockchain
  - -Ethereum blockchain
- Lottery protocol using Solidity

Blockchain

- Allows digital information to be distributed but not copied.
- Blockchain is the technology behing Bitcoin.
- It is a digital ledger in which transactions made in bitcoin or another cryptocurrency are recorded chronologically and publicly.

Issues in existing system

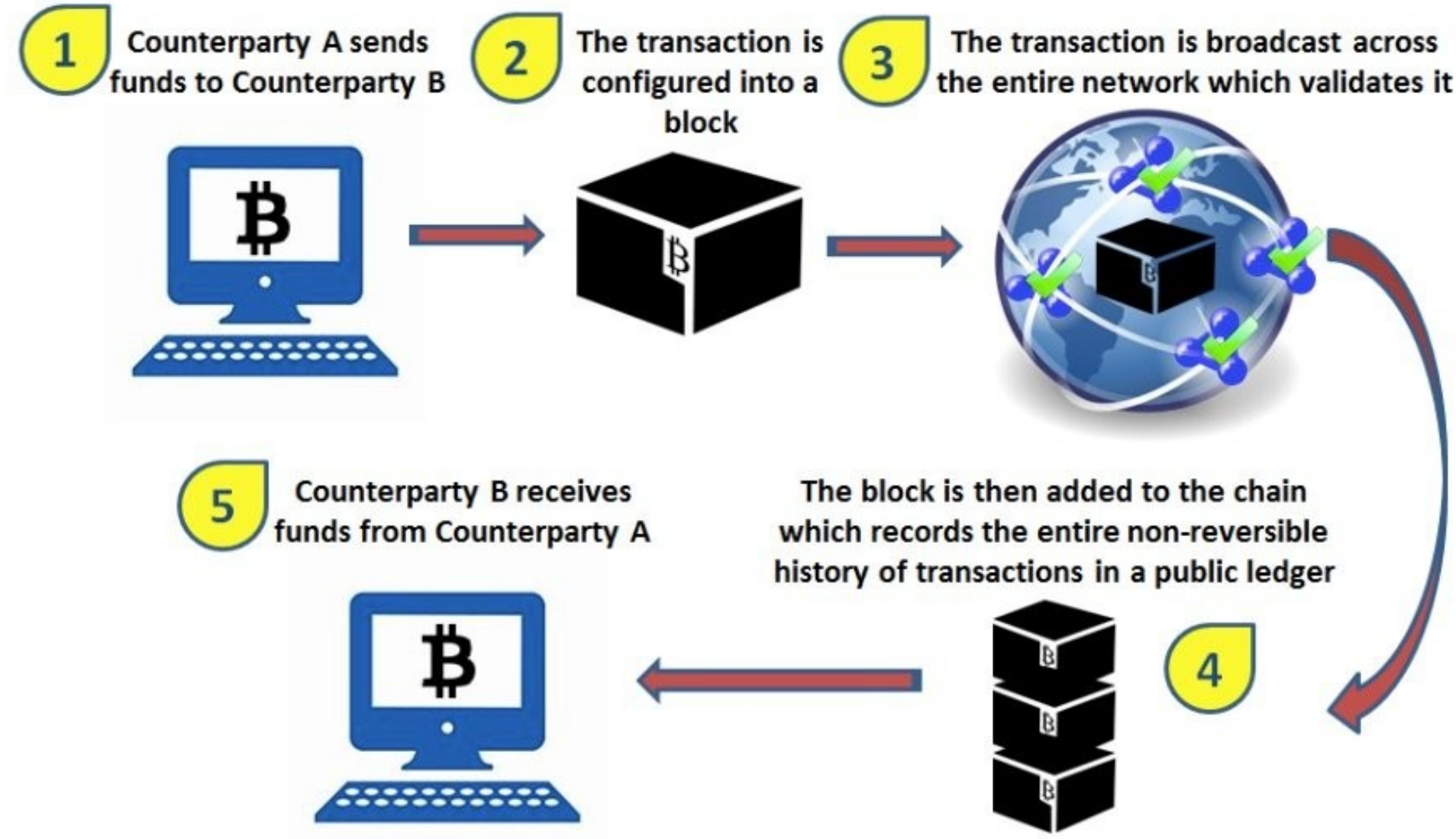
- Transaction fees involved in banks.
- Banks are a centralized system.
- Double Spending.

  Two transaction going throught at the same time.

  Happens in current banking system.

  Digital signature can be falsified.

Exhibit 1: The Blockchain is a distributed, public ledger, most commonly known as the core underlying technology for Bitcoin



Source: Goldman Sachs Global Investment Research.

Bitcoin

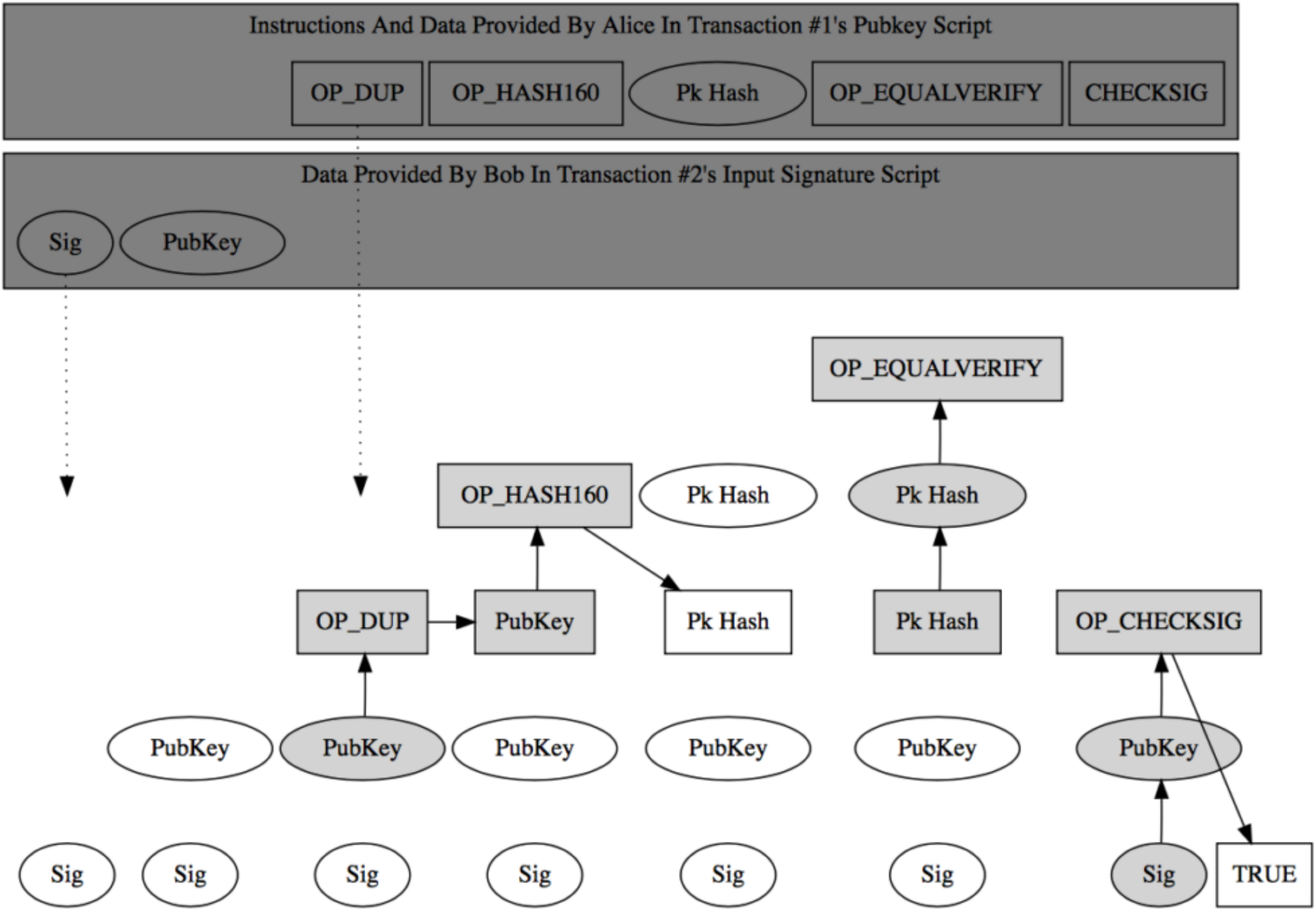
- First decentralized digital currency.
- Ledger system is completely electronic and highly secure.
- Double spending not possible.
- Currencies are controlled by central authority,but with Bitcoin system got decentralized and distributed to everyone.
- Cannot be hacked.

Transactions in Bitcoin.

A transacion is 'issued' from wallet. It is signed with the private key. To produce a digital signature.

-Transactions consists of inputs and outputs.

-Bitcoin wallet doesn't actually hold the bitcoin, bitcoin address has record of all transactions.



Evaluation Stack Over Time During Succesful P2PKH Script Validation

Transactions ...cont

scriptSig appears in the Input Script.

scriptSig = <sig> <pubKey>

Here the public key is of sender.

scriptPubKey

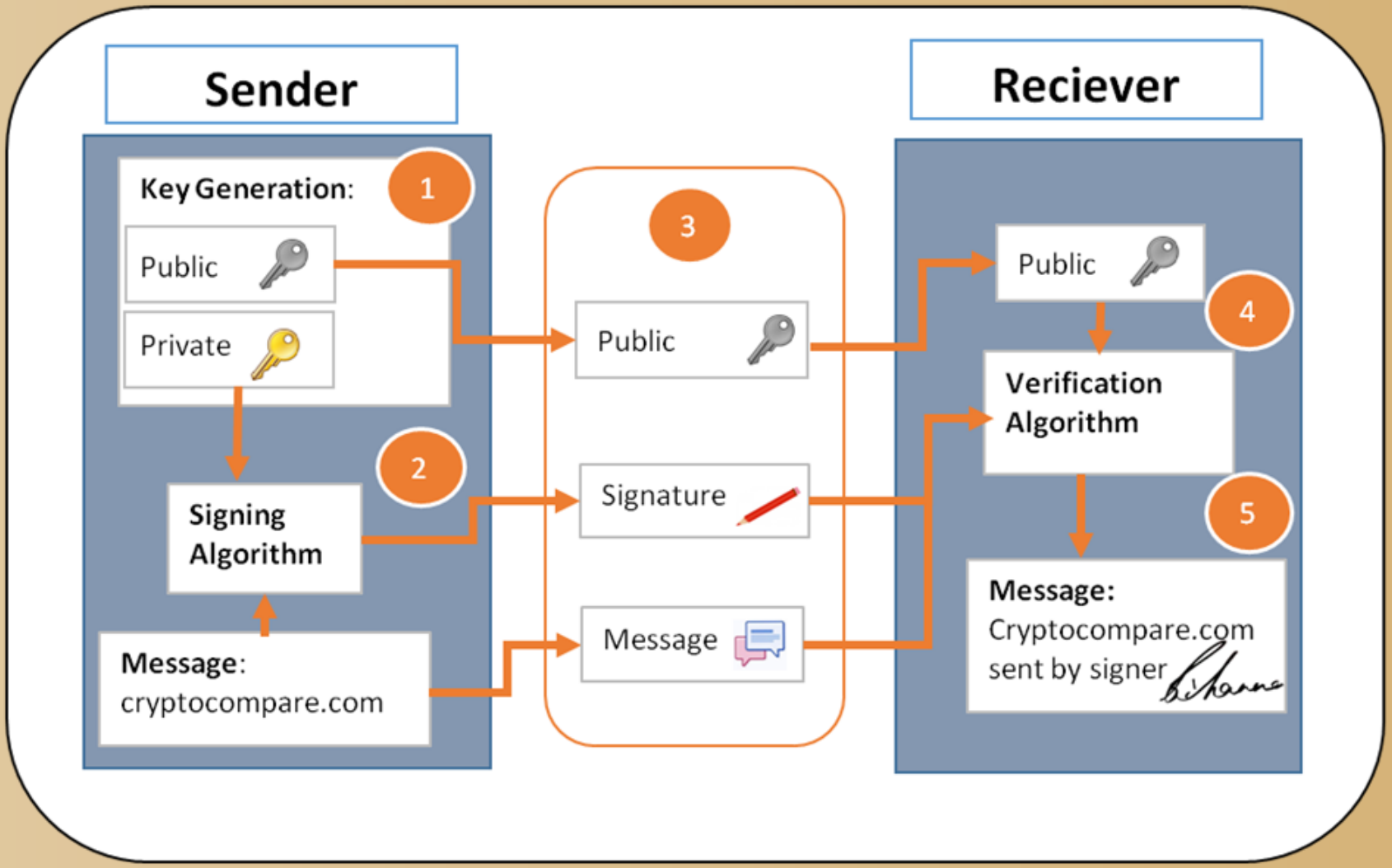
scriptPubKey = OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

pubKeyHash = Hash of the Public Key of Recipient

scriptPubkey-- It is created by previous transaction so that only the authorized person can take the outut as his input.

The input's scriptSig and the referenced output's scriptPubKey are evaluated (in that order), with scriptPubKey using the values left on the stack by scriptSig.

Verification process



Mining

Bitcoin mining is the process of adding transaction records to Bitcoin's public ledger of past transactions or blockchain. This ledger of past transactions is called the block chain as it is a chain of blocks.

Mining

- Some nodes are mining nodes. Block has all transactions that reached the particular mining node that created it at the same time. There can be differences then in the block that each mining node wants to broadcast.
- If two nodes broadcast different versions of the next block simultaneously, some

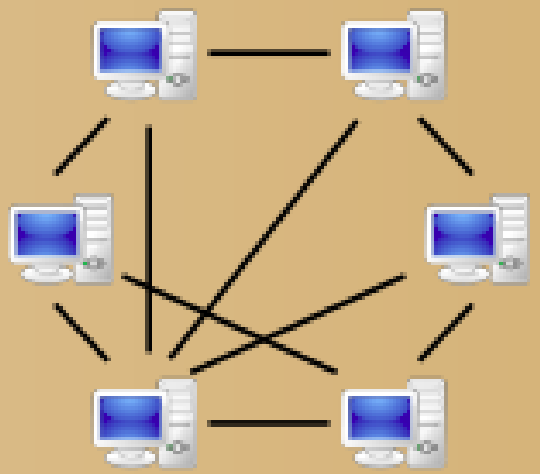
  nodes may receive one or the other first. In that case, they work on the first one they received,

  but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other

  branch will then switch to the longer one.

P2P network

- A peer-to-peer (P2P) network in which interconnected nodes ("peers") share resources amongst each other without the use of a centralized administrative system



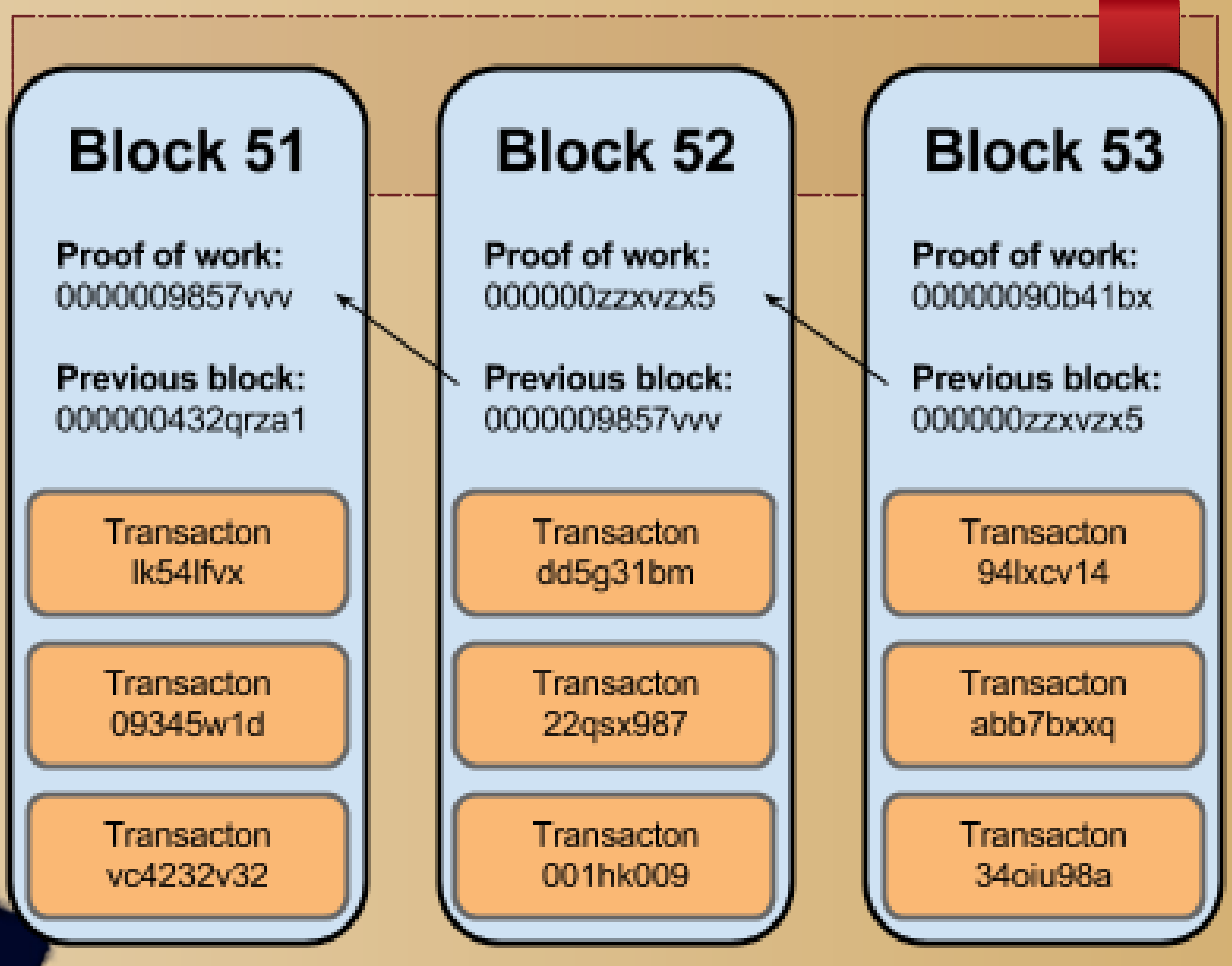
Blockchain

- . Hash function is used to check if the transaction has been tampered with or not. Hash function produces different output even for small change.

- Each block contains as part of its data a hash of previous block.

  This is how the 'blockchain' is generated.

  If you change even a small part of previous block. Next block would have to change. By the time another one would have come. Hence its difficult to tamper with.


Block 51
Proof of work:
0000009857vvv
Previous block:
000000432qrza1
Transacton
lk54lfvx
Transacton
09345w1d
Transacton
vc4232v32
Block 52
Proof of work:
000000zzxvzx5
Previous block:
0000009857vvv
Transacton
dd5g31bm
Transacton
22qsx987
Transacton
001hk009
Block 53
Proof of work:
00000090b41bx
Previous block:
000000zzxvzx5
Transacton
94lxcv14
Transacton
abb7bxxq
Transacton
34oiu98a

Structure of a Block

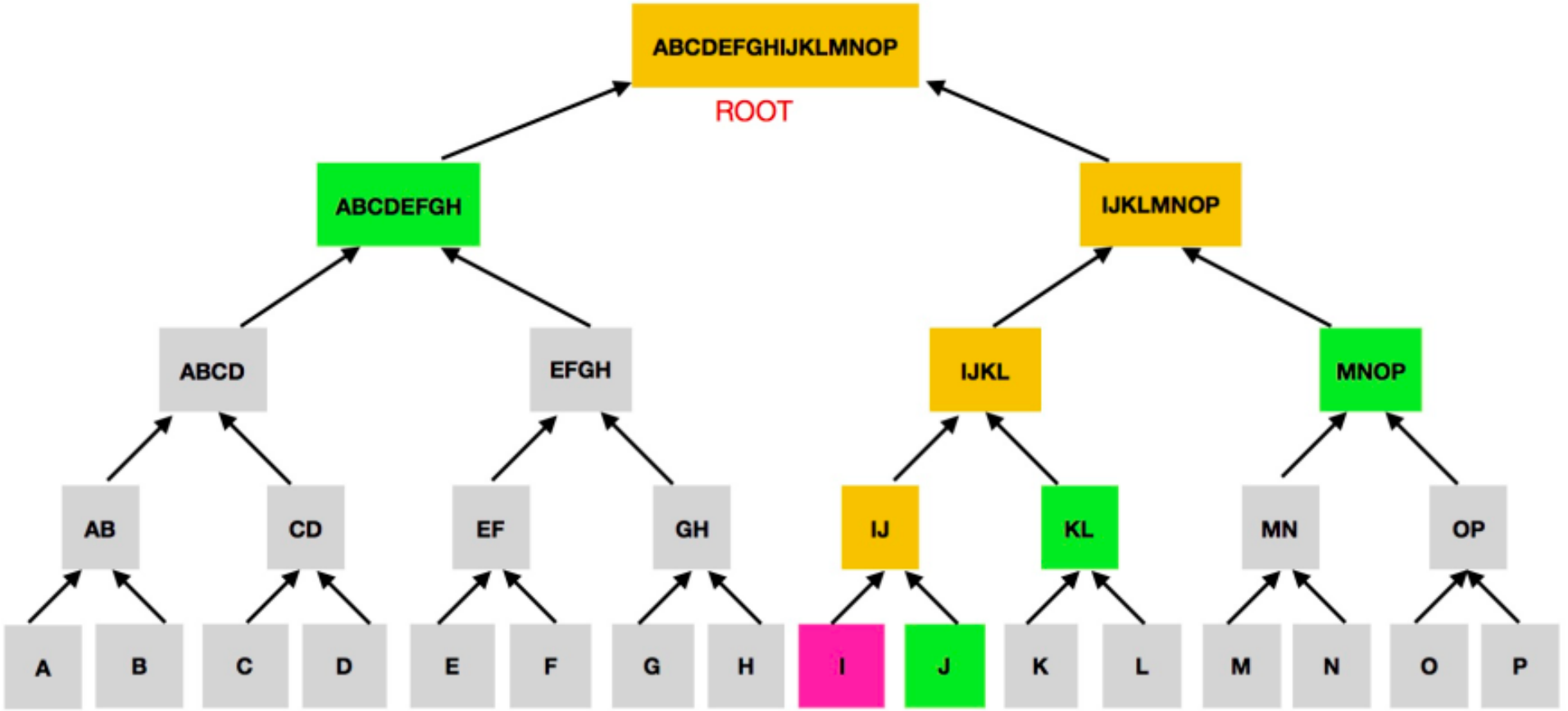
- A block is a container data structure. In the Bitcoin world, a block contains more than 500 transactions on average.
- block is composed of a header and a long list of transactions.
- Header-
- The previous block hash. Remember that in a blockchain, every block is inherits from the previous block because we use the previous block's hash to create the new block's hash. For every block N, we feed it the hash of the block N-1.
- Mining competition. For a block to be part of the blockchain, it needs to be given a valid hash. This contains the timestamp, the nonce and the difficulty.
- The third part is a merkle tree root. This is a data structure to summarize the transactions in the block.

Structure of a Block:cont.

- To identify a block, you have a cryptographic hash, a digital signature if you will. This is created by hashing the block header twice with the SHA256 algorithm.
- Remember that we used the second hash to create the first. Every block uses the previous block's hash to construct its own hash. The block hash is a unique identifier. You won't find two blocks with the same hash.
- The other way to identify a specific block is the block height. The is the position of the block in the blockchain.

Merkel Tree

- The transactions in a block are contained in a structure called a merkle tree or binary hash tree.
- A merkle tree is constructed by recursively hashing pairs of nodes (in this case, transactions), until there is only one hash, called the root or merkle root. If we stay in the Bitcoin world, the cryptographic hash algorithm used is SHA256. This is applied twice each time.


ABCDEFGHIJKLMNOP
ROOT
ABCDEFGH
IJKLMNOP
ABCD
EFGH
IJKL
MNOP
AB
CD
EF
GH
IJ
KL
MN
OP
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P

NONCE

The "nonce" in a bitcoin block is a 32-bit (4-byte) field whose value is adjusted by miners so that the hash of the block will be less than or equal to the current target of the network. The rest of the fields may not be changed, as they have a defined meaning.

What does a secure network mean?

When you're talking about security in respect to Bitcoin (and its surrounding network), you need to accomplish the following goals:

Coins cannot be double spent

Balances of who owns which coins must be accurate

It is very difficult to "steal" someone's coins using a network attack

1.PREVENTING DOUBLE SPEND

The network's main innovation in preventing a double spend is the distributed ledger (the blockchain.) By broadcasting all transactions to all nodes and waiting for those nodes to confirm a certain number of times, you're allowing the network to self-police. It might be easy to compromise a single node, but it'd be difficult to compromise enough nodes and have them all allow a double spend. This is where you hear fears of a "51% attack" - in theory, if you had 51% of the computing power that verifies transactions, you could allow double spends.

2.Maintaining Balances

A key part of Bitcoin's security relies on the concept of hash functions. A hash function will take a unique input, and from that input produce a unique output. If I ran the number 1 through a fictional hash function, it would always produce something like FFTA9. The Blockchain uses this to secure transactions. A group of these transactions are wrapped into a block, and the hash of those transactions is broadcast. If someone wanted to remove a transaction or insert a fake one, they'd somehow have to find a way to do that while outputting the same hash. While in theory this could be possible, it is extremely unlikely and would require more computing power than exists today.

3.Stealing Coins

The network broadcasts public addresses, which are derived from hashing a public key (that pairs with a single private key.) Only people with the private key can "sign" for coins to be spent, so if you wanted to get someone's coins, you'd need to get the private key from the public data. In most cases, this is impossible: each transaction uses random values to hide this. However, you might've heard of some attacks where it WAS possible to decode the private key. If you use a weak random number source, with enough data, you could get the private key for a given public key. That's why a major initiate in the Bitcoin community is to use standards-backed, open source and audited libraries for your cryptography and randomness. With the proper libraries, it becomes mathematically infeasible to steal someone's private key.

Ethereum Blockchain

- Very similar to bitcoin blockchain,shared record of entire transaction history,every node stores a copy.
- Every node stores the most recent state of each smart contract,in addition to all ether transaction.
- For each application,network needs to keep track of ‘state’, or current info of all these apllications.
- Ethereum doesn’ use UTXO, it uses accounts.

CONT..

Ethereum Virtual Machine

Contracts written in some language are compiled into ‘bytecode’ which ‘ethereum virtual machine’ can read and execute.

- Ethereum node is any device running the ethereum protocol.
- When connect to Ethereum protocol,we are on Ethereum blockchain network and running a node.
- A node can:-
  - Can mine blocks
  - Send transactions
  - Deploy smart contracts

What is GAS?

- Gas is ethereum’s metering scheme and it accounts for bandwidth used, cost of storage and cost of computation on Ethereum blockchain.
- Gas is used because it measure amount of computational work as th ether prices are volatile.
- Gas price-amount of ether you are willing to pay for every unit of gas.
- Gas limit- maximum amount of gas you’re willing to pay on a particular transaction.

Setting up an Ethereum testnet

STEP-1

Installation

sudo yum install golang -y

- git clone https://github.com/ethereum/go-ethereum
- cd go-ethereum
- git tag
- git checkout tags/v1.8.8 -b EdurekaEthereumV1.8.8
- git branch
- make all
- git checkout master
- make geth
- cd ~/go-ethereum

STEP-2

Creating a Genesis file

example:-

vi genesis.json

```
{
    "config": {
        "chainId": 13,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
    "difficulty": "200000000",
    "gasLimit": "2100000",
    "alloc": {
        "15f7d46274d10a0a479fc7b9cf872d0ff761547e": { "balance": "100000000" }
    }
}
```

STEP-3

Starting the testnet

```
~/go-ethereum/build/bin/geth --datadir
~/ethereum/net3 init genesis/genesis.json
```

```
~/go-ethereum/build/bin/geth --datadir
~/ethereum/net3/ --networkid 3 console
```

Cont...

After this testnet will be set.

Now you can make account.

> personal.newAccount()

Passphrase: 123456

"0x5fec0aeba91ff6e63d954915476bc15b223e1130"
--account address

Algorithm for multiparty lottery protocol

1.One node calls setup function- which sets up the refund amount,lottery amount, open time ,no. Of players and time limit for registering.

2.All the players call the register function with argument commit.(hash of choice)

While calling commit they need to deposit the refund amount which would be given to each player if they are not able to provide the corresponding choice.

Algorithm: cont.

3.Refund amount will be given to n-1 players so. So the refund money has to be (n-1)*refund_money.

If the choice is provided then this money will be paid back to the player

else refund_money will be given to each player.

Algorithm: cont.

4.After time limit for register is over Open will be called by each player with the required choice and the lottery_money.

If player enters the correct choice store the address in rightchoice and add the amount to tournament pot and return the refund amount entered at the time of register.

else enter the player in wrongchoice and return the lottery money amount.

Algorithm: cont.

5.Any player can call the Refund amount function which pays back all the players who entered correct choice with the refund money deposited by those who din’t.

6.Any player can they call compute winner function which computes the winner randomly and transfers the tournamet pot.

THANKYOU!