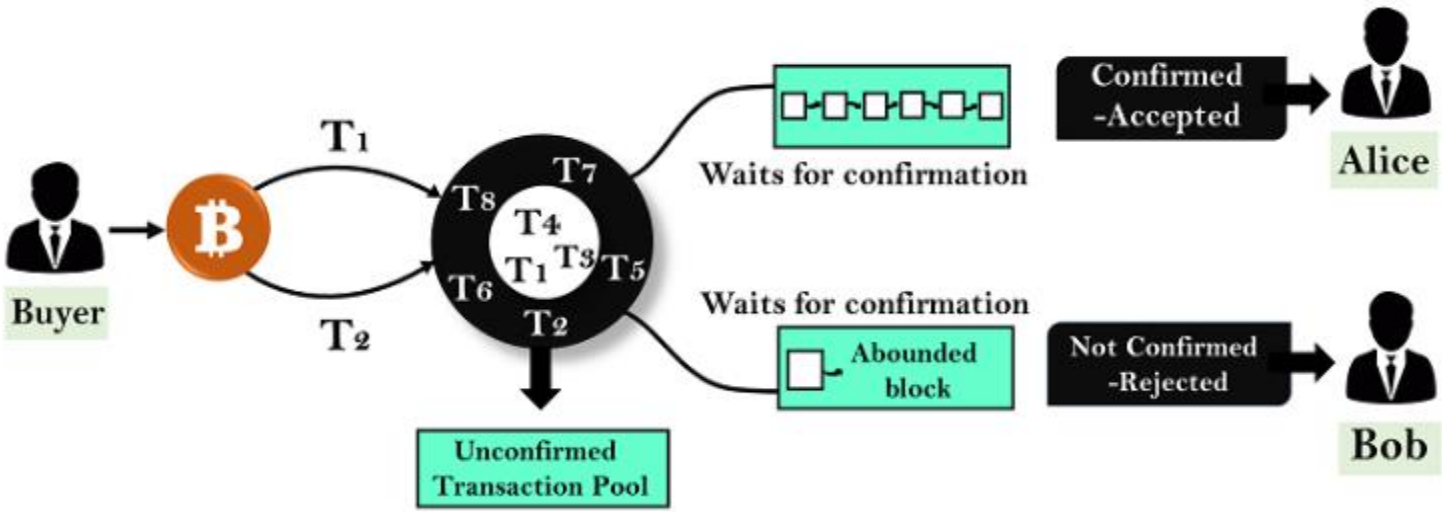
Handling Double Spending:

When the same money is spent more than once, it is called Double Spending.
For example, Alice has 50 bitcoins and she sent 50 bitcoins to Bob and 50 bitcoins to Eve simultaneously.
Then, only one of the transaction will be successful because double spending is not allowed in Bitcoin.

Let us suppose you have 1 BTC and try to spend it twice. You made the 1 BTC transaction to Alice. Again, you sign and send the same 1 BTC transaction to Bob



Bitcoin prevents double spending by following ways:

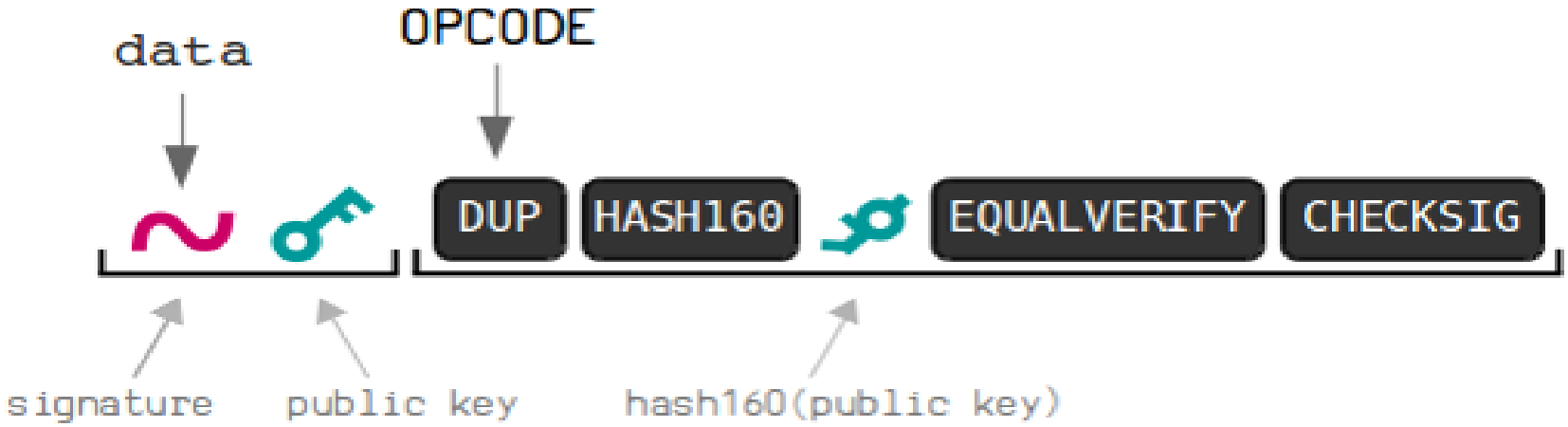
Details about the transaction are sent and forwarded to all or as many other computers as possible.

A constantly growing chain of blocks that contains a record of all transactions is collectively maintained by all computers (each has a full copy).

To be accepted in the chain, transaction blocks must be valid and must include proof of work (one block generated by the network every 10 minutes).

Bitcoin script

Bitcoin script is simple stack based programming language that enables the processing of transaction on the bitcoin blockchain.

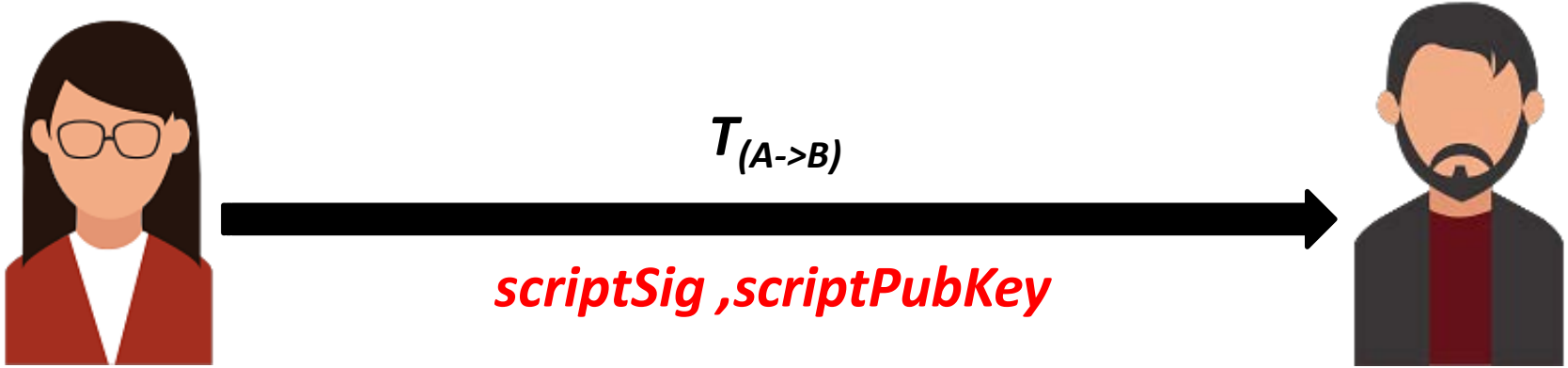


Bitcoin Scripts

- Simple, compact, stack-based and processed left to right
  - FORTH like language

- **Not Turing Complete** (no loops)
  - Halting problem is not there

Bitcoin Scripts – A Simple Example



Bob can spend the bitcoins only if both the scripts return true after execution

Bitcoin Scripts

Transaction Input

scriptSig:

18E14A7B6A30…
D61967F63C7DD…

Transaction Output

scriptPubKey:

OP_DUP
OP_HASH160
16UwLL9Risc3QfPqBUvKof…
OP_EQUALVERIFY
OP_CHECKSIG

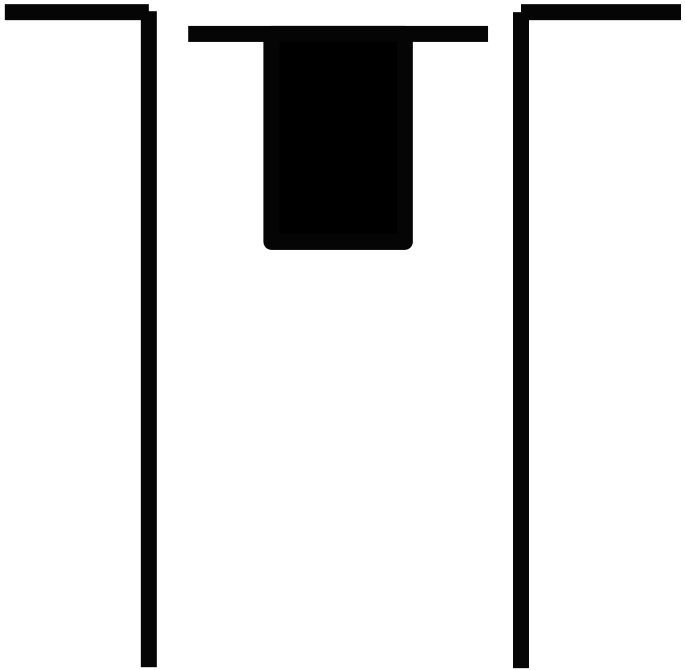
Bitcoin Scripts

scriptPubKey: OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

scriptSig: <sig> <pubKey>

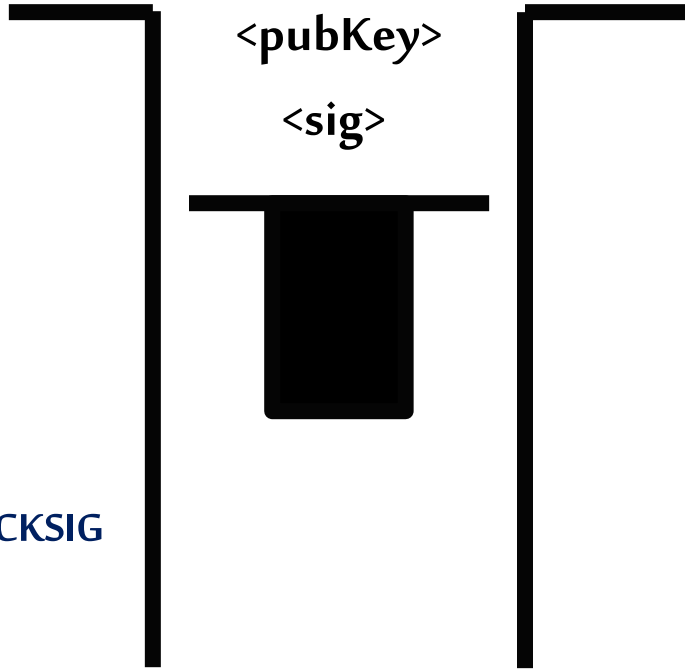
- The stack is initially empty. Both the scripts are combined – input followed by output

<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG



Bitcoin Scripts

<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG



- The stack is initially empty. Both the scripts are combined

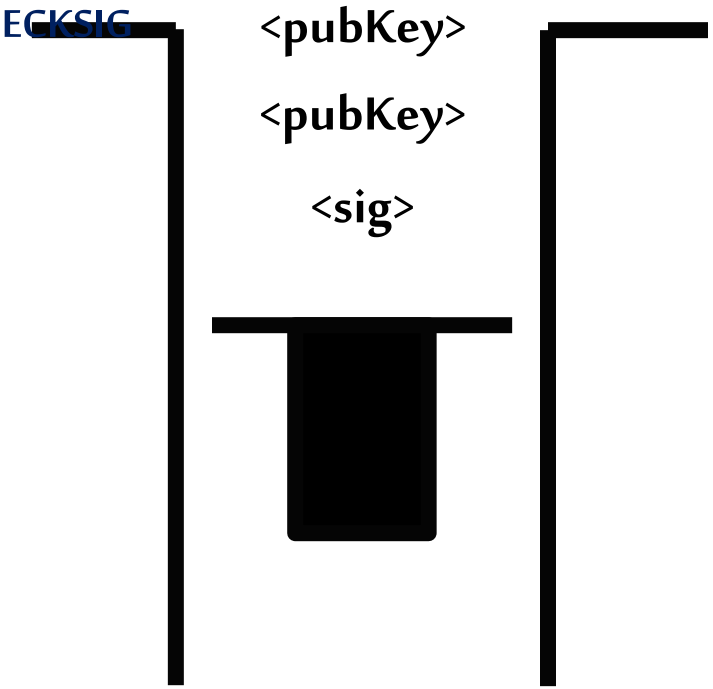
OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

Bitcoin Scripts

OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

- Top stack item is duplicated

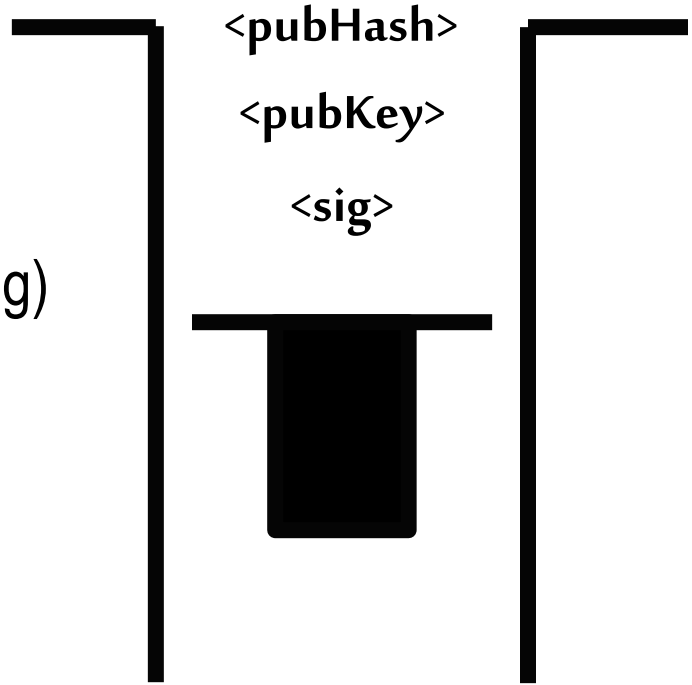
OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG



Bitcoin Scripts

OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

- Top stack item is hashed (RIPEMD-160 hashing)



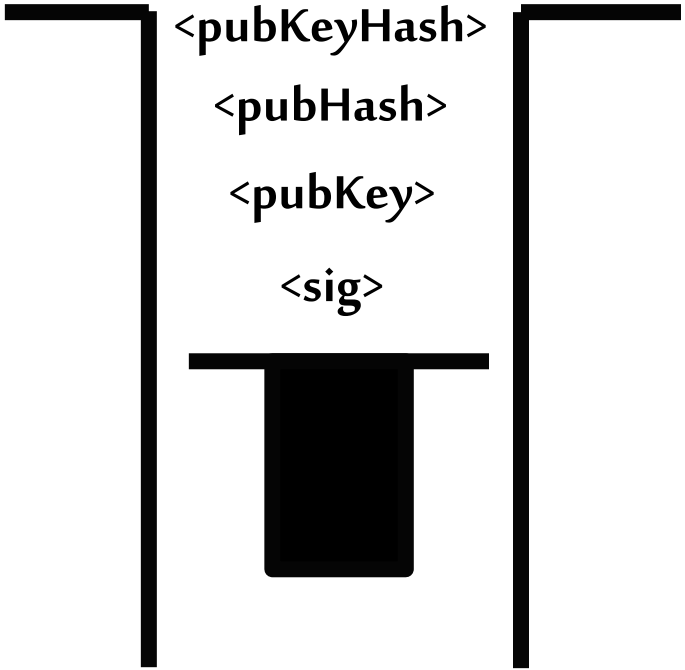
<pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

Bitcoin Scripts

<pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

- The constant is pushed in the stack

OP_EQUALVERIFY OP_CHECKSIG

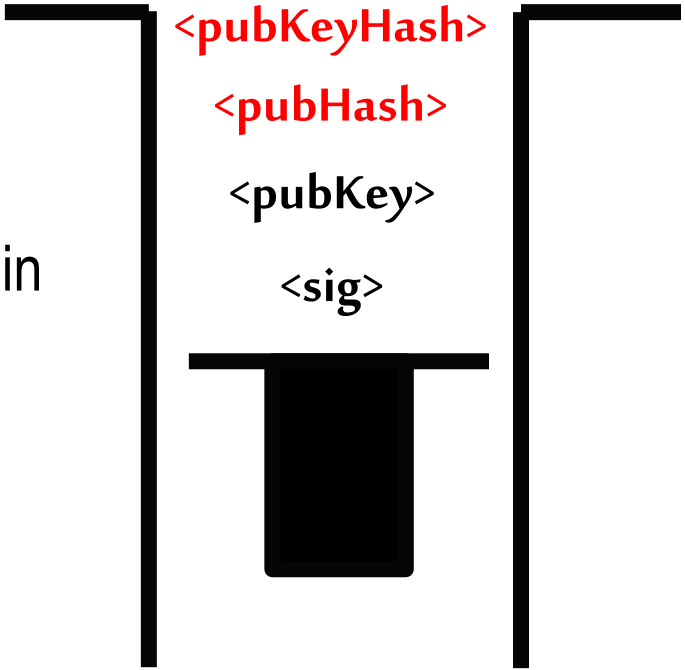


Bitcoin Scripts

OP_EQUALVERIFY OP_CHECKSIG

- Equality is checked between the top two items in the stack

OP_CHECKSIG

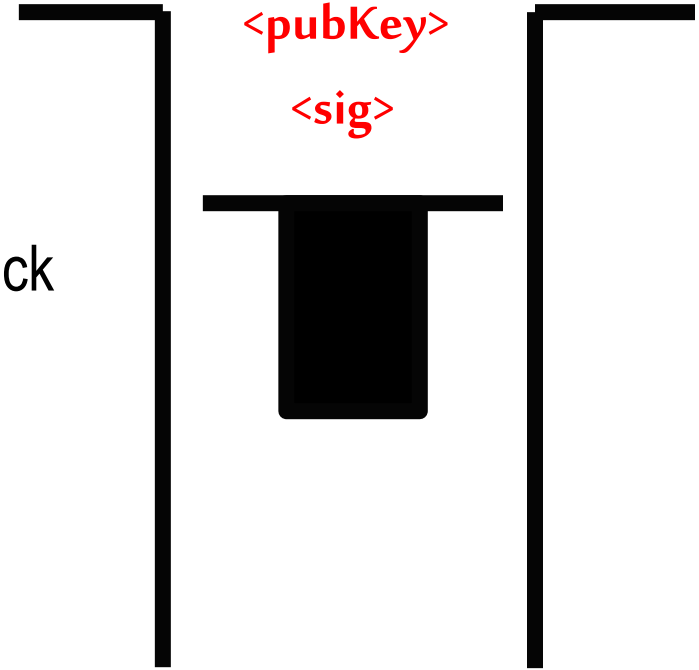


Bitcoin Scripts

OP_CHECKSIG

- Signature is checked based on the top two stack item

TRUE



Interesting Bitcoin Scripts

- **Freezing funds until a time in the future**

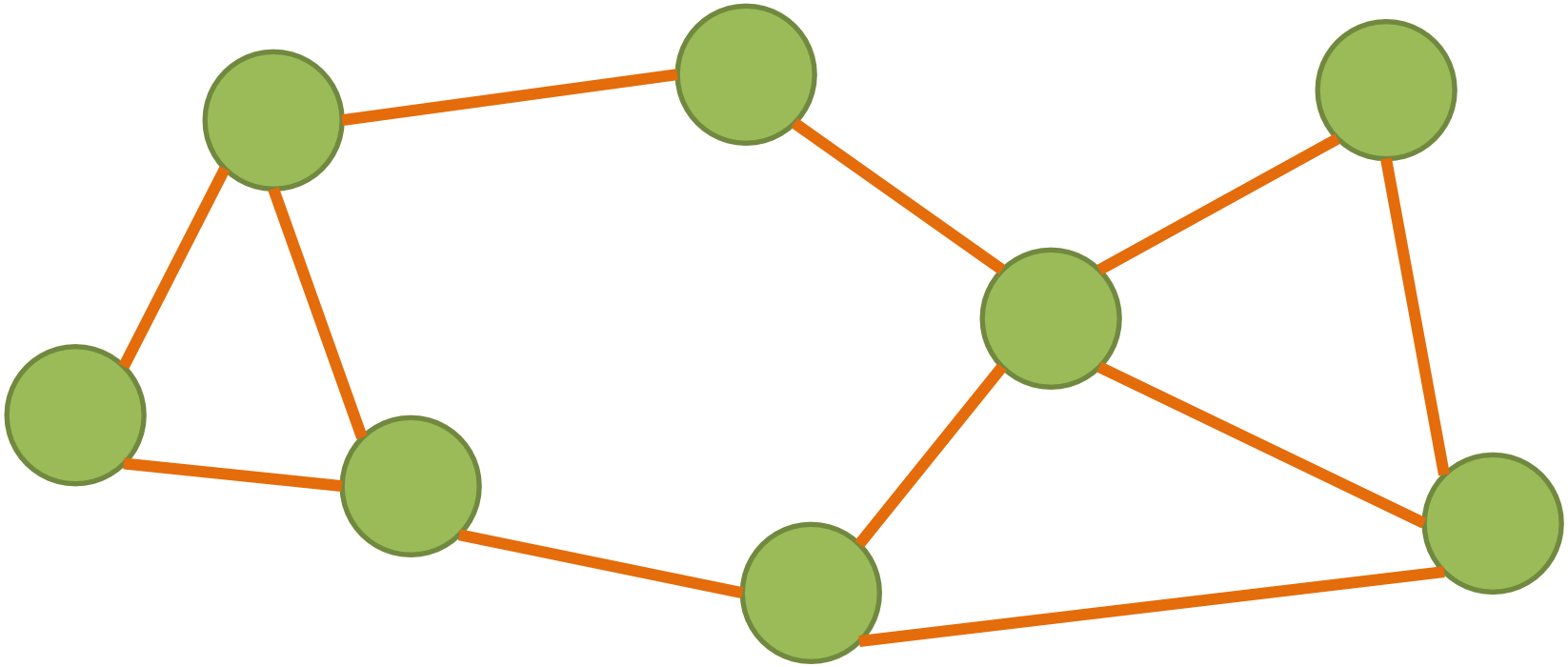
scriptPubKey: <expiry_time> OP_CHECKLOCKTIMEVERIFY OP_DROP OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

scriptSig: <sig> <pubKey>

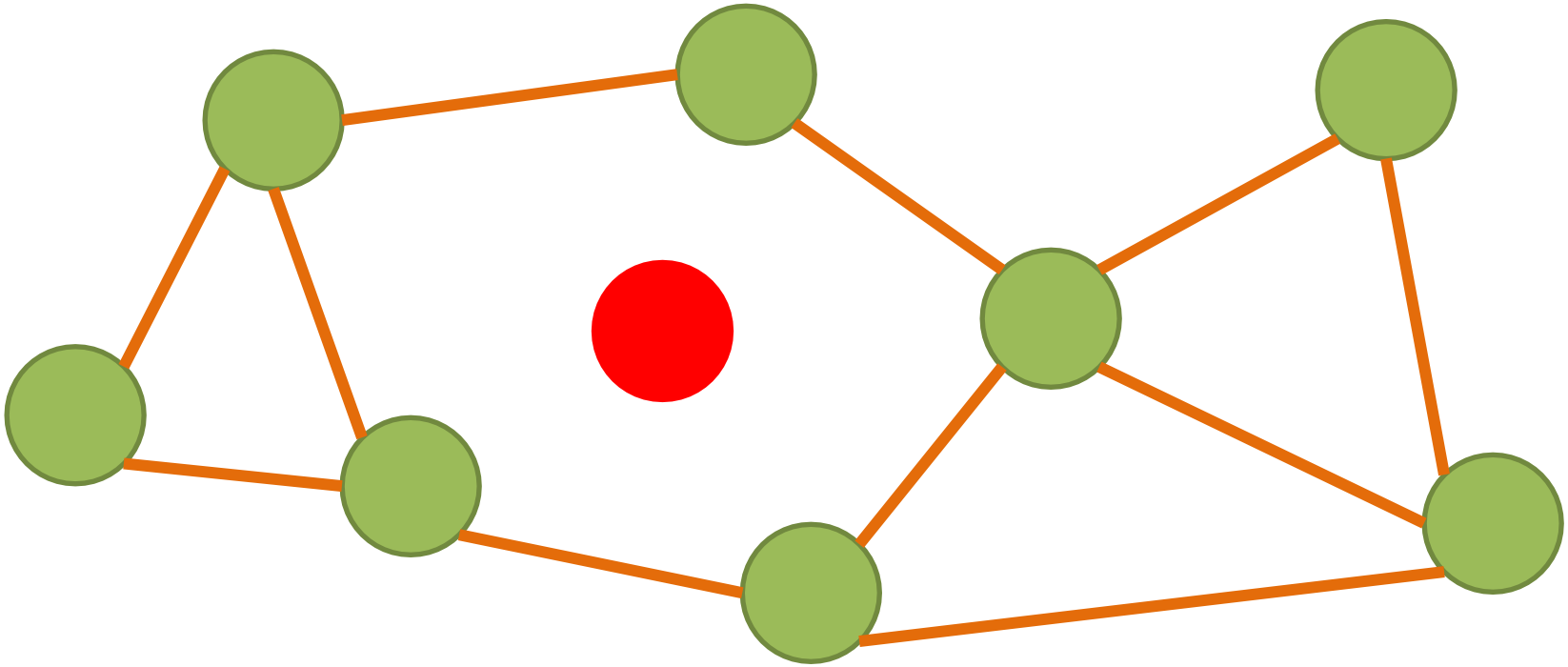
Bitcoin P2P Network

- An ad-hoc network with random topology, Bitcoin protocol runs on TCP port 8333
- All nodes (users) in the bitcoin network are treated equally
- New nodes can join any time, non-responding nodes are removed after 3 hours

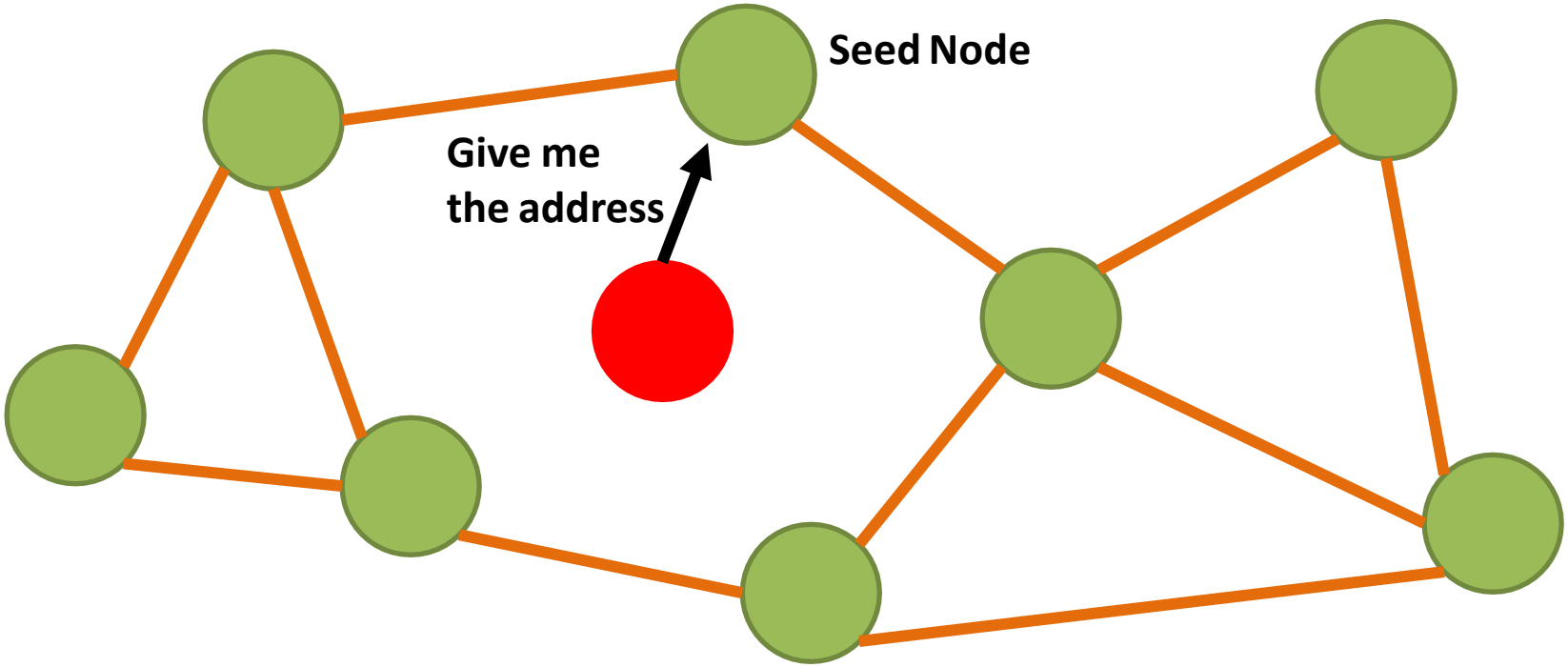
Joining in a Bitcoin P2P Network



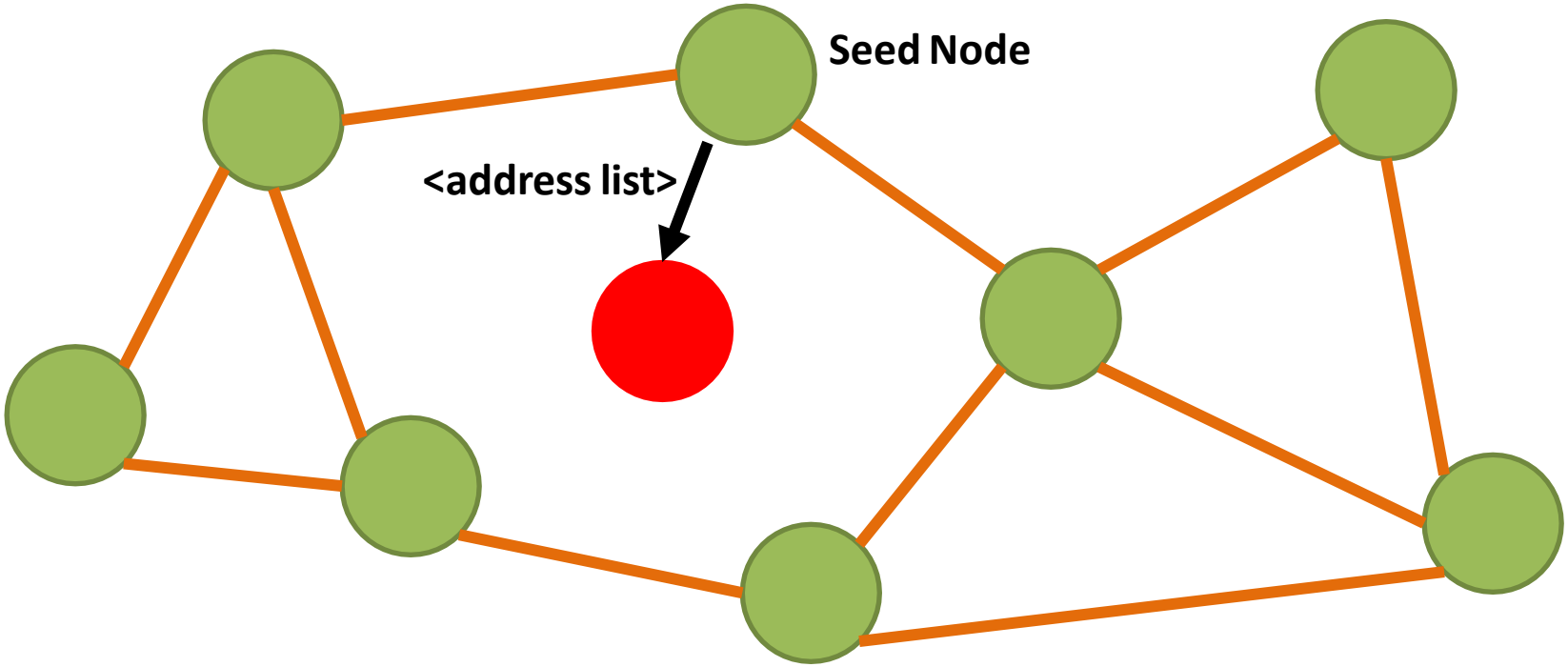
Joining in a Bitcoin P2P Network



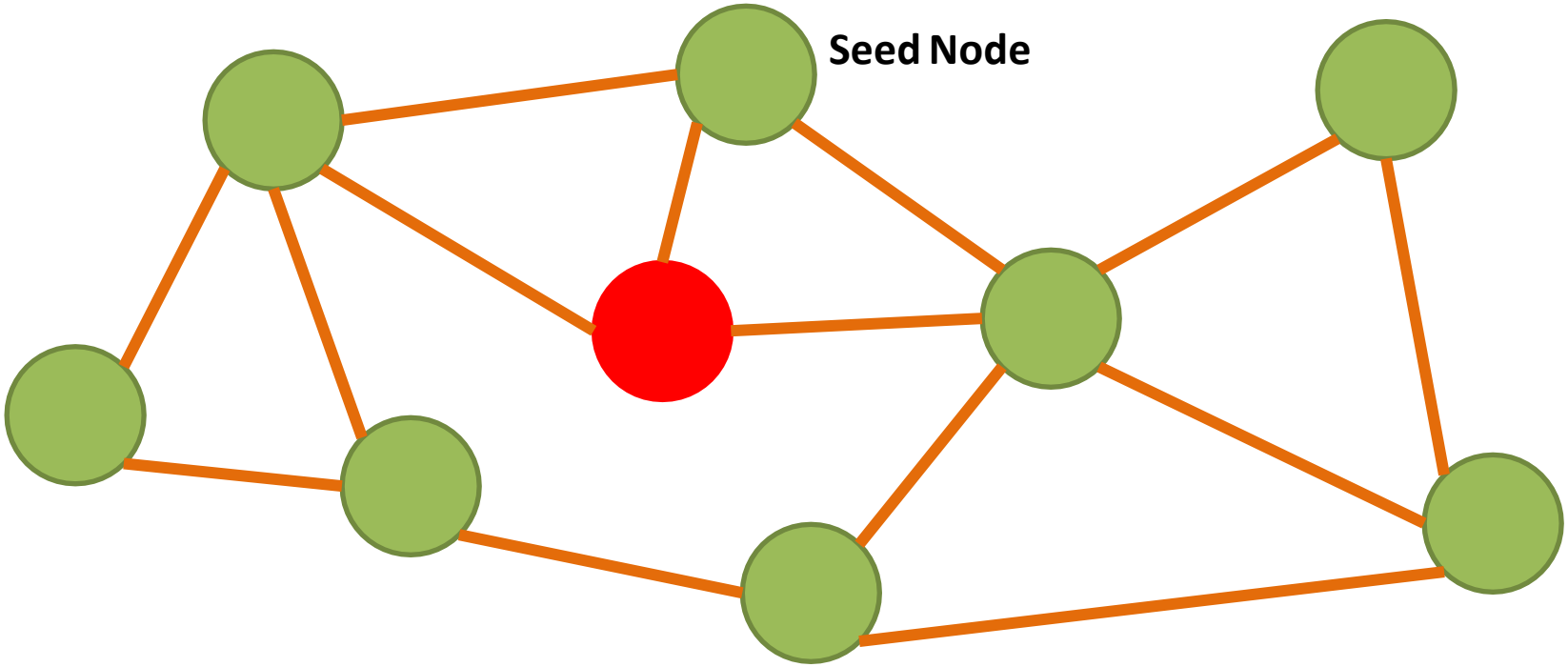
Joining in a Bitcoin P2P Network



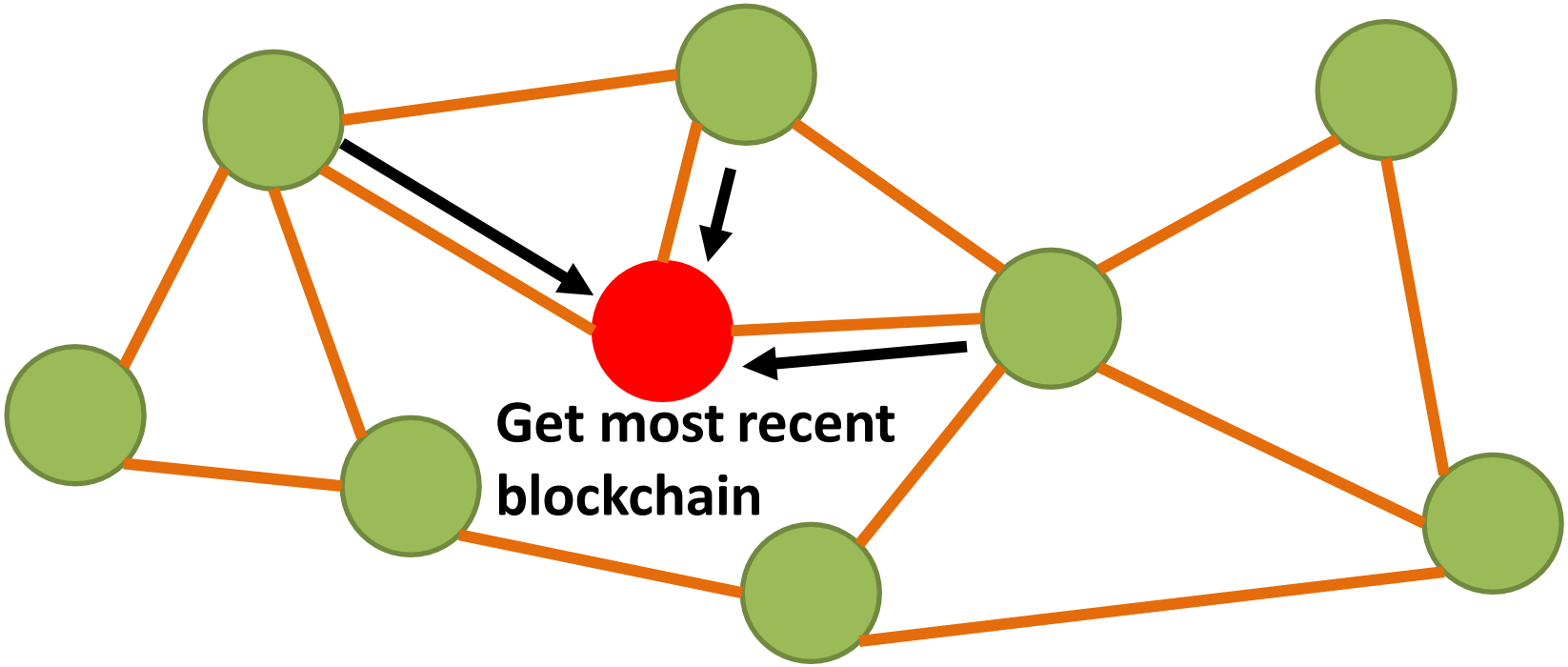
Joining in a Bitcoin P2P Network



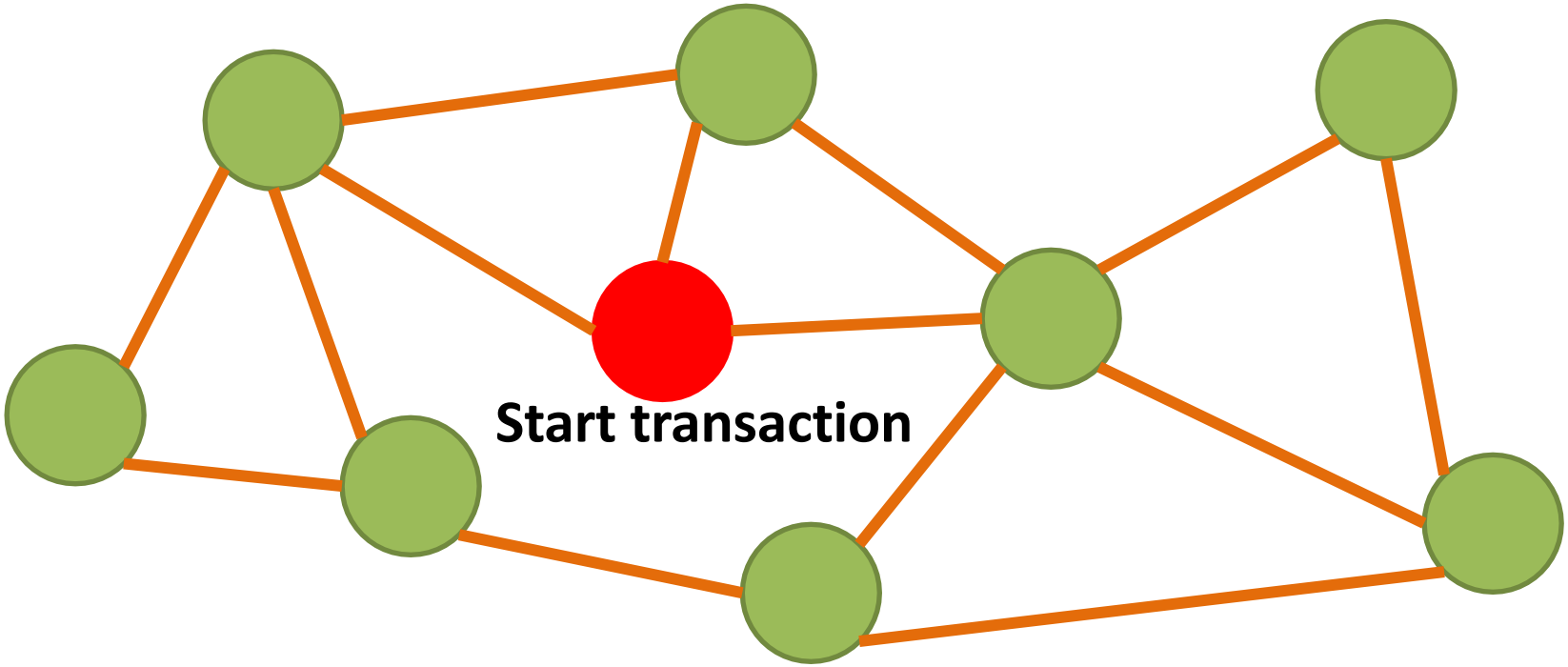
Joining in a Bitcoin P2P Network



Joining in a Bitcoin P2P Network



Joining in a Bitcoin P2P Network



Overview

Initiate the transaction

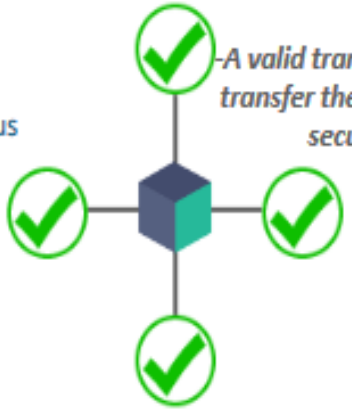
-Multiple parties transact

Somebody decides to transfer a security

The transaction is broadcasted to a P2P network of nodes

Validation via consensus

-The network of nodes validates, verifies using known consensus algorithm



-A valid transaction will transfer the title of the security

Consensus Mechanism Applied

The new block is then added to the existing Blockchain

Once verified, the transaction becomes a part of new block for the ledger

The transaction is complete