

CS61065: Theory and Applications of Blockchain

BLOCKCHAIN ELEMENTS

Department of Computer Science
and Engineering



INDIAN INSTITUTE OF TECHNOLOGY
KHARAGPUR

Sandip Chakraborty
sandipc@cse.iitkgp.ac.in

Shamik Sural
shamik@cse.iitkgp.ac.in

What is Blockchain?

- A Platform for executing transactional services
- Spanned over multiple organizations or individuals who may not (**need not**) **trust** each other
- An append-only shared ledger of digitally signed and encrypted transactions replicated across a network of peer nodes

The Block in a Blockchain – Securing Data Cryptographically

- Digitally signed and encrypted transactions “**verified**” by peers
- **Cryptographic security** – Ensures that participants can only view information on the ledger that they are authorized to see

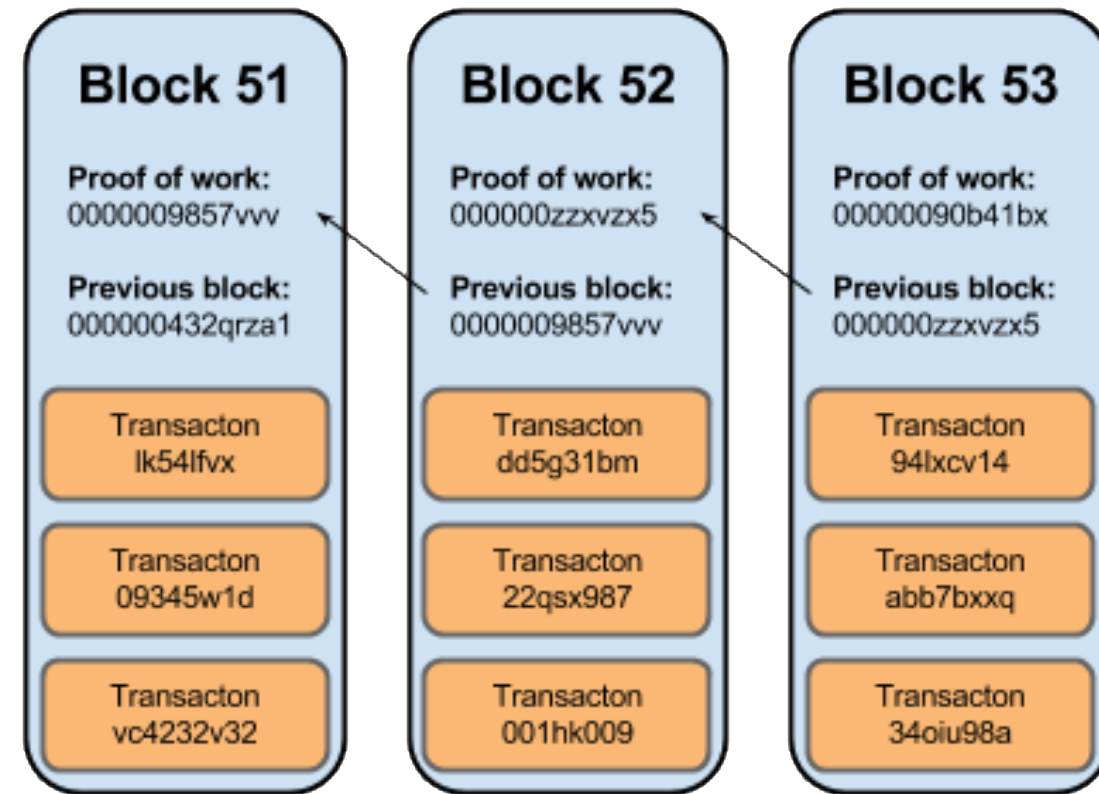





Image source: <http://dataconomy.com/>

Structure of a Block

- A block is a **container data structure** that contains a series of transactions
- **In Bitcoin:** A block may contain more than 500 transactions on average, the average size of a block is around 1 MB (an upper bound proposed by Satoshi Nakamoto in 2010)
 - May grow up to 8 MB or sometime higher (several conflicting views on this!!)
 - Larger blocks can help in processing large number of transactions in one go.
 - But longer time for verification and propagation

Structure of a Block (Reference: Bitcoin)

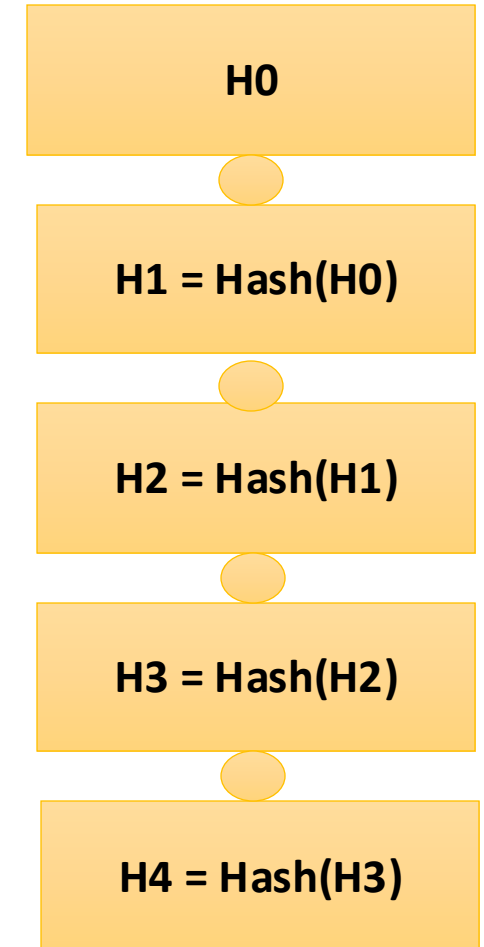
- Two components:
 - **Block Header**
 - **List of Transactions**

| | | | |
|--|---------------------|--------------|---|
| Explorer / Bitcoin Explorer / Blocks / Block | | | |
| Block Hash | | | |
| 000000000000000000050c4cdfc194497debe275b08f6fef6a45bd1310702c3f  | | | |
| Summary | | | |
| Height | ◀ 697,125 ▶ | Relayed By |  BTC.com |
| Confirmations | 27 | Difficulty | 55.76 T / 15.56 T |
| Block Size | 1,487,145 Bytes | Block Reward | 6.25000000 BTC |
| Stripped Size | 834,379 Bytes | Fee Reward | 0.20304797 BTC |
| Weight | 3,990,282 | Tx Count | 2,126 |
| Time | 2021-08-23 10:53:47 | Tx Volume | 21,673.40662679 BTC |
| Merkle Root | | | |
| 84ed654724b7ae29e885569ddac4dc2fdef11e247e833f0289de27ca6243d635 | | | |
| Version | | | |
| 0x3fff0004 | | | |
| Nonce | | | |
| 0x7028f9f2 | | | |
| Bits | | | |
| 0x1712180b | | | |
| Other Explorers | | | |
|  BLOCKCHAIR | | | |

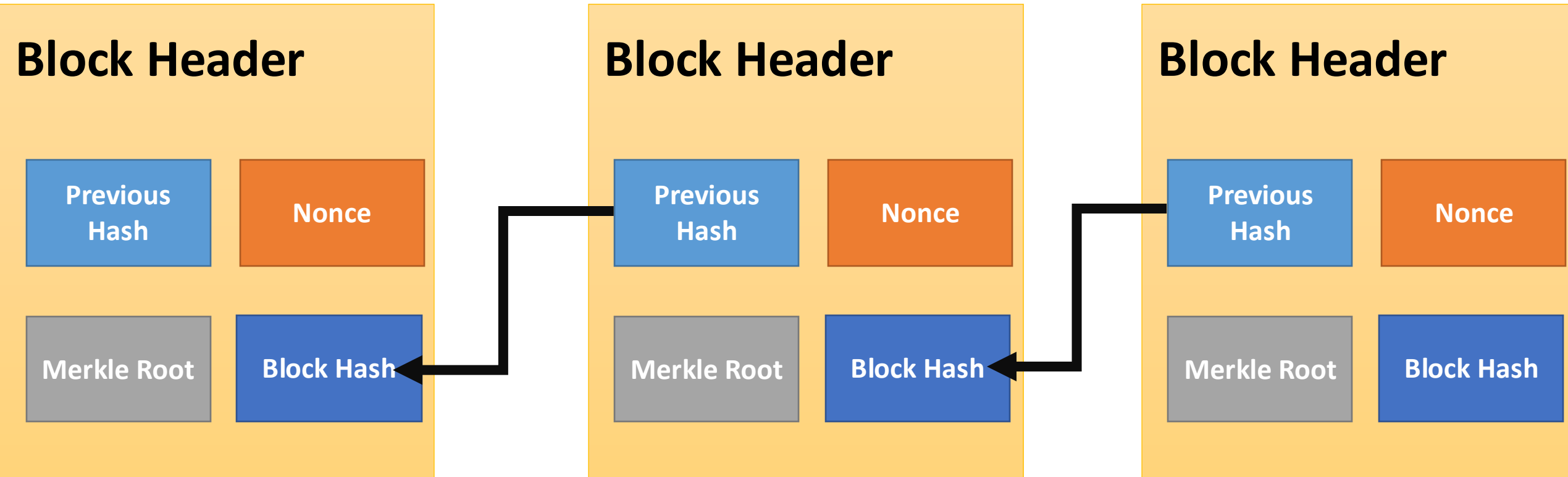
Block Source: <https://btc.com/btc/blocks> OR <https://blockchain.com/explorer>

Block Header (Reference: Bitcoin)

- Metadata about a block – (1) Previous block hash, (2) Mining statistics used to construct the block, (3) Merkle tree root
- **Previous block hash:** Every block inherits from the previous block – we use previous block's hash to create the new block's hash – make the blockchain **tamper proof**.



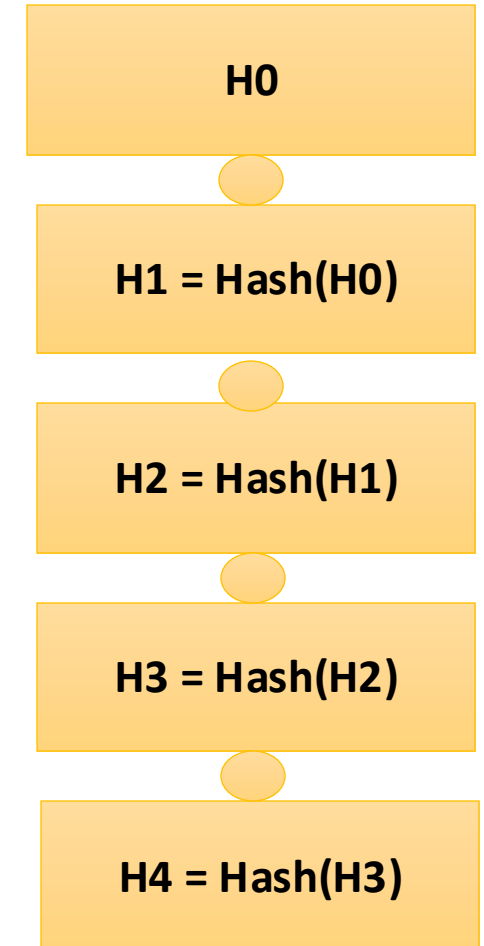
Block Generation Puzzle



Find out the nonce which generates the desired hash (certain zero bits at the prefix - **0000000000000000000000004a2b84f93a285b7a7.....**)

Block Header (Reference: Bitcoin)

- **Mining** – the mechanism to generate the hash
 - The mechanism needs to be complicated enough, to make the blockchain **tamper proof**
 - **Bitcoin Mining:** $H_k = \text{Hash}(H_{k-1} || T || \text{Nonce} || \text{Something more})$
 - Find the nonce such that H_k has certain predefined **complexity** (value less than a target value)
- The header contains mining statistics – timestamp, nonce and difficulty
- Understanding Difficulty and Bits
<https://medium.com/@dongha.sohn/bitcoin-6-target-and-difficulty-ee3bc9cc5962>
- Difficulty is the largest target (0x0000 0000 00FF FF00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000) divided by the current target, e.g., (0x0000 0000 0000 0000 0012 180B 0000 0000 0000 0000 0000 0000 0000 0000 0000)
- Remember: “Cost of Mining” – Pretty High (Computing Power and Energy)



The Hashes in a Block Header (Reference: Bitcoin)

- Block identifier – the hash of the current block header (Hash algorithm: Double SHA256)
- Merkle Root
- Previous block hash is used to compute the current block hash
- Timestamp, Previous hash, Merkle root, Difficulty Bits, Nonce and Version used to compute current hash
- DEMO

<https://cse.buffalo.edu/blockchain/blockhash.html>

Block Source: <https://btc.com/btc/blocks>

Block Generation Cost

- Energy efficiency $\sim 0.098 \text{ J/GH} = \sim 100 \text{ J/TH}$
- ASIC Hardware for bitcoin can perform about 750 TH/s
- Hash rate of the Bitcoin network approx. 120M TH/s!! Many actually go waste 😞 <https://www.blockchain.com/charts/hash-rate>
- Bitcoin network consumes about 80 TW-hours of electricity annually. These figures vary between sources and are all some form of estimates
- Average household in Germany of four people consumes approx. 4,000 KW-hours of electricity per year.
- Can power about 20,000 households
- Concept of Pooling is used
- What ensures tamperproof operation in terms of honest nodes??

The Blockchain Replicas




- Every peer in a Blockchain network maintains a local copy of the Blockchain.
- Size is just about 591 GB 😊
- As a new user joins the network, she can get the whole copy
- **Requirements**
 - All the replicas need to be **updated** with the last mined block
 - All the replicas need to be **consistent** – the copies of the Blockchain at different peers need to be **exactly similar**

Transactions in a Block (Reference: Bitcoin)

- Transactions are organized as a Merkle Tree. The Merkle Root is used to construct the block hash
- If you change a transaction, you need to change all the subsequent block hashes
- The **difficulty** of the mining algorithm determines the **toughness** of tampering with a block in a blockchain

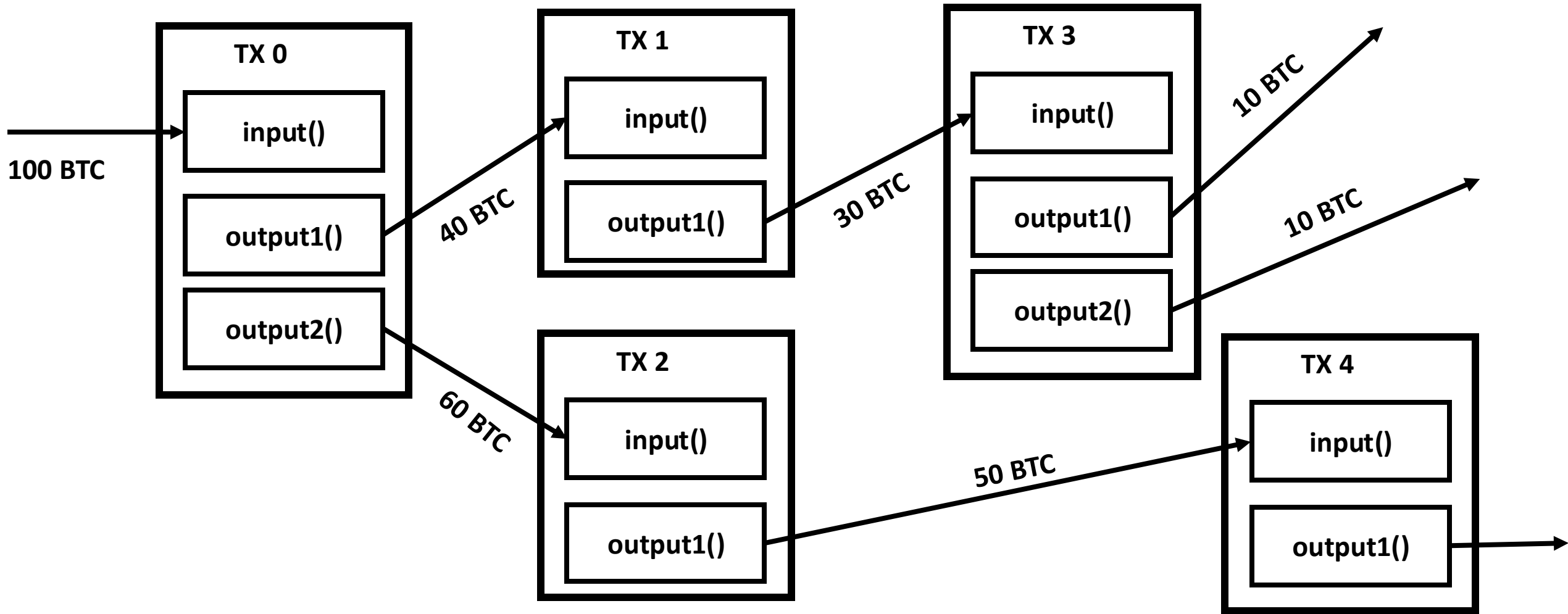
Transactions in a Block (Reference: Bitcoin)

Transactions

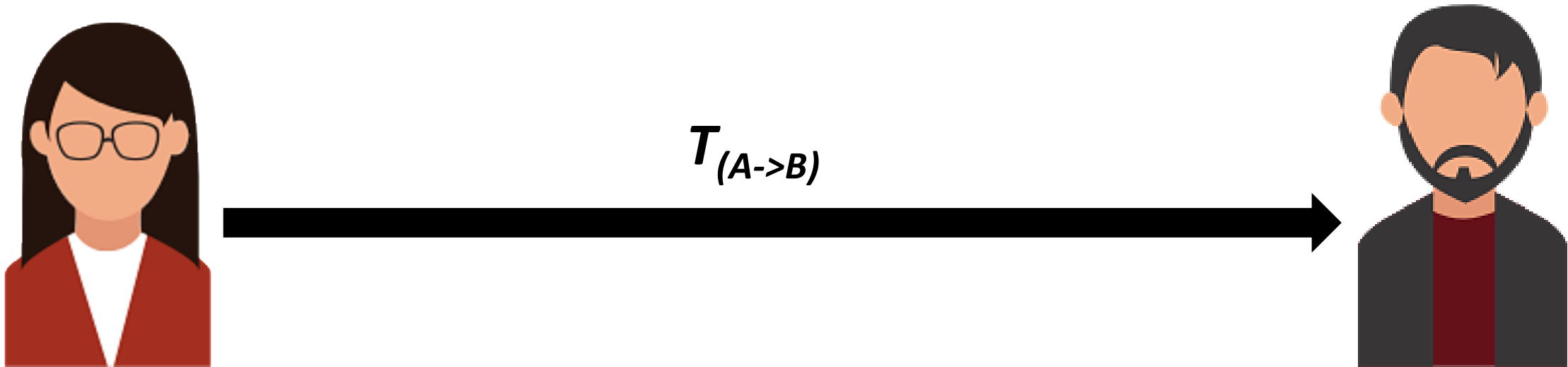
| | | |
|--|---|--------------------------|
| 3f5ebfaf7fe18176cffe973f4d609ba2d366bdb1755ddf464c93b5f7ba3d787 | | 2017-12-20 20:02:40 |
| No Inputs (Newly Generated Coins) |  1Hz96kJKF2HLPGY15JWLB5m9qGNxvt8tHJ Unable to decode output address | 19.69384324 BTC 0 BTC |
| | | 19.69384324 BTC |
| 717e4d969a2241065afe896986bf2b481ab5059d3dba901dc0c0f1feca796524 | | 2017-12-20 20:00:14 |
| 3GsDfabsubnrUSdm9oUedZJSPTnrevVvz |  1H744xJpRVctkTU3jnQtXZg1jVbPfuorLS | 2.96441546 BTC |
| | | 2.96441546 BTC |
| 8ce2ddf6236b3252c49fb3ad28c4a2584047de91643bc9724d272c91295423ee | | 2017-12-20 19:59:57 |
| 16oQyApVNxWkwyXZok9eHSKxYX57SHLgvV |  1Dv56y3i1DzcD3nENAvkq4QR3eKdoGytbd | 0.02983573 BTC |
| | | 0.02983573 BTC |

Block Source: <https://blockchain.info/>

Bitcoin Transactions and Input and Output

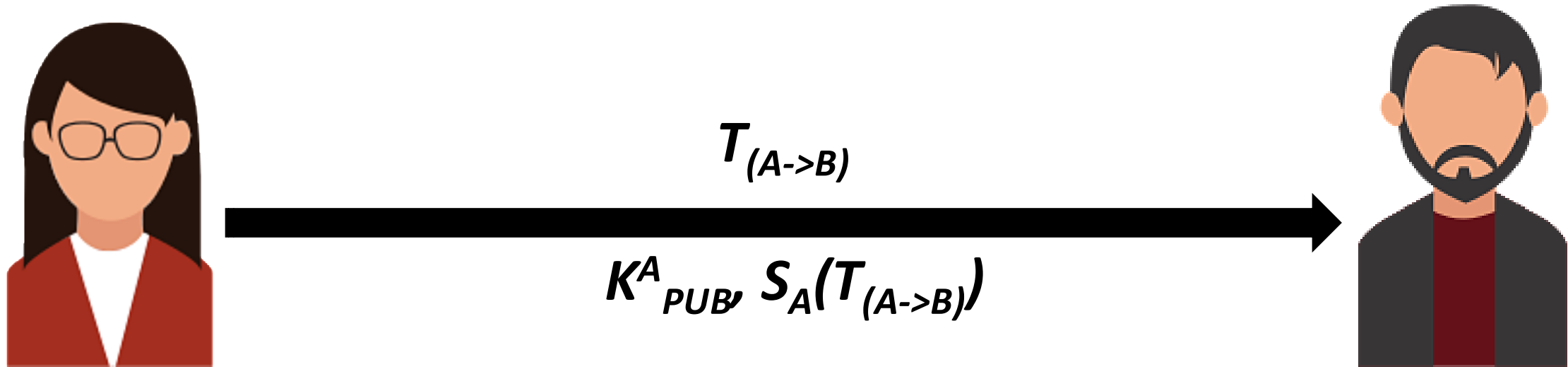


Bitcoin Scripts – A Simple Example



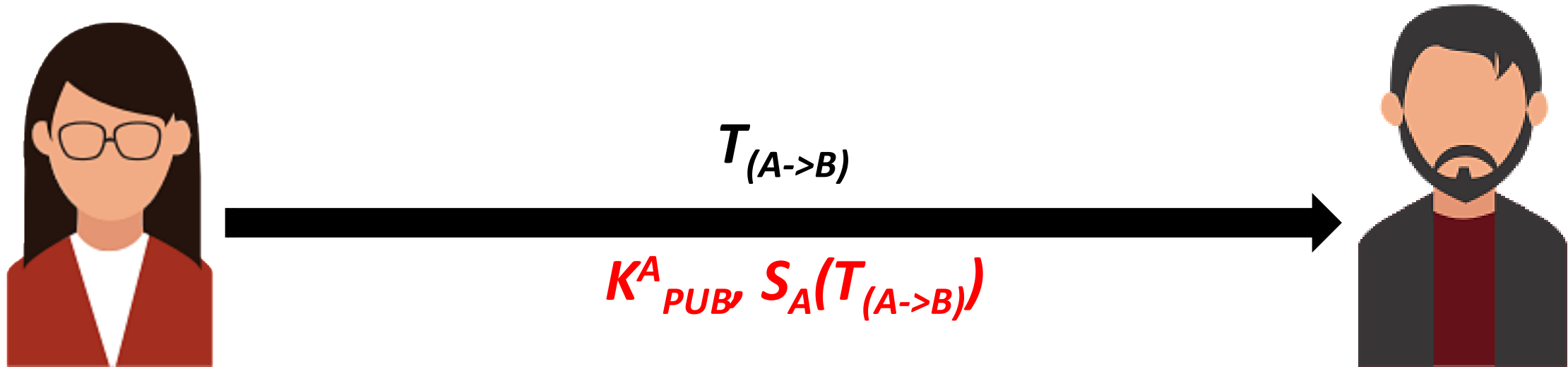
How Bob will verify that the transaction is actually originated from Alice?

Bitcoin Scripts – A Simple Example



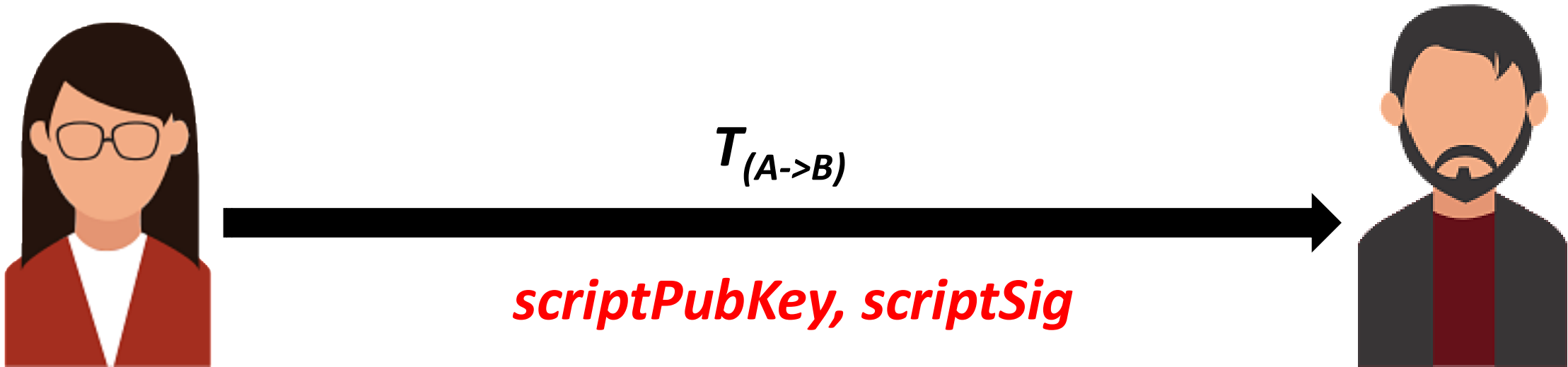
Send the public key of Alice along with the signature -> Bob can verify this

Bitcoin Scripts – A Simple Example



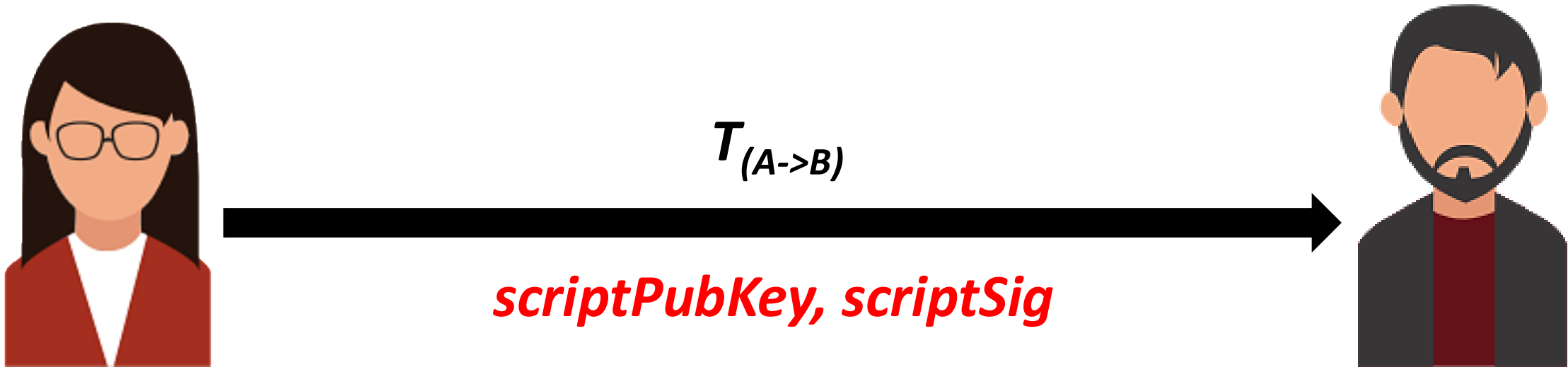
Bitcoin indeed transfers scripts instead of the signature and the public key

Bitcoin Scripts – A Simple Example



Bitcoin indeed transfers scripts instead of the signature and the public key

Bitcoin Scripts – A Simple Example

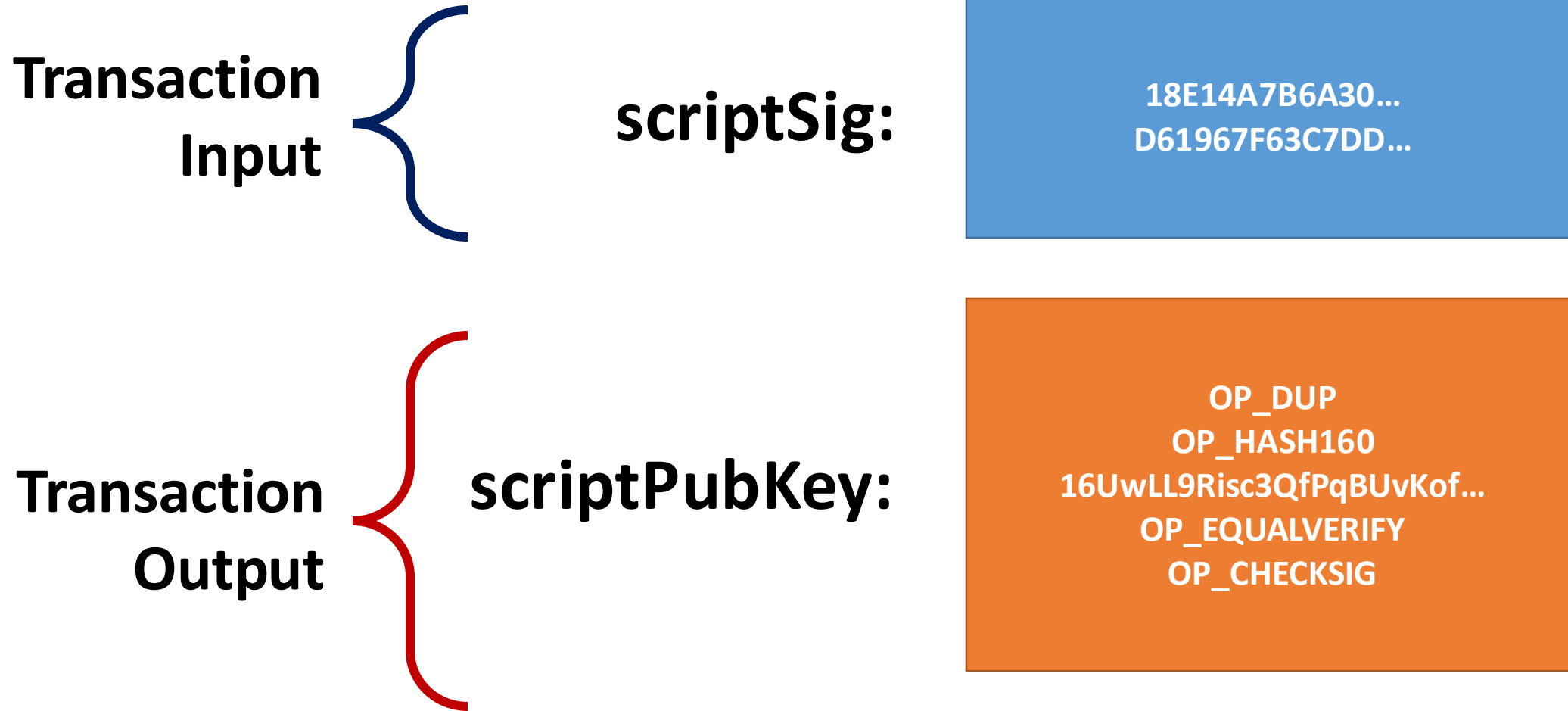


Bob can spend the bitcoins only if both the scripts return `true` after execution

Bitcoin Scripts

- With every transaction Bob must provide
 - A public key that, when hashed, yields the address of Bob embedded in the script
 - A signature to provide ownership of the private key corresponding to the public key of Bob

Bitcoin Scripts



See for detailed steps: <https://developer.bitcoin.org/devguide/transactions.html>

Simple Example: <https://medium.com/@aalim.khan/bitcoin-transactions-scripts-and-digital-signatures-506688e1630a>

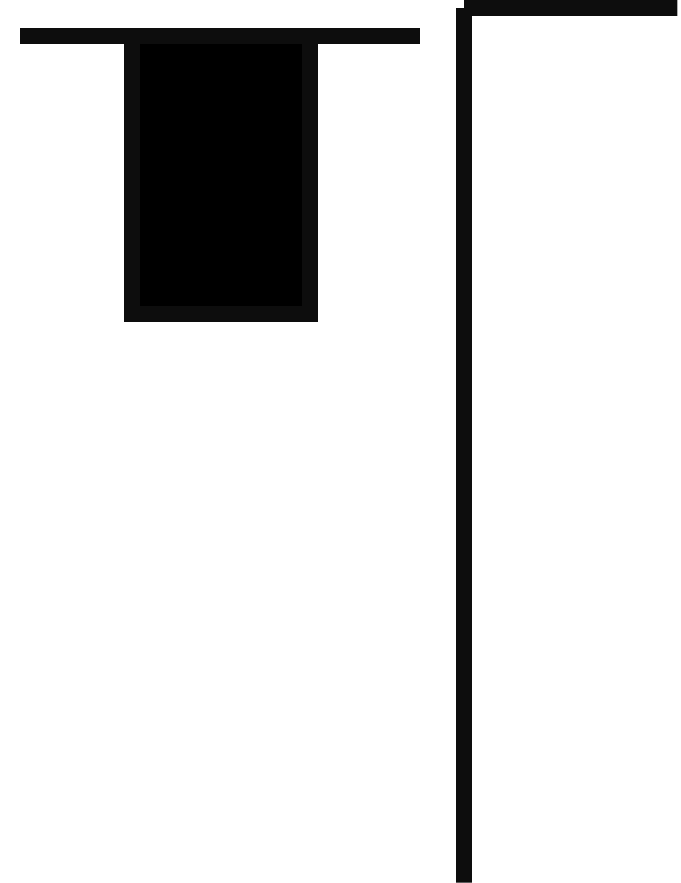
Bitcoin Scripts

```
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
```



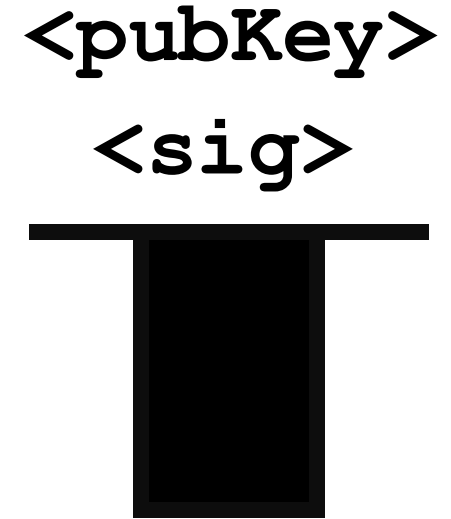
Bitcoin Scripts

`<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`

`<pubKey>
<sig>`

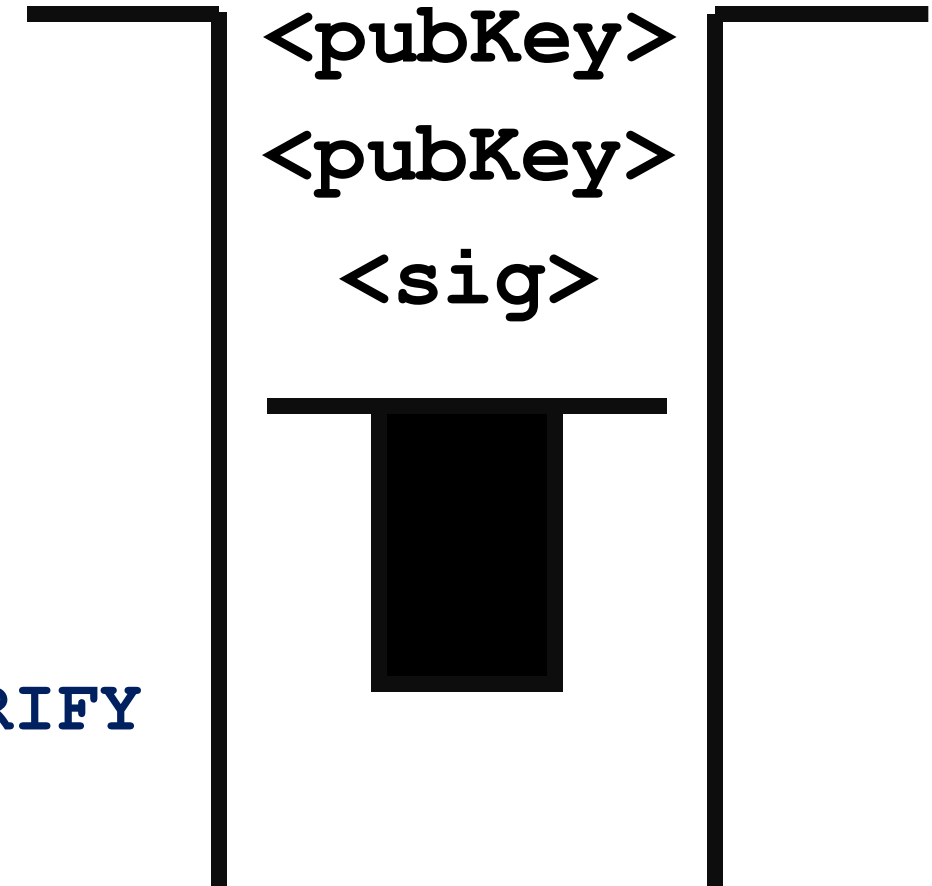


Bitcoin Scripts

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

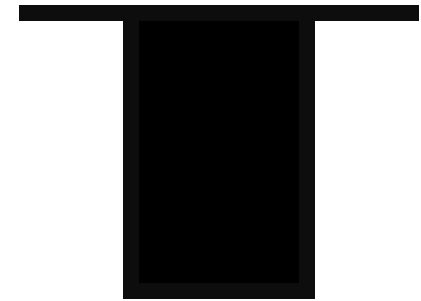
- Top stack item is hashed (RIPEMD-160 hashing)

<pubKeyHash> **OP_EQUALVERIFY** **OP_CHECKSIG**

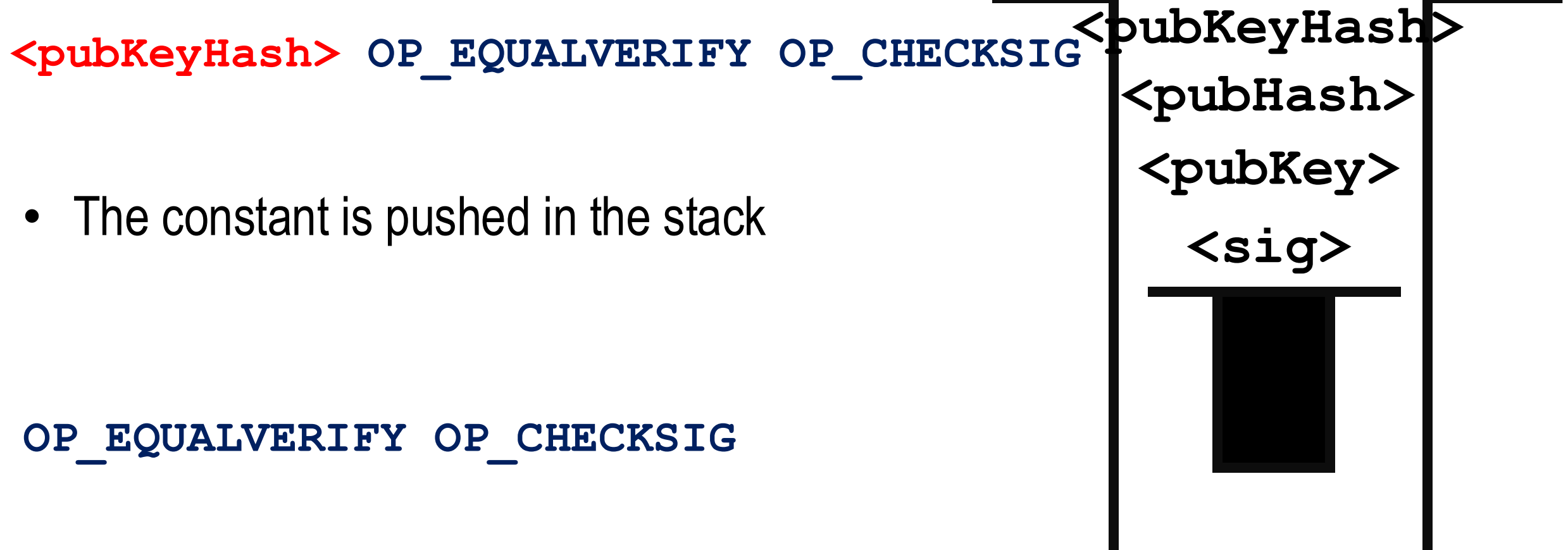
<pubHash>

<pubKey>

<sig>



Bitcoin Scripts

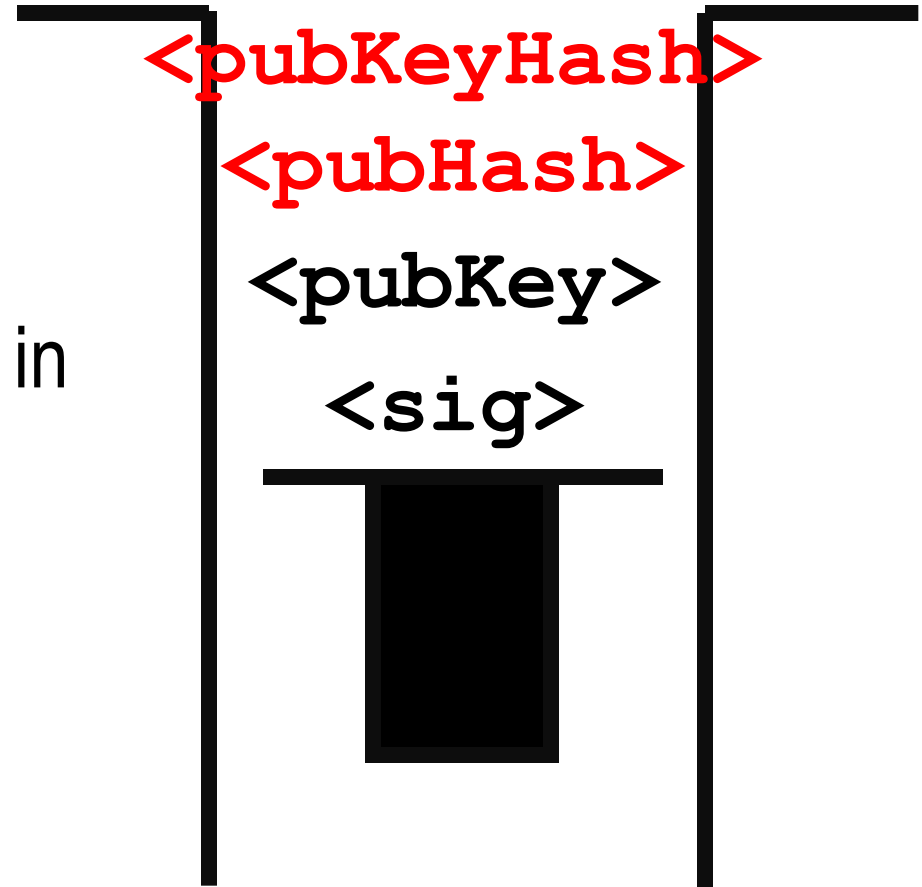


Bitcoin Scripts

OP_EQUALVERIFY **OP_CHECKSIG**

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

OP_CHECKSIG

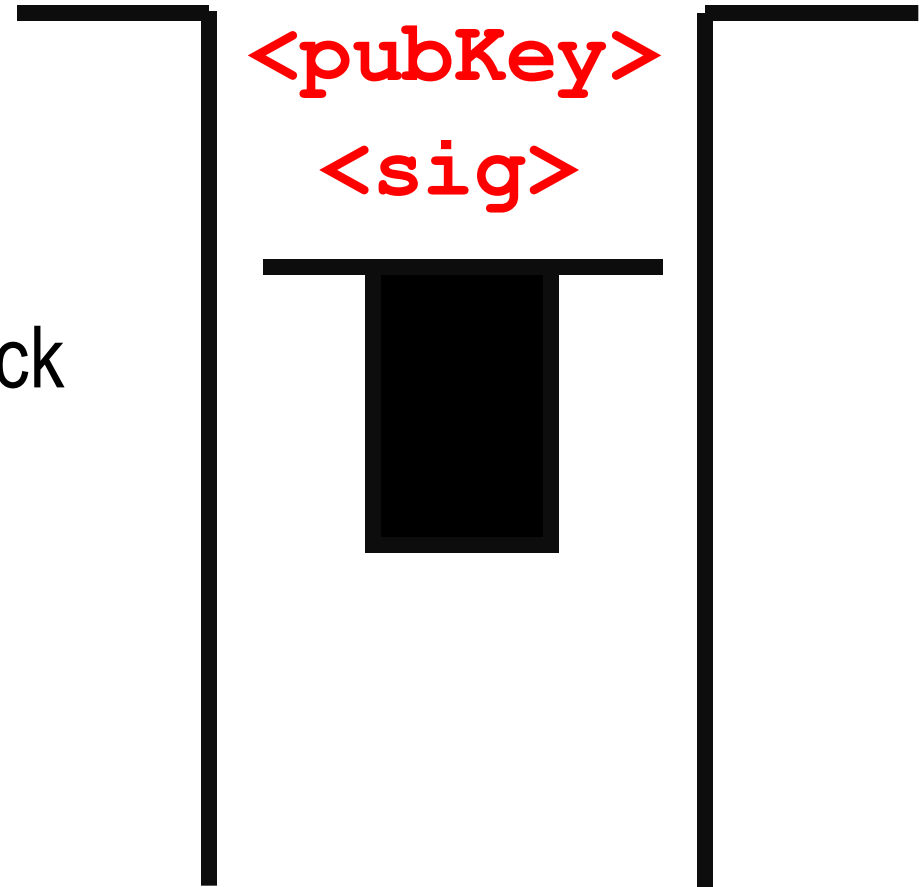


Bitcoin Scripts

OP_CHECKSIG

- Signature is checked based on the top two stack item

TRUE



Bitcoin Script Instructions

- Total 256 opcodes (15 disabled, 75 reserved)
 - Arithmetic operations
 - if-then conditions
 - Logical operators
 - Data handling (like OP_DUP)
 - Cryptographic operations
 - Hash functions
 - Signature verification
 - Multi-signature verification

Interesting Bitcoin Scripts

- Provably un-spendable or prunable outputs

```
scriptPubKey: OP_RETURN {zero or more ops}
```

- Anyone-can-spend outputs

```
scriptPubKey: {empty}
```

```
scriptSig: OP_TRUE
```

Source: <https://en.bitcoin.it/wiki/Script>

Interesting Bitcoin Scripts

- Freezing funds until a time in the future

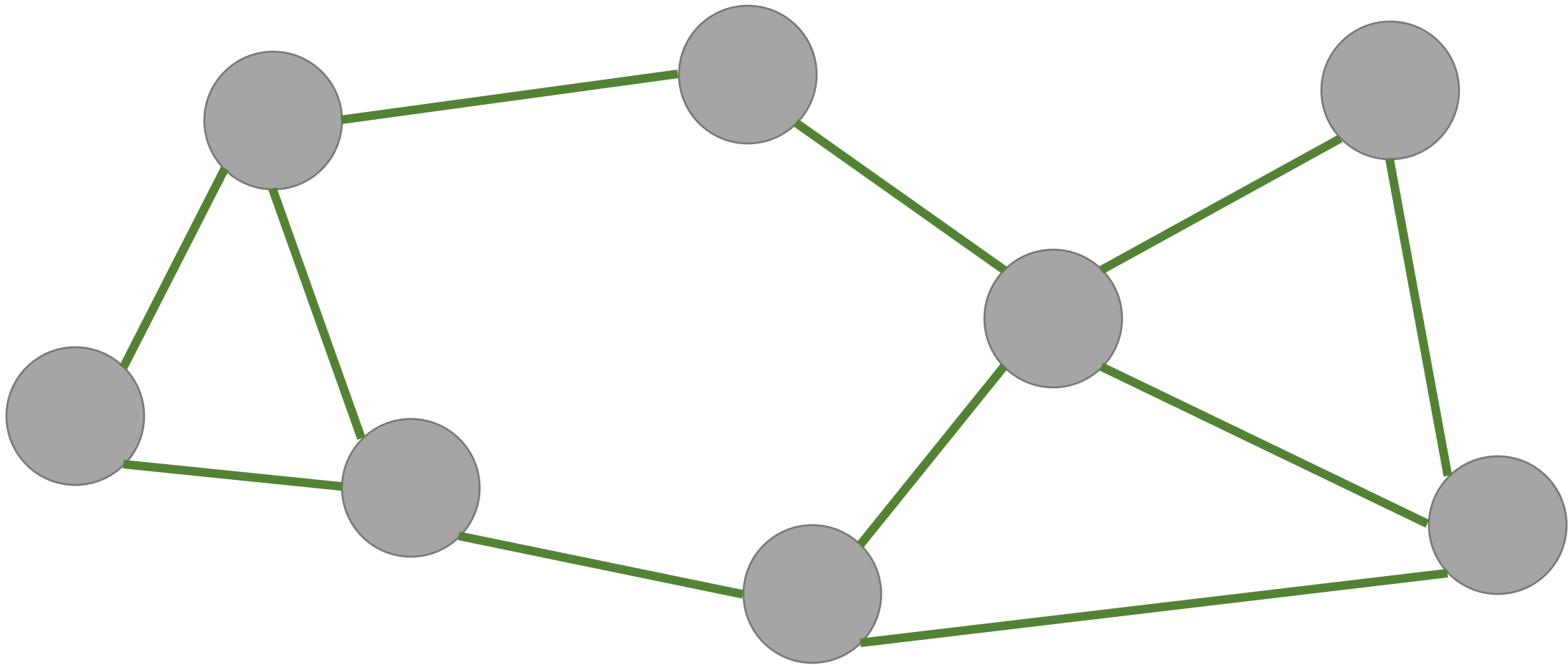
```
scriptPubKey: <expiry_time> OP_CHECKLOCKTIMEVERIFY  
OP_DROP OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY  
OP_CHECKSIG  
scriptSig: <sig> <pubKey>
```

Source: <https://en.bitcoin.it/wiki/Script>

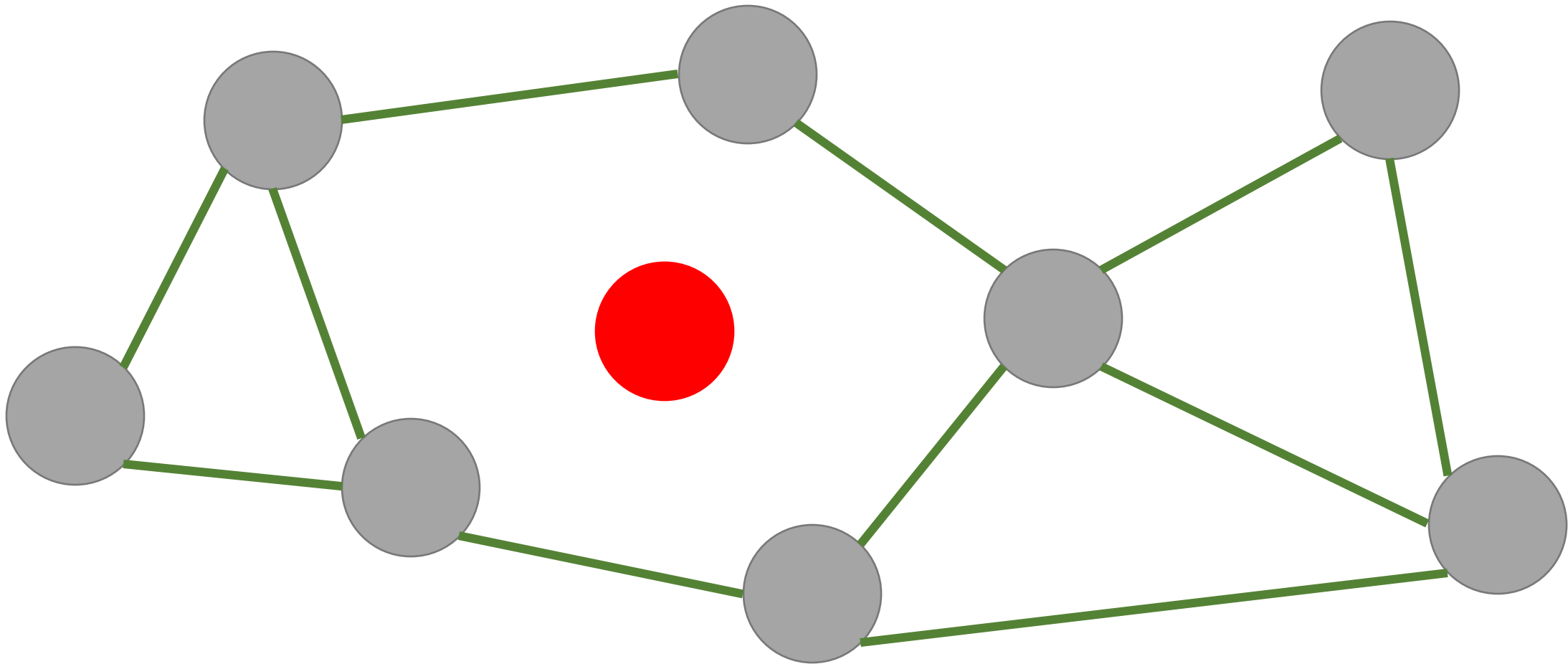
Bitcoin P2P Network

- 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

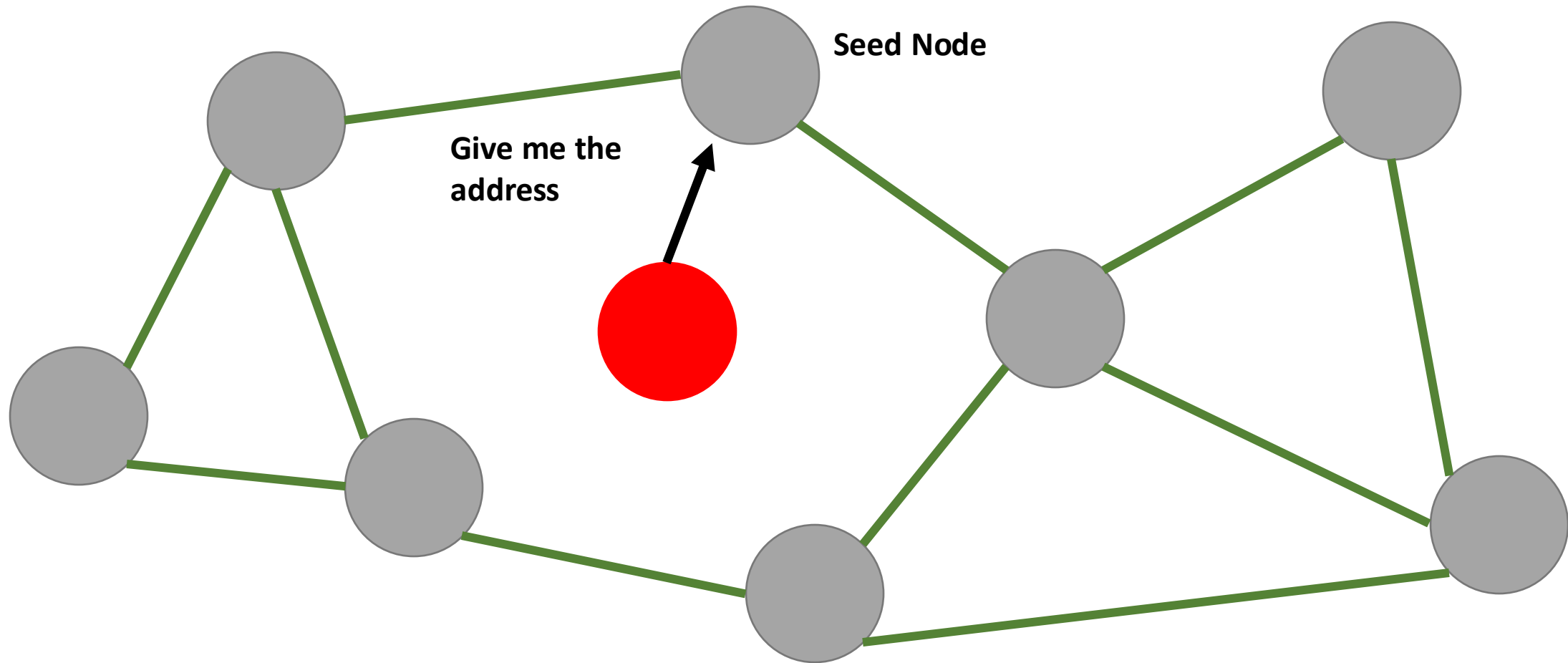
Joining in a Bitcoin P2P Network



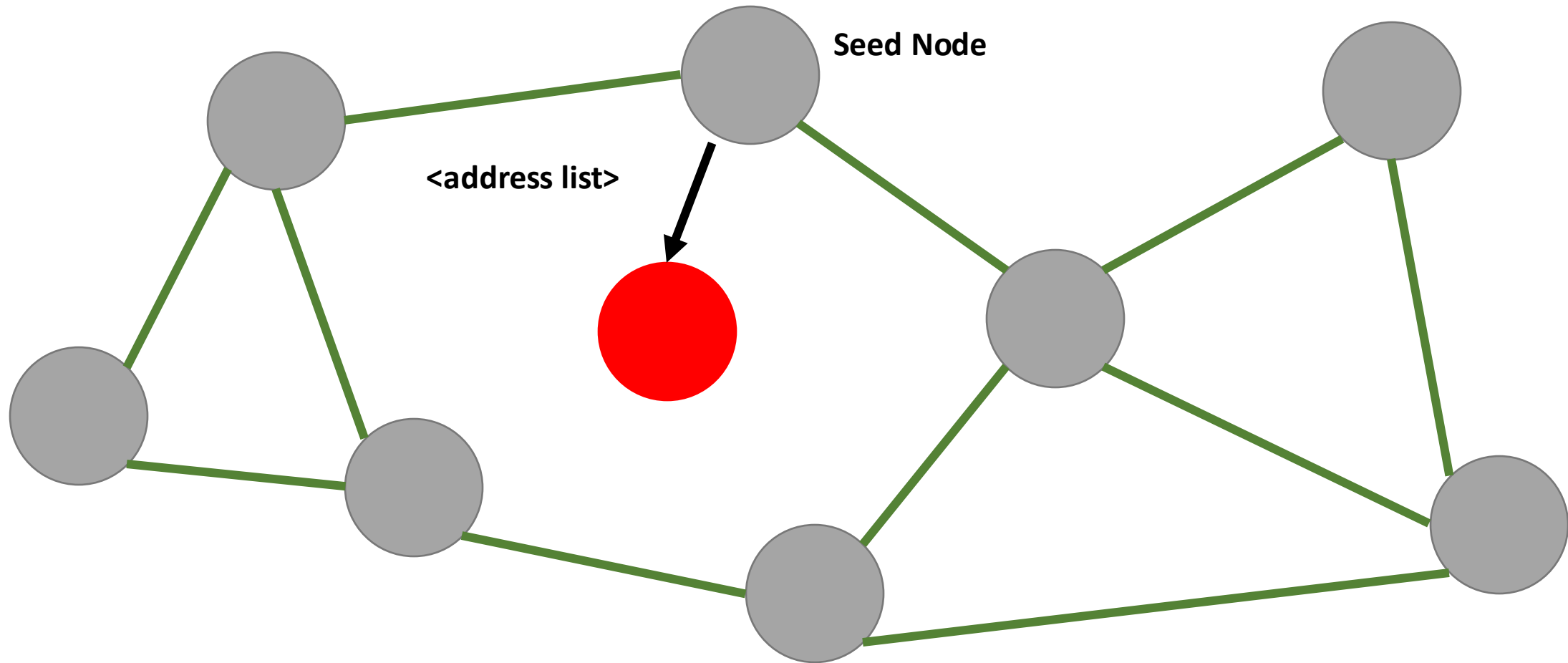
Joining in a Bitcoin P2P Network



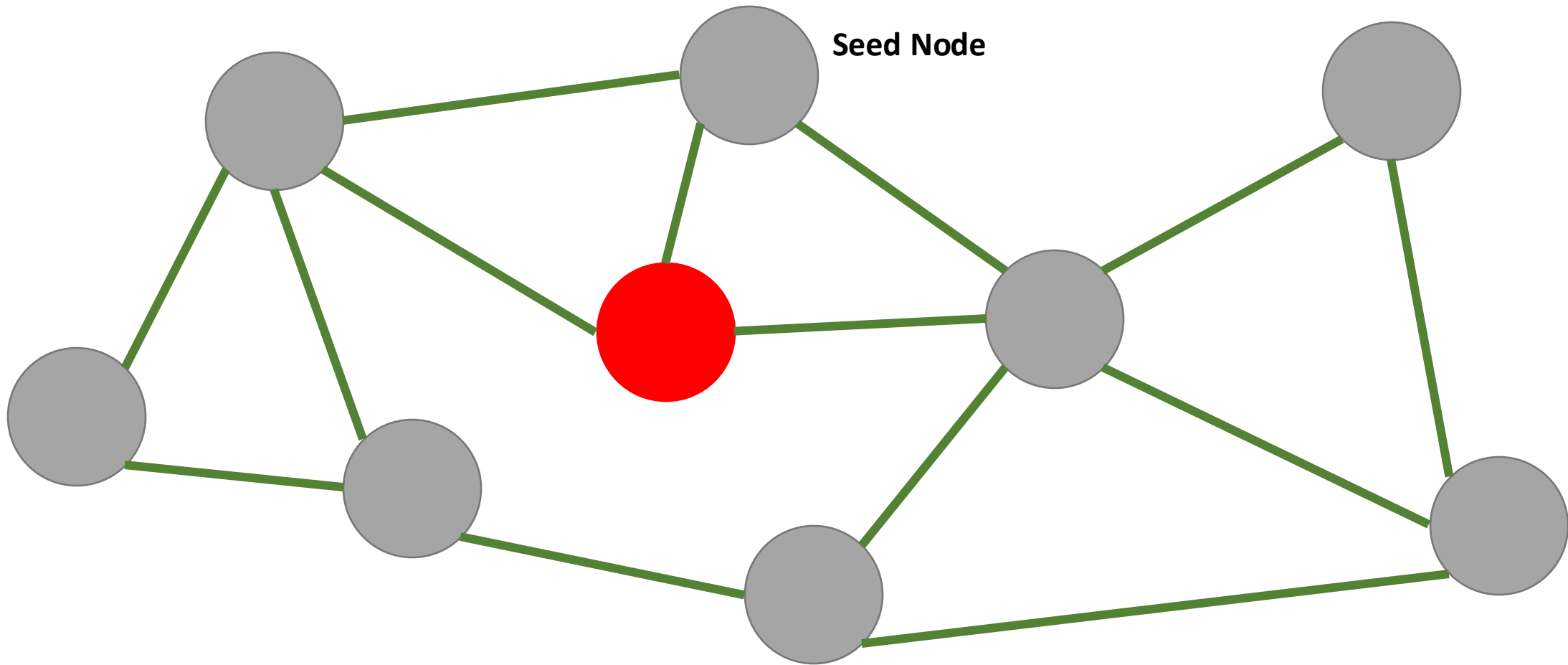
Joining in a Bitcoin P2P Network



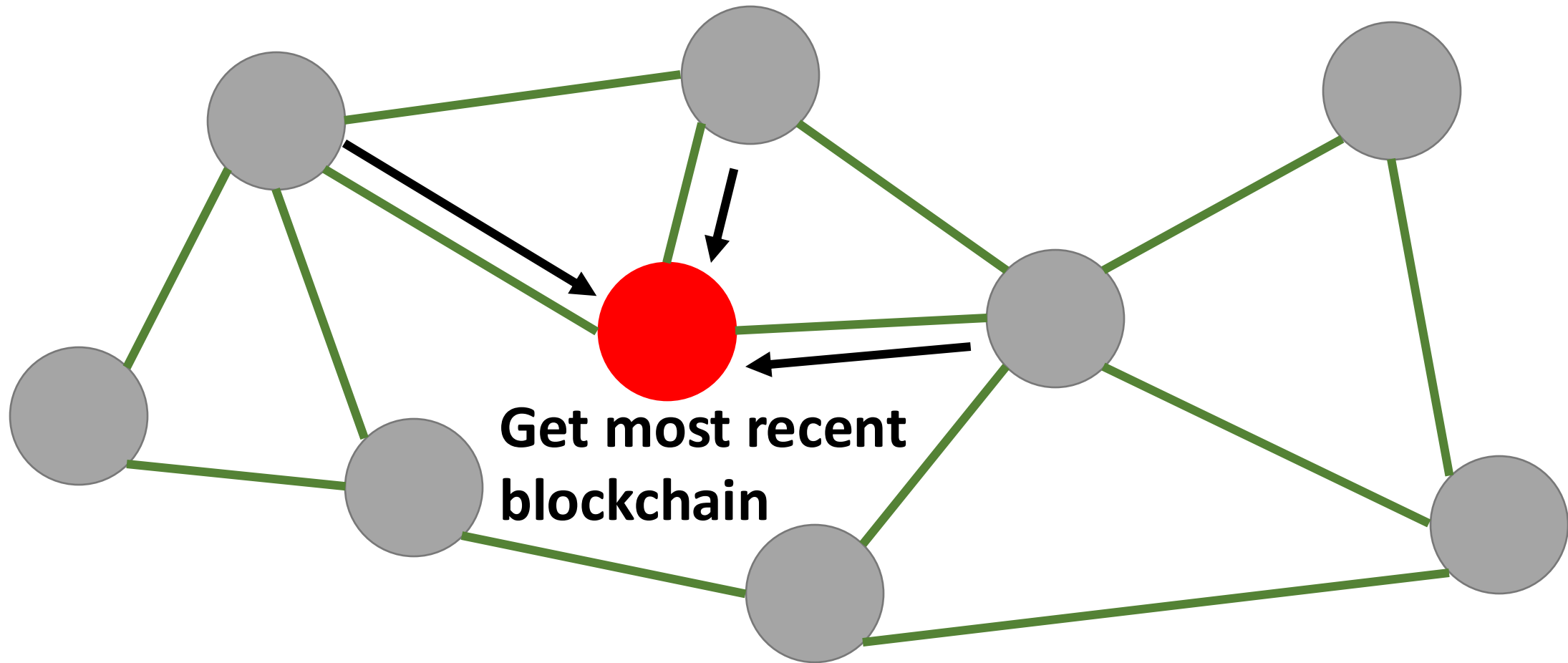
Joining in a Bitcoin P2P Network



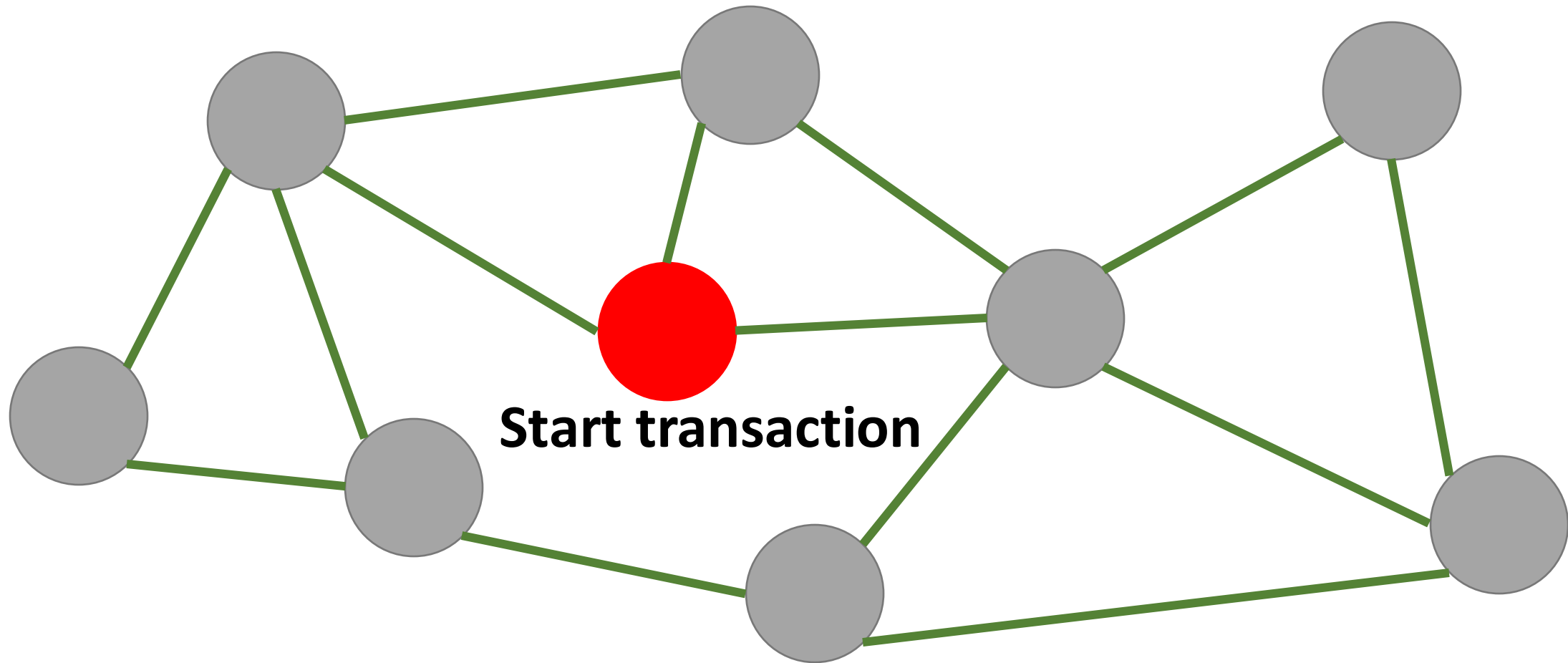
Joining in a Bitcoin P2P Network



Joining in a Bitcoin P2P Network

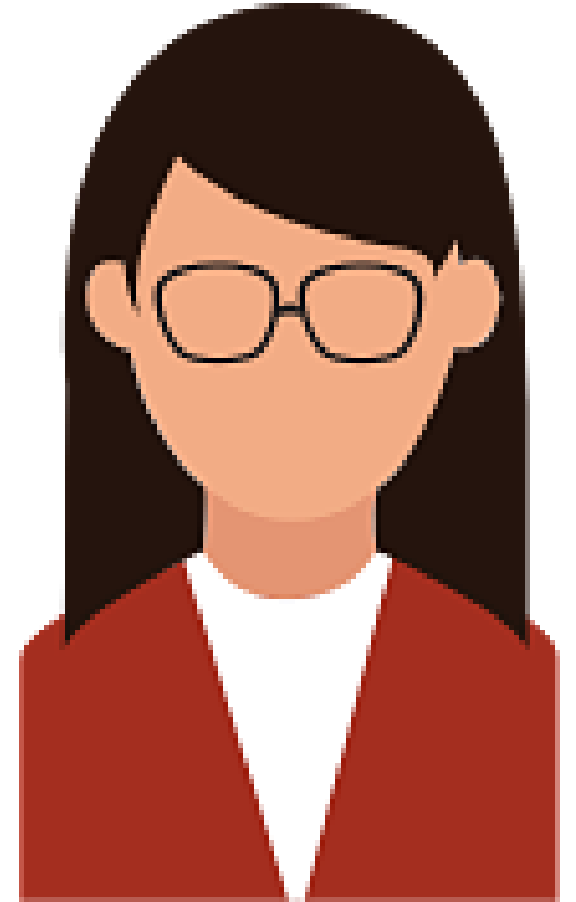


Joining in a Bitcoin P2P Network

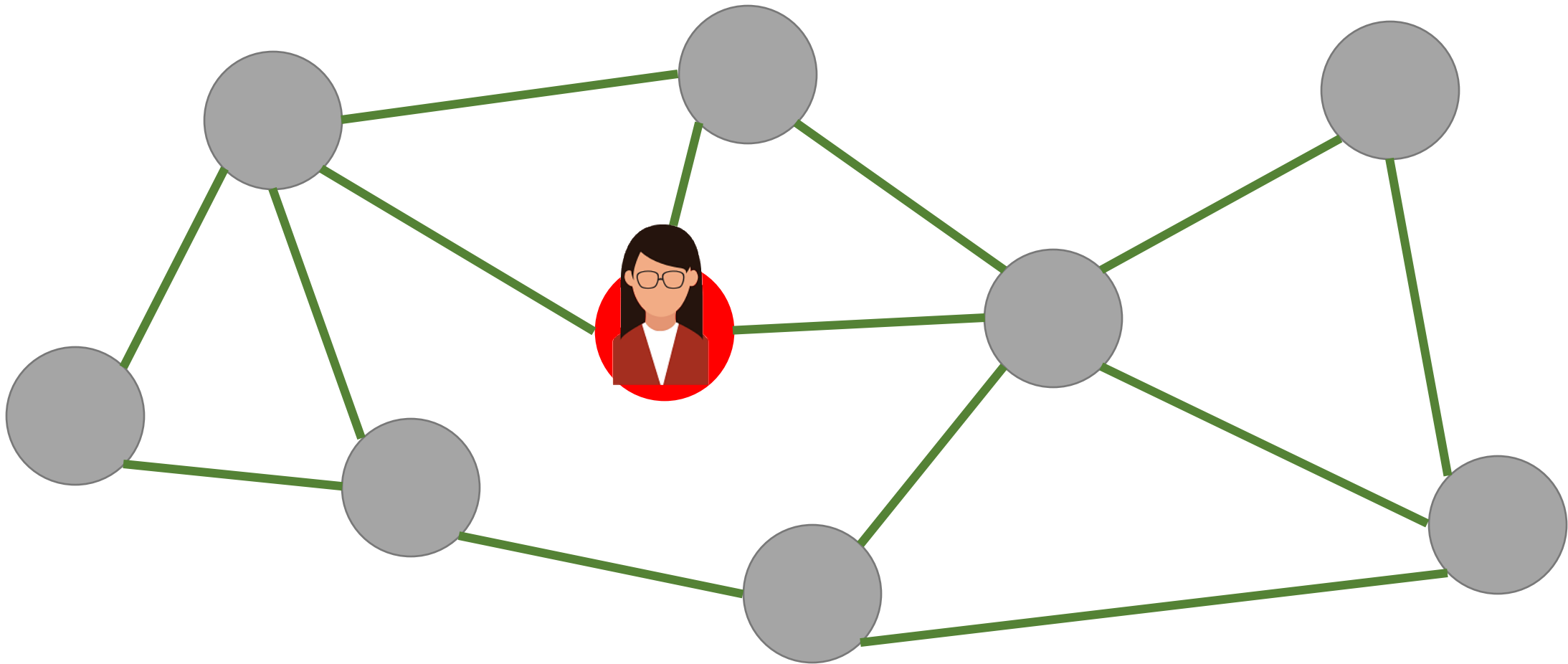


Transaction in a Bitcoin Network

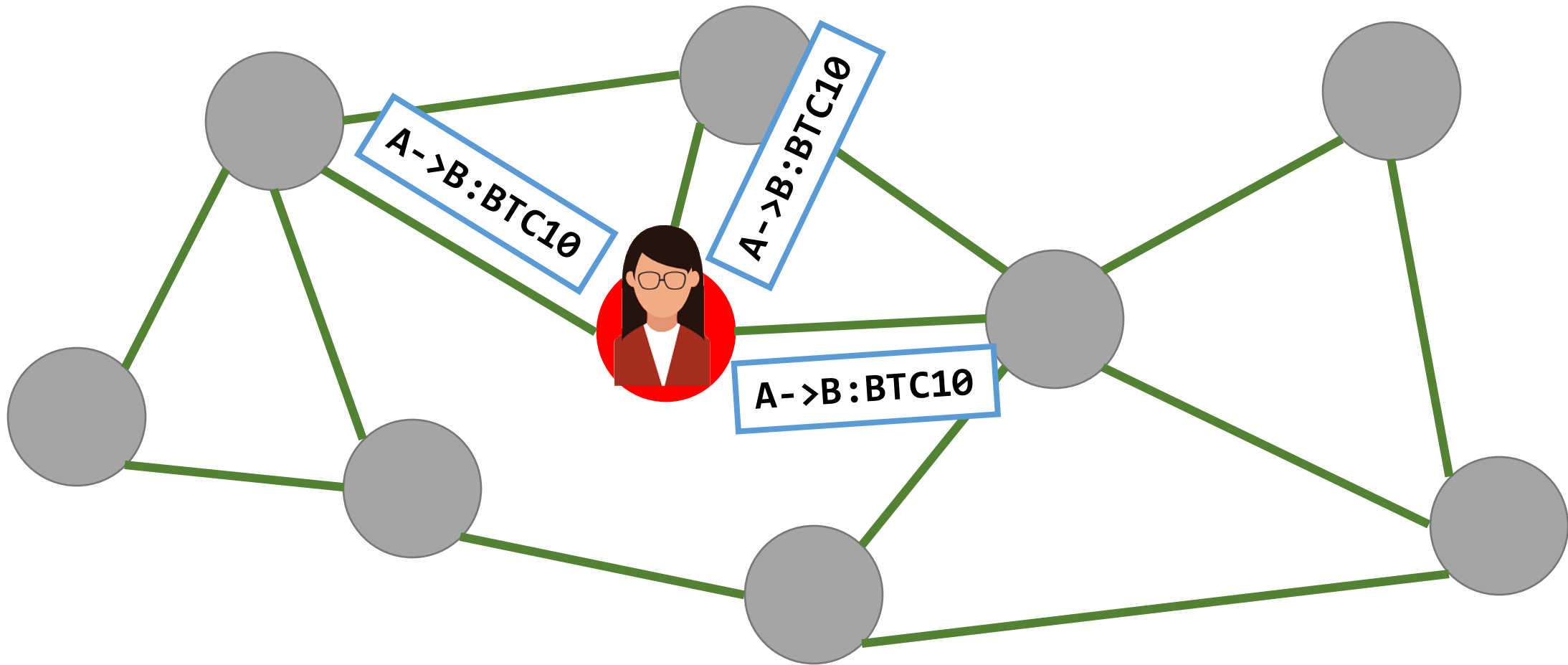
- Alice joins the Bitcoin network by opening her applet
- Alice makes a transaction to Bob: **A- > B: BTC 10**
- Alice includes the scripts with the transactions
- Alice broadcasts this transaction in the Bitcoin network



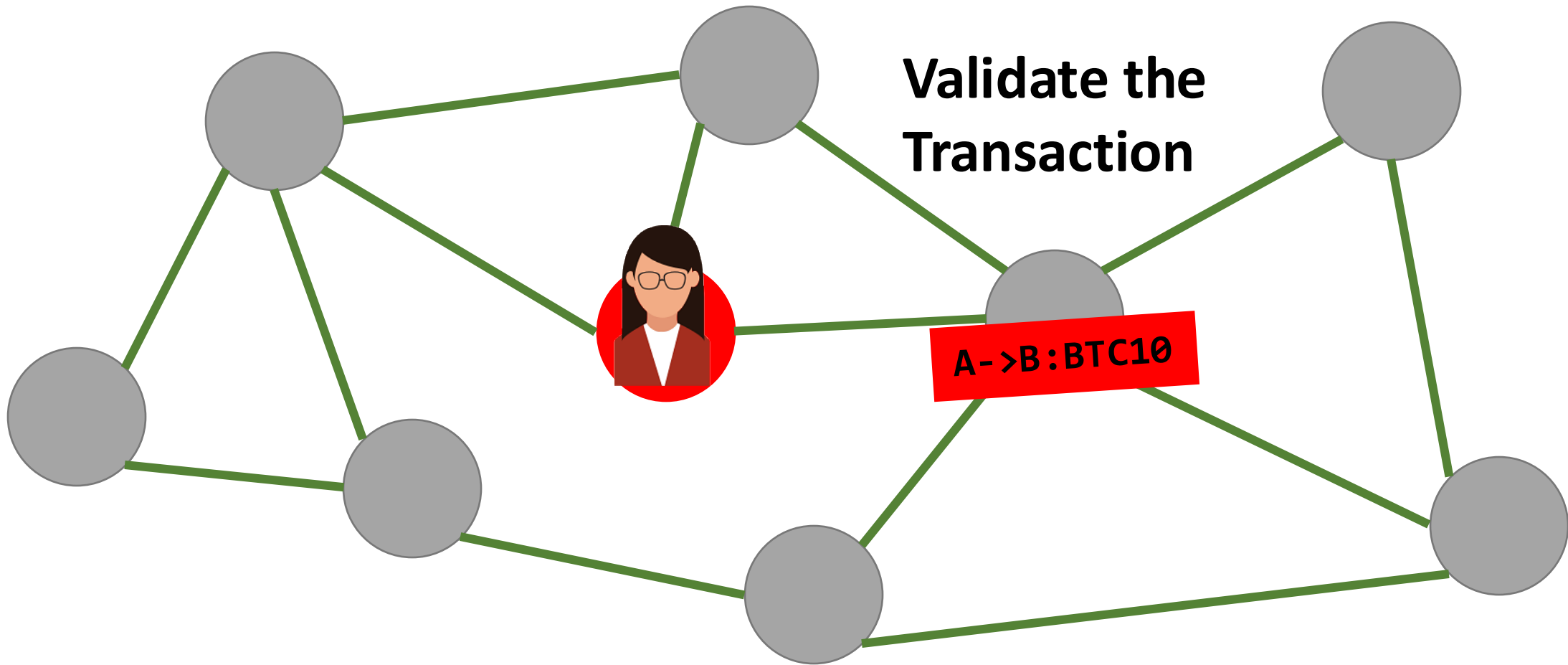
Transaction Flooding in a Bitcoin Network



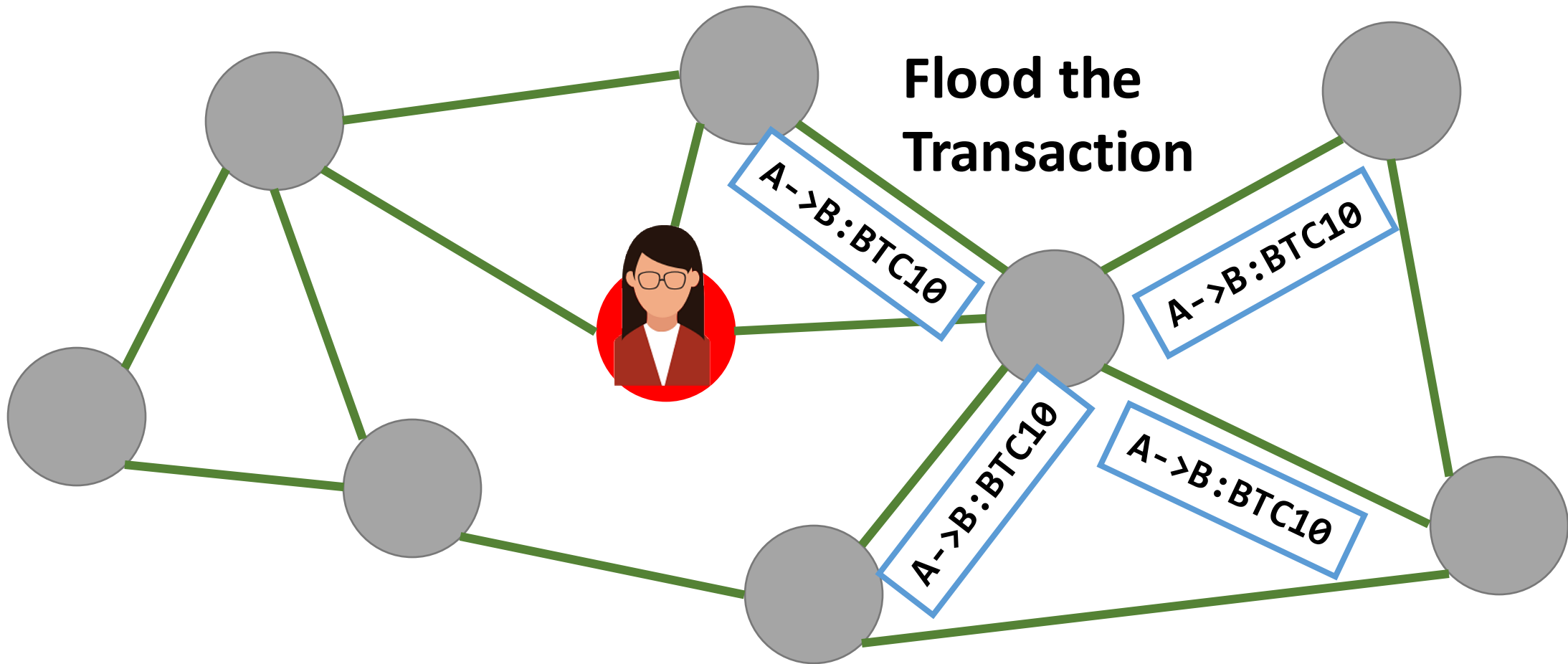
Transaction Flooding in a Bitcoin Network



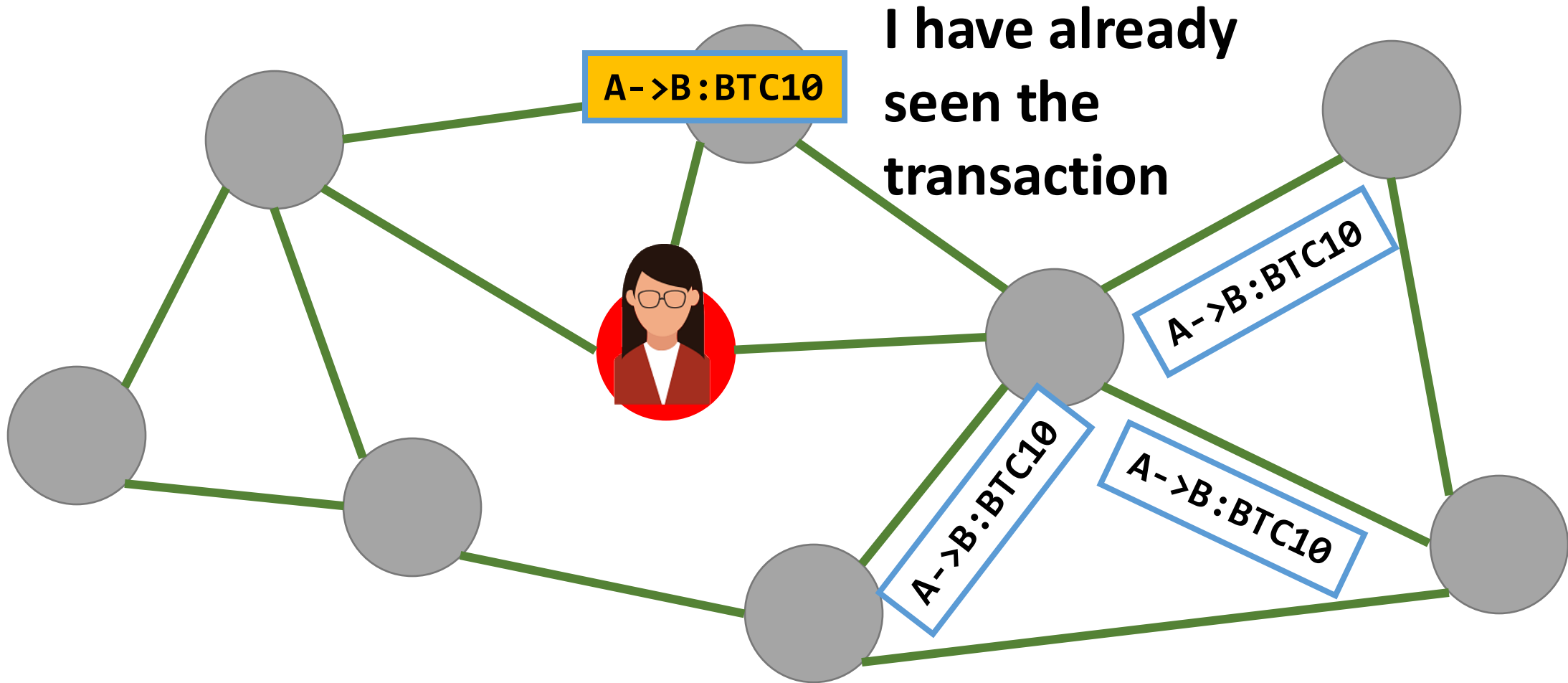
Transaction Flooding in a Bitcoin Network



Transaction Flooding in a Bitcoin Network



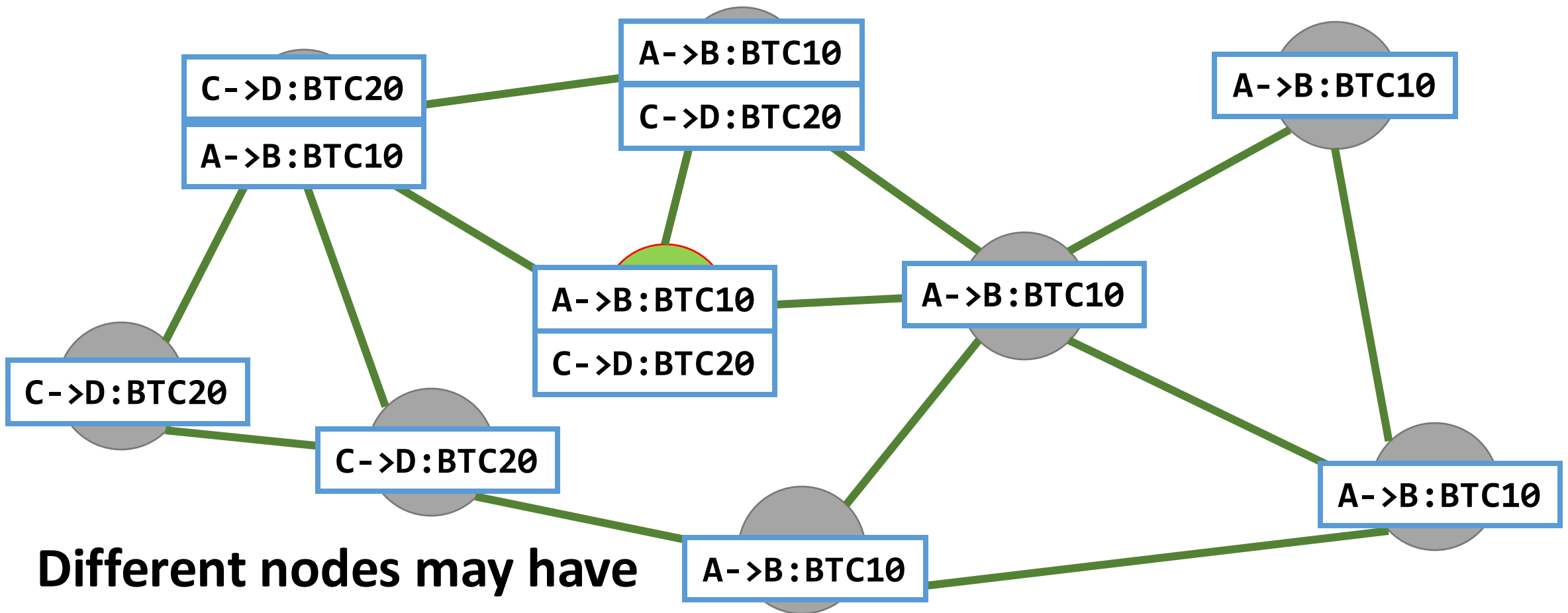
Transaction Flooding in a Bitcoin Network



Which Transactions Should You Relay?

- The transaction is valid with current blockchain
 - No conflict
 - No double spending
- The script matches with a pre-given set of whitelist scripts – avoid unusual scripts, avoid infinite loops
- Does not conflict with other transactions that I have relayed after getting the blockchain updated – avoid double spending

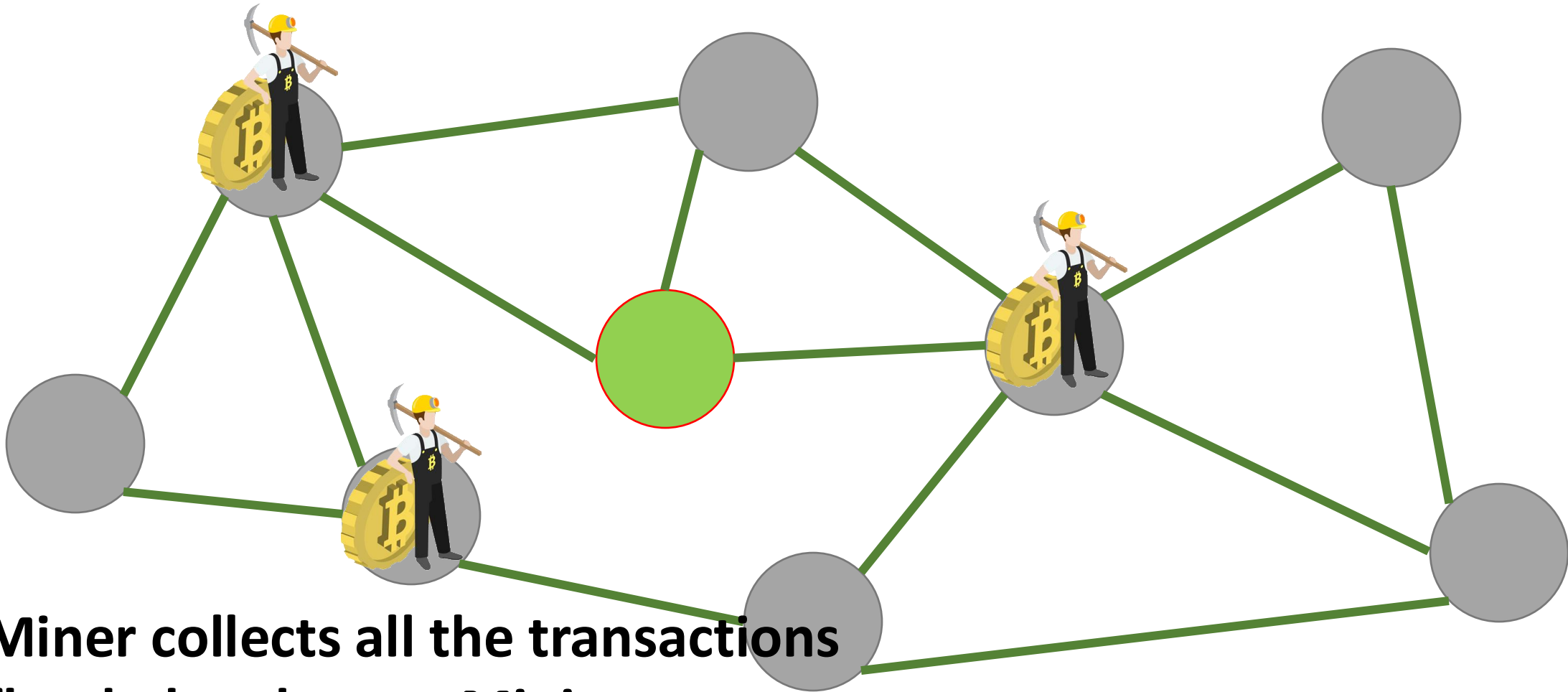
Transaction Flooding in a Bitcoin Network



Different nodes may have different transaction pools

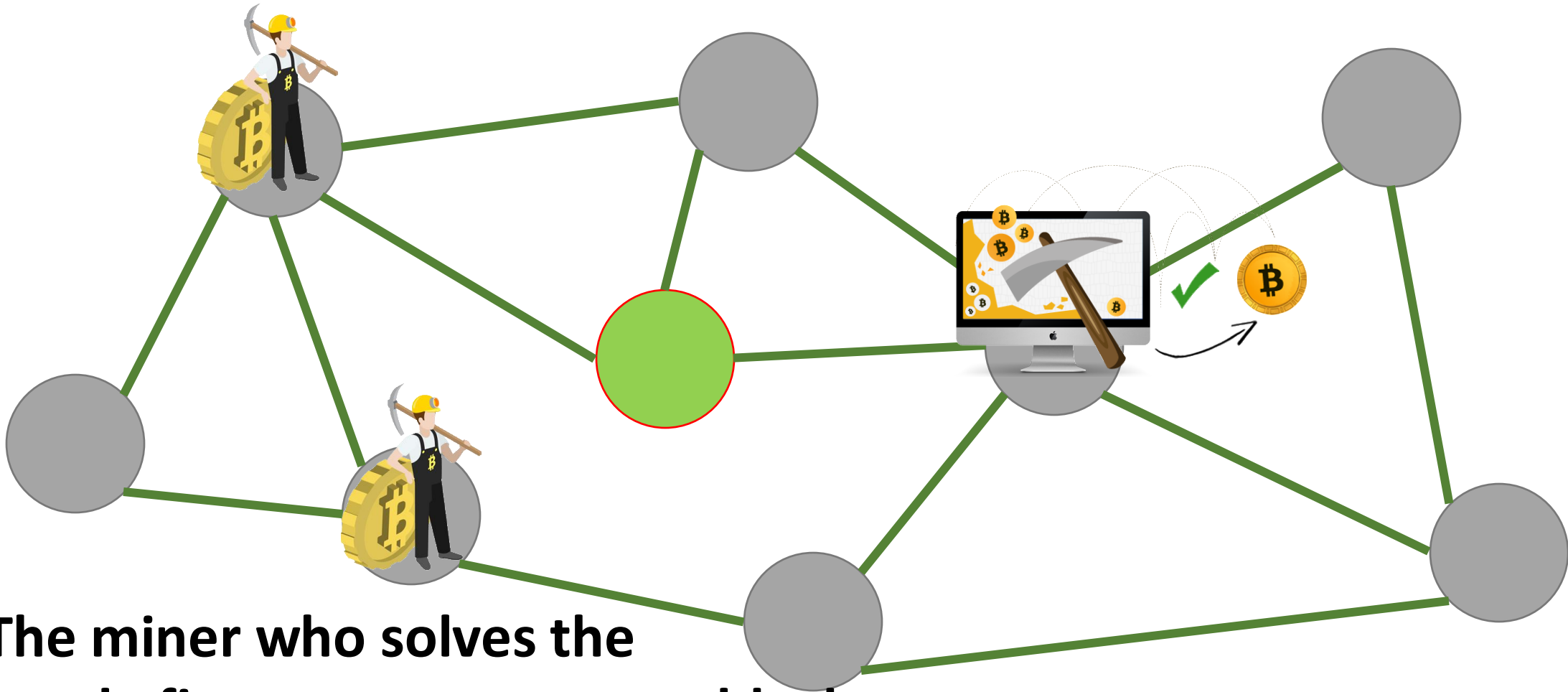
Accept the first set of transactions that you have heard

Mining in a Bitcoin Network



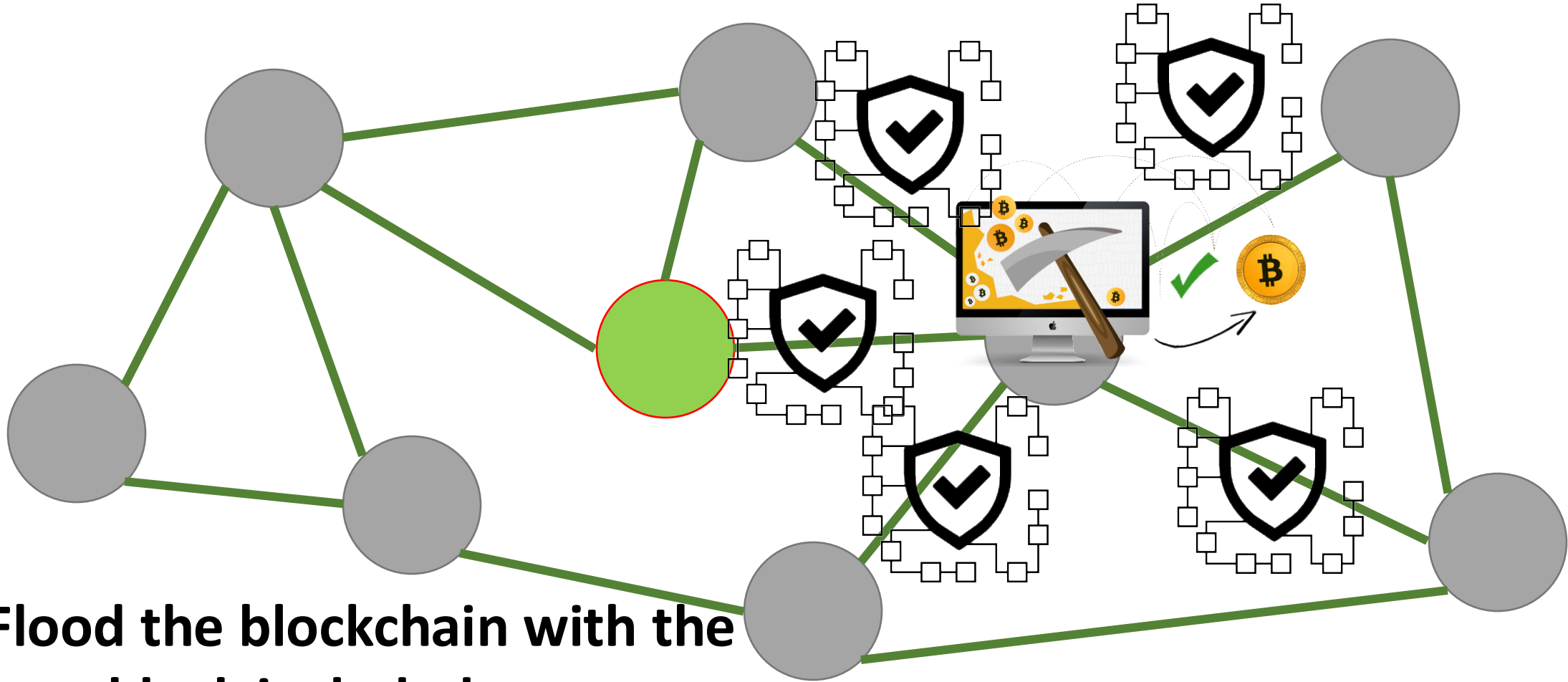
Miner collects all the transactions flooded and starts Mining

Block Generation



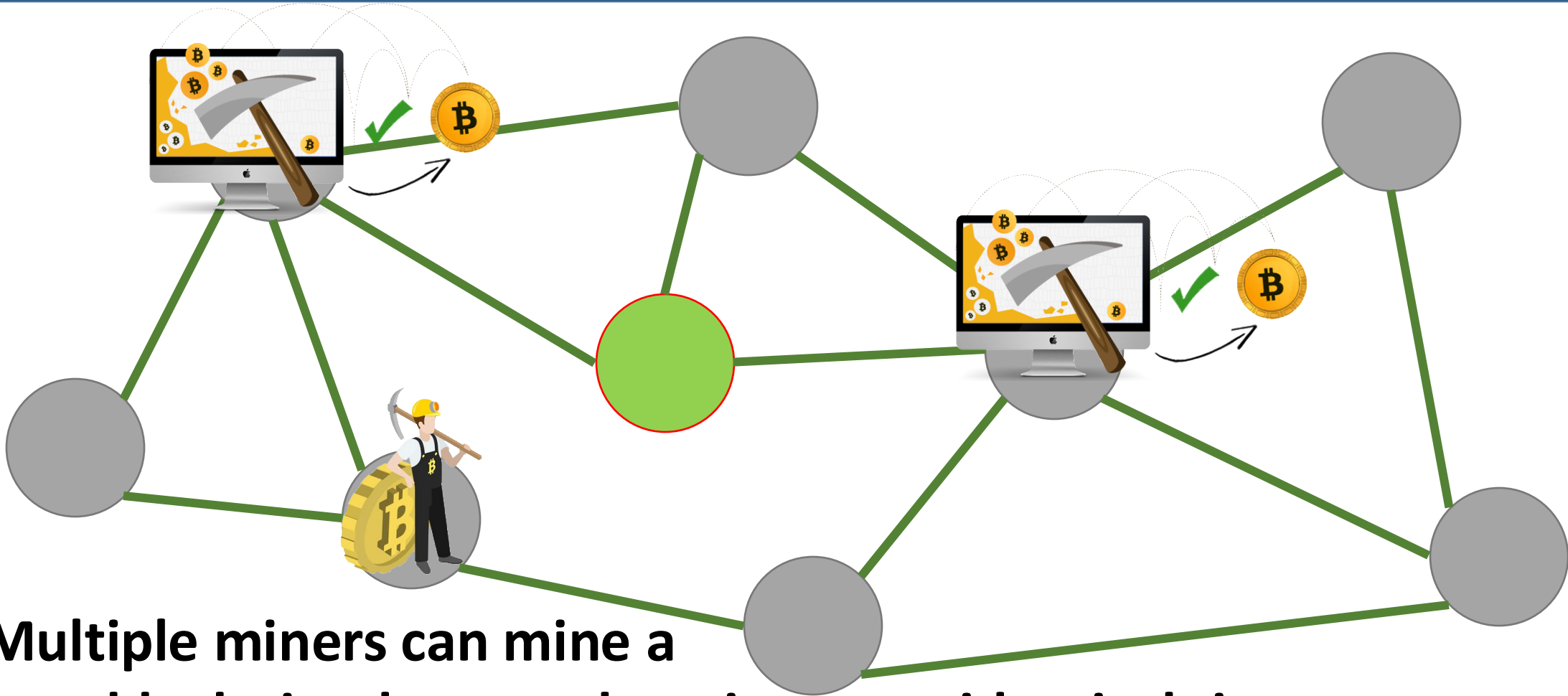
The miner who solves the puzzle first, generates a new block

Block Flooding



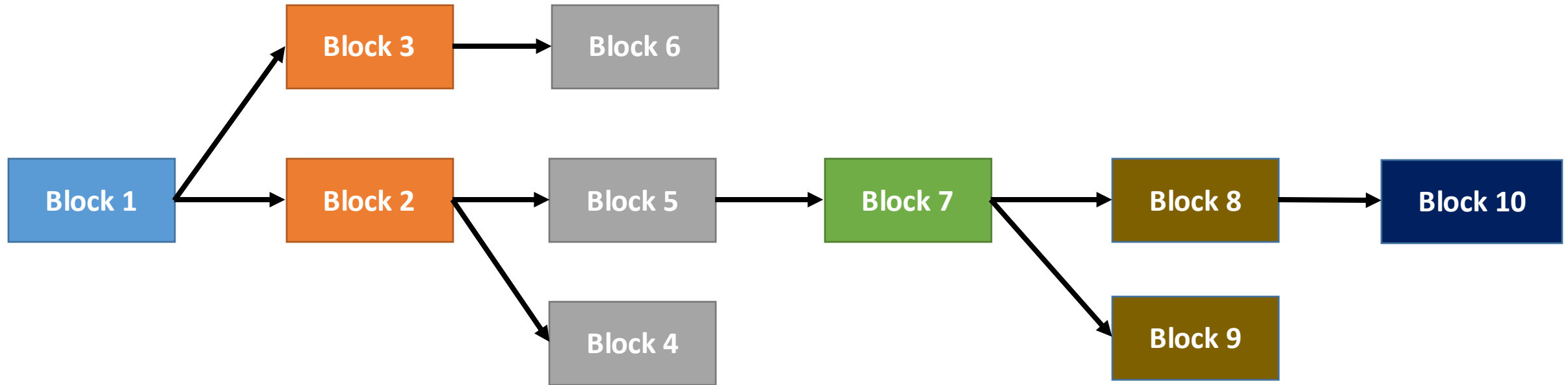
Flood the blockchain with the new block included

Block Propagation

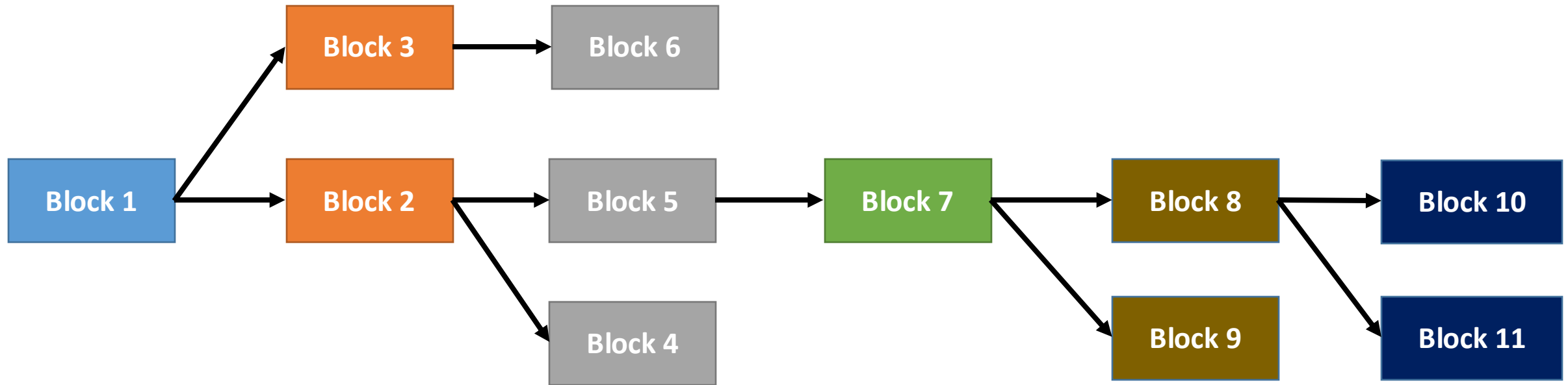


Multiple miners can mine a new block simultaneously or in a near identical time

Block Propagation – Accept the Longest Chain



Block Propagation – Accept One of the Longest Chains



“Accidental” forks occur rarely. Even if they occur, eventually only one becomes part of the longest chain

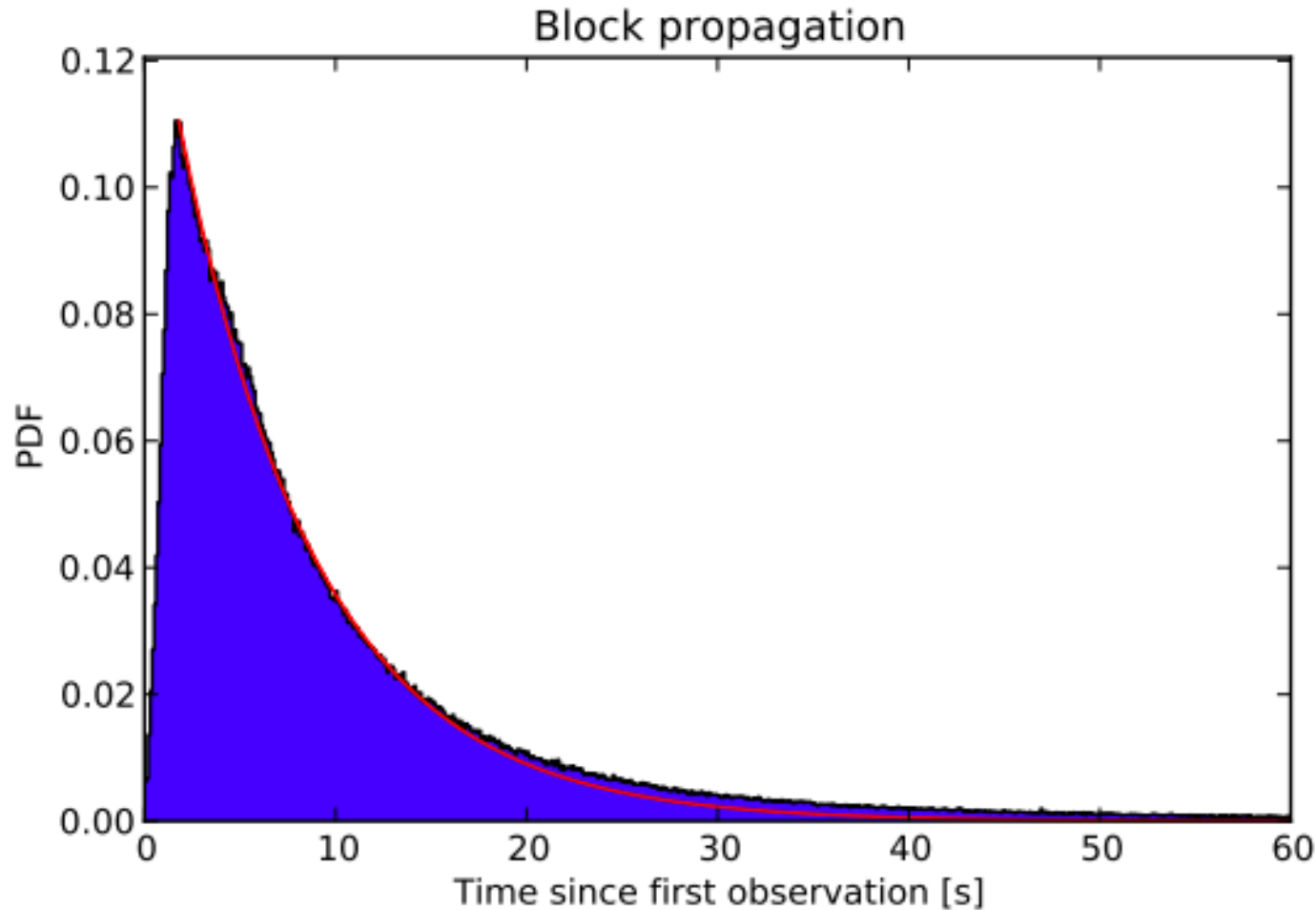
There are “intentional” forks of two type: hard forks and soft forks to come up with new versions like Bitcoin Cash, etc., or to upgrade software versions

Will be further discussed during our lectures on consensus

Which Block to Relay

- Block contains the correct hash based on the existing blockchain
- All the transactions inside the block are valid
 - Check the scripts
 - Validate with the existing blockchain
- The block is included in the current longest chain
 - Do not relay the forks

Block Propagation Latency



Mean time = 12.6 Seconds
**95% of the nodes can see
the block within 40 seconds**

Decker, Christian, and Roger Wattenhofer.
"Information propagation in the bitcoin
network." *2013 IEEE Thirteenth International
Conference on Peer-to-Peer Computing (P2P)*.
IEEE, 2013.

Bitcoin – The Beginning

- “A **decentralized** digital currency enables instant payments to anyone, anywhere in the world” – en.bitcoin.it
- No central authority, uses peer-to-peer technology
- Two broad operations
 - **Transaction Management** – transfer of bitcoins from one user to another
 - **Money Issuance** – regulate the monetary base

Bitcoin Basics – Creation of Coins

- **Controlled Supply:** Must be limited for the currency to have value – any maliciously generated currency needs to be rejected by the network
- Bitcoins are generated **during the mining** – each time a user discovers a new block
- The rate of block creation is adjusted every 2016 blocks to aim for a **constant two week adjustment period**
- The last bitcoin will be mined in 2140 (estimated and unless changed)

Bitcoin Basics – Creation of Coins

- The number of bitcoins generated per block is set to decrease **geometrically**, with a 50% reduction for every 210,000 blocks, or approximately 4 years
- This reduces, with time, the amount of bitcoins generated per block
 - Theoretical limit for total bitcoins: Slightly less than *21 million*
 - Miners will get less reward as time progresses
 - How to pay the mining fee – increase the transaction fee

Projected Bitcoins

| Date reached | Block | Reward Era | BTC/block | Year (estimate) | Start BTC | BTC Added | End BTC | BTC Increase | End BTC % of Limit |
|--------------|--------|------------|-----------|-----------------|-----------|-----------|----------|--------------|--------------------|
| 2009-01-03 | 0 | 1 | 50.00 | 2009 | 0 | 2625000 | 2625000 | infinite | 12.500% |
| 2010-04-22 | 52500 | 1 | 50.00 | 2010 | 2625000 | 2625000 | 5250000 | 100.00% | 25.000% |
| 2011-01-28 | 105000 | 1 | 50.00 | 2011* | 5250000 | 2625000 | 7875000 | 50.00% | 37.500% |
| 2011-12-14 | 157500 | 1 | 50.00 | 2012 | 7875000 | 2625000 | 10500000 | 33.33% | 50.000% |
| 2012-11-28 | 210000 | 2 | 25.00 | 2013 | 10500000 | 1312500 | 11812500 | 12.50% | 56.250% |
| 2013-10-09 | 262500 | 2 | 25.00 | 2014 | 11812500 | 1312500 | 13125000 | 11.11% | 62.500% |
| 2014-08-11 | 315000 | 2 | 25.00 | 2015 | 13125000 | 1312500 | 14437500 | 10.00% | 68.750% |
| 2015-07-29 | 367500 | 2 | 25.00 | 2016 | 14437500 | 1312500 | 15750000 | 9.09% | 75.000% |
| 2016-07-09 | 420000 | 3 | 12.50 | 2016 | 15750000 | 656250 | 16406250 | 4.17% | 78.125% |
| 2017-06-23 | 472500 | 3 | 12.50 | 2018 | 16406250 | 656250 | 17062500 | 4.00% | 81.250% |
| | 525000 | 3 | 12.50 | 2019 | 17062500 | 656250 | 17718750 | 3.85% | 84.375% |
| | 577500 | 3 | 12.50 | 2020 | 17718750 | 656250 | 18375000 | 3.70% | 87.500% |
| | 630000 | 4 | 6.25 | 2021 | 18375000 | 328125 | 18703125 | 1.79% | 89.063% |
| | 682500 | 4 | 6.25 | 2022 | 18703125 | 328125 | 19031250 | 1.75% | 90.625% |
| | 735000 | 4 | 6.25 | 2023 | 19031250 | 328125 | 19359375 | 1.72% | 92.188% |
| | 787500 | 4 | 6.25 | 2024 | 19359375 | 328125 | 19687500 | 1.69% | 93.750% |

Information Source: <https://en.bitcoin.it/wiki/>

Bitcoin Basics – Sending Payments

- Alice wants to send bitcoin to Bob
 - Bob sends his address to Alice
 - Alice adds Bob's address and the amount of bitcoins to transfer in a "transaction" message
 - Alice signs the transaction with her private key, and announces her public key for signature verification
 - Alice broadcasts the transaction on the Bitcoin network for all to see

Handle Double Spending using Blockchain

- When multiple valid continuation to this chain appear, only the longest such branch is accepted and it is then extended further (**longest chain**)
- Once a transaction is committed in the blockchain, everyone in the network can validate all the transactions by using Alice's public address
- The validation prevents double spending in bitcoin

Bitcoin Anonymity

- Bitcoin is permission-less, you do not need to setup any “account”, or required any e-mail address, user name or password to login to the wallet
- The public and the private keys do not need to be registered, the wallet can generate them for the users
- The **bitcoin address** is used for transaction, not the user name or identity
- A **bitcoin address** mathematically corresponds to a public key based on ECDSA – the digital signature algorithm used in bitcoin
- A sample bitcoin address: 1PHYrmdJ22MKbJevpb3MBNpVckjZHt89hz
- Each person can have many such addresses, each with its own balance
 - Difficult to know which person owns what amount

To Sum it All Up!!

- Bitcoins do not really “exist” as any tangible or electronic object.
- There is no bit”coin” as you see in its logo
- Owning a bitcoin simply means you have access to a key pair that includes
 - A public key to which somebody else had sent some bitcoin
 - A matching private key that gives you the authority to send the previously received bitcoin to another address
- If you lose your private key, you lose the corresponding bitcoin(s)

Physical Payment using Bitcoin

- All that is needed is a (set of) private key(s) – Public key can be generated from the private key.
- Safely store the private key – in your desktop, on the web, mobile phone, special hardware attachment, printed on a piece of paper as QR
- For online payment, you can use the wallet and an appropriate mode of applying the private key
- For off line payments like in store payments or paying to your friend, you can use your mobile phone to present the private key or use the hardcopy!! As simple as using PayTm, Google Pay and so on.

Bitcoin Exchange

- Trading bitcoin as commodity
- Centralized exchanges – (In India: WazirX, CoinDCX, Zebpay, CoinSwitch Kuber, etc.)
 - Identity verification using KYC documents
 - Maintain your balance in Bitcoin and another currency like INR.
 - You set the buying and selling prices and quantities
 - If necessary, you can take the money out in a preferred currency
 - Some exchanges provide the payout option in anonymous prepaid cards
- There can also be decentralized exchanges with appropriate procedures for handling similar requirements

thank you!