How Bitcoin Works

Kai Brünnler

Research Institute for Security in the Information Society
Bern University of Applied Sciences

What is Bitcoin?

Bitcoin

- an open-source software
- a peer-to-peer network
- a decentralized payment network
- a decentralized currency

Problems with Centralization

- Payment Networks
  - censorship
  - fees
  - chargebacks
  - identity theft
  - onramp cost
- Currencies
  - inflation

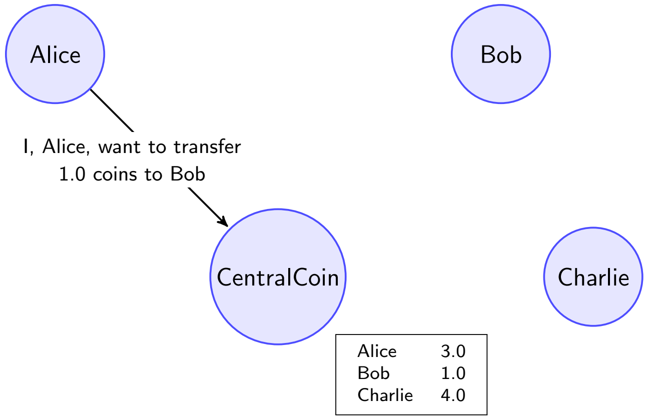
Outline

1. CentralCoin
2. NaiveCoin
3. SerialNumberCoin
4. PublicAnnouncementCoin
5. ElectionCoin
6. PuzzleCoin
7. BlockchainCoin
8. Bitcoin

Outline

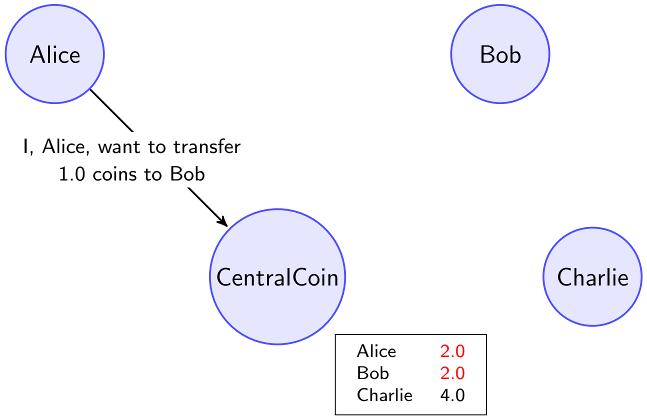
1. CentralCoin
2. NaiveCoin
3. SerialNumberCoin
4. PublicAnnouncementCoin
5. ElectionCoin
6. PuzzleCoin
7. BlockchainCoin
8. Bitcoin

CentralCoin



Alice	3.0
Bob	1.0
Charlie	4.0

CentralCoin



Alice	2.0
Bob	2.0
Charlie	4.0

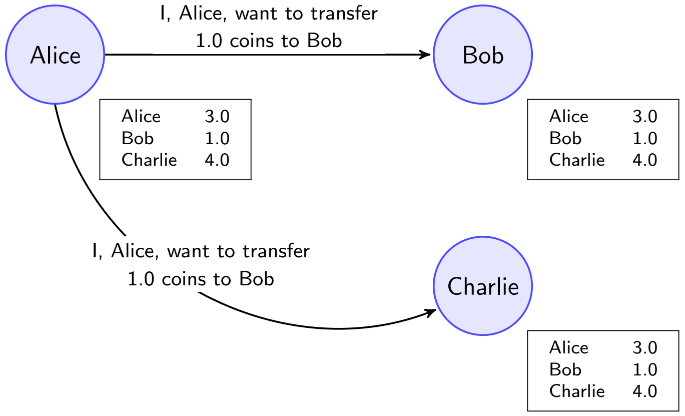
Outline

1. CentralCoin
2. NaiveCoin
3. SerialNumberCoin
4. PublicAnnouncementCoin
5. ElectionCoin
6. PuzzleCoin
7. BlockchainCoin
8. Bitcoin

NaiveCoin

- every node keeps ledger
- transactions are broadcast to all nodes
- every node accepts all valid transactions it receives

NaiveCoin



Alice → Bob: I, Alice, want to transfer 1.0 coins to Bob

Alice → Charlie: I, Alice, want to transfer 1.0 coins to Bob

Alice:

Alice	3.0
Bob	1.0
Charlie	4.0

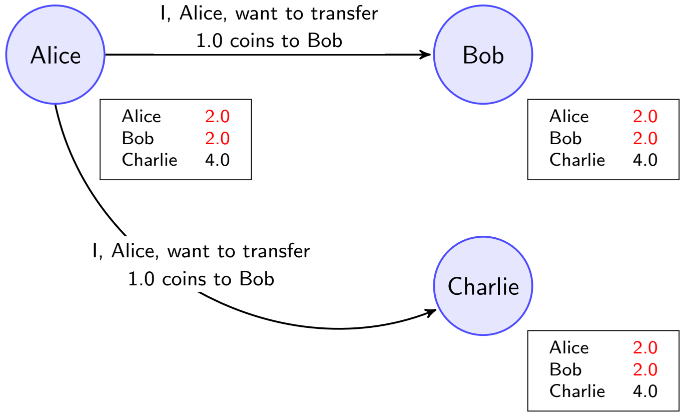
Bob:

Alice	3.0
Bob	1.0
Charlie	4.0

Charlie:

Alice	3.0
Bob	1.0
Charlie	4.0

NaiveCoin



Alice → Bob: I, Alice, want to transfer 1.0 coins to Bob

Alice → Charlie: I, Alice, want to transfer 1.0 coins to Bob

Alice:

Alice	2.0
Bob	2.0
Charlie	4.0

Bob:

Alice	2.0
Bob	2.0
Charlie	4.0

Charlie:

Alice	2.0
Bob	2.0
Charlie	4.0

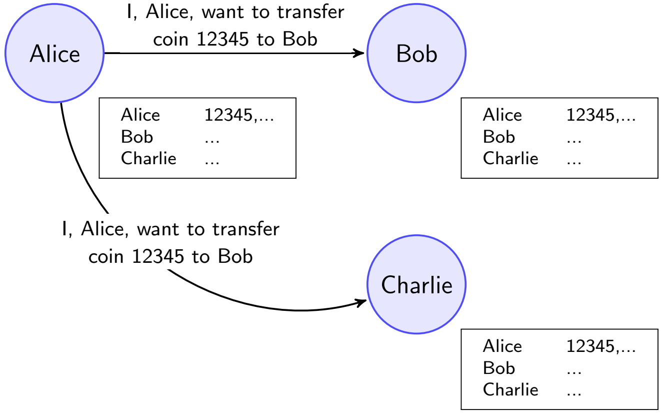
Outline

1. CentralCoin
2. NaiveCoin
3. SerialNumberCoin
4. PublicAnnouncementCoin
5. ElectionCoin
6. PuzzleCoin
7. BlockchainCoin
8. Bitcoin

SerialNumberCoin

- as before but coins have serial numbers

SerialNumberCoin



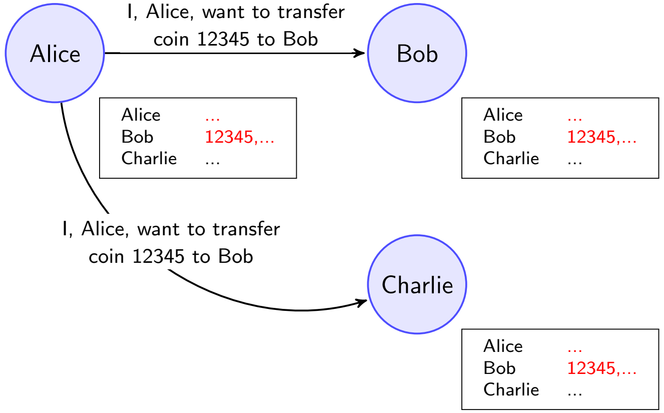
Alice
I, Alice, want to transfer coin 12345 to Bob
Bob
I, Alice, want to transfer coin 12345 to Bob
Charlie

Alice	12345,...
Bob	...
Charlie	...

Alice	12345,...
Bob	...
Charlie	...

Alice	12345,...
Bob	...
Charlie	...

SerialNumberCoin



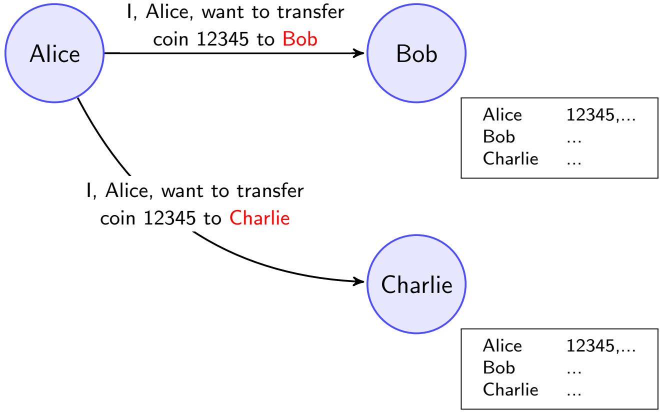
Alice
I, Alice, want to transfer coin 12345 to Bob
Bob
I, Alice, want to transfer coin 12345 to Bob
Charlie

Alice	...
Bob	12345,...
Charlie	...

Alice	...
Bob	12345,...
Charlie	...

Alice	...
Bob	12345,...
Charlie	...

The Double Spending Attack

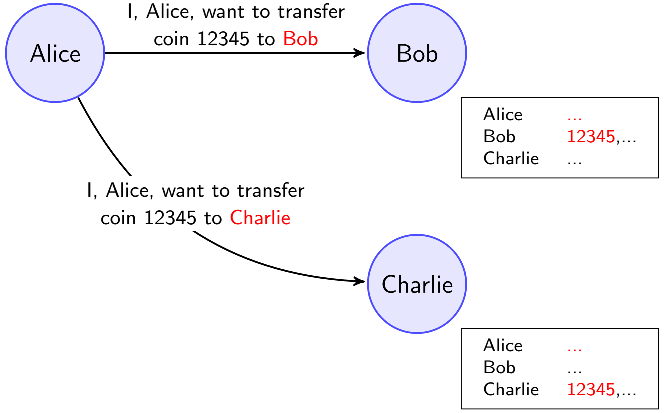


Alice
I, Alice, want to transfer coin 12345 to Bob
Bob
I, Alice, want to transfer coin 12345 to Charlie
Charlie

Alice	12345,...
Bob	...
Charlie	...

Alice	12345,...
Bob	...
Charlie	...

The Double Spending Attack



Alice
I, Alice, want to transfer coin 12345 to Bob
Bob
I, Alice, want to transfer coin 12345 to Charlie
Charlie

Alice	...
Bob	12345,...
Charlie	...

Alice	...
Bob	...
Charlie	12345,...

Outline

1. CentralCoin
2. NaiveCoin
3. SerialNumberCoin
4. PublicAnnouncementCoin
5. ElectionCoin
6. PuzzleCoin
7. BlockchainCoin
8. Bitcoin

PublicAnnouncementCoin

Protocol is as before, but now:

- instead of sending a transaction to everybody, a transaction is publicly announced
- and everybody only accepts transactions that are publicly announced

Public Announcement is very different from just sending to everybody:

- not just everybody knows,
- but everybody knows and everybody knows that everybody knows, etc.!

The double spending attack is now impossible.

- that is essentially Bitcoin...
- ...but how to implement public announcements on the internet?

Outline

1. CentralCoin
2. NaiveCoin
3. SerialNumberCoin
4. PublicAnnouncementCoin
5. ElectionCoin
6. PuzzleCoin
7. BlockchainCoin
8. Bitcoin

ElectionCoin

Protocol is like SerialNumberCoin, but now:

- every node keeps all received transactions in the unconfirmed transaction pool
- every 10 minutes nodes randomly elect a leader (say that's possible)
- the leader node updates its ledger according to its transaction pool and broadcasts the ledger
- all nodes take over the ledger from the leader and discard their transaction pool

- Problem: Sybil Attack

Outline

1. CentralCoin
2. NaiveCoin
3. SerialNumberCoin
4. PublicAnnouncementCoin
5. ElectionCoin
6. PuzzleCoin
7. BlockchainCoin
8. Bitcoin

PuzzleCoin

Protocol is like before, but now:

- all nodes try to solve a hard computational puzzle
- a node that solved it
  - updates its ledger according to its transaction pool
  - broadcasts the updated ledger together with the puzzle solution
- a node takes over a ledger if it can verify the puzzle solution
- to incentivize nodes to do the hard computational work, they are rewarded with coins for solving a puzzle
- the puzzle: find a number, called nonce, which, when hashed, gives a bitstring starting with a number of zeros

Now a Sybil attack is hard.

- Problem: What if two nodes find a solution at roughly the same time?

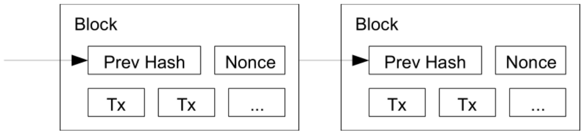
Outline

1. CentralCoin
2. NaiveCoin
3. SerialNumberCoin
4. PublicAnnouncementCoin
5. ElectionCoin
6. PuzzleCoin
7. BlockchainCoin
8. Bitcoin

BlockchainCoin

Protocol is like before, but now:

- each node not only stores balances, but the entire transaction history
- a node that solves a puzzle broadcasts its block of transactions and includes the nonce and the hash of the previous block
- each node takes over the longest available valid chain of blocks
- the puzzle: find a nonce, which, when hashed together with the transactions and the previous hash, gives a bitstring starting with a number of zeros



BlockchainCoin

Now it's no problem if two nodes solve the puzzle at the same time:

- Alice and Bob both find the nonce at the same time
- half the network takes over Alice's block and the other half Bob's block
- at some point someone will find the next block,
- let's say it's Charlie, and Charlie is in Alice' part of the network
- everybody (including Bob) will take over Charlie's and thus Alice' block, because it forms the longest chain

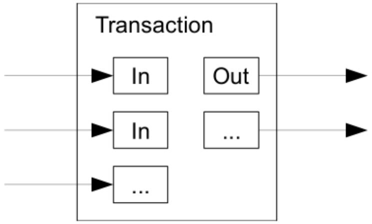
Problem: Transacting single coins is cumbersome.

Outline

1. CentralCoin
2. NaiveCoin
3. SerialNumberCoin
4. PublicAnnouncementCoin
5. ElectionCoin
6. PuzzleCoin
7. BlockchainCoin
8. Bitcoin

Bitcoin

- there are no coins and no serial numbers in Bitcoin
- there are only transactions with inputs and outputs, each input is the output of a previous transaction



A Transaction

```
{
 "hash":"7c4025...",
 "ver":1,
 "vin_sz":1,
 "vout_sz":1,
 "lock_time":0,
 "size":224,
 "in":[{"prev_out":{"hash":"2007ae...","n":0},
        "scriptSig":"304502... 042b2d..."}],
 "out":[{"value":"0.31900000",
         "scriptPubKey":"OP_DUP OP_HASH160 a7db6f OP_EQUALVERIFY OP_CHECKSIG"}]
}
```

Bitcoin Script

- `scriptSig`: Part of an input, unlocks the referenced output from a previous transaction
  Example: `<signature>`
- `scriptPubKey`: Part of an output, locks that output
  Example: `<pubkey> OP_CHECKSIG`
- when a transaction input tries to spend an output, essentially the `scriptSig` and `ScriptPubkey` are concatenated and run, if in the end `true` is on the stack, the input is valid

Bitcoin Script allows for applications like:

- both of these two given keys need to sign
- at least two of those three given keys need to sign

Other Blockchain Uses, Outlook

- incorruptible and fairly cheap registry: land registries, notary services etc.
- p2p tradable assets: stocks, art, luxury items, local currencies
- smart contracts (unstoppable programs that control funds): escrow, prediction markets, p2p gaming

Sources

- Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf
- Michael Nielsen. How the Bitcoin protocol actually works. http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/
- Andreas Antonopoulos. Mastering Bitcoin. O'Reilly Media, 2014, http://shop.oreilly.com/product/0636920032281.do