Introduction to Blockchain

CryptoSoc - March 2019

By Thomas Tumiel

What is Money?

CryptoSoc

Recording Transactions

- Digital money needs proof
- Otherwise anyone can claim that they have/have not been paid

Friendly Ledger
Alice pays Bob R100
Bob pays Jenny R200
Sally pays Bob R300
Jenny pays Alice R150

From	To	Amount
Alice	Bob	R100
Bob	Jenny	R200
Sally	Bob	R300
Jenny	Alice	R150

CryptoSoc

Problems?

- Who controls the ledger?
- What if someone owes a lot of money but doesn’t pay?
- How do we know if someone actually agreed to their money being spent?
- Can someone send the same money to different people?

CryptoSoc

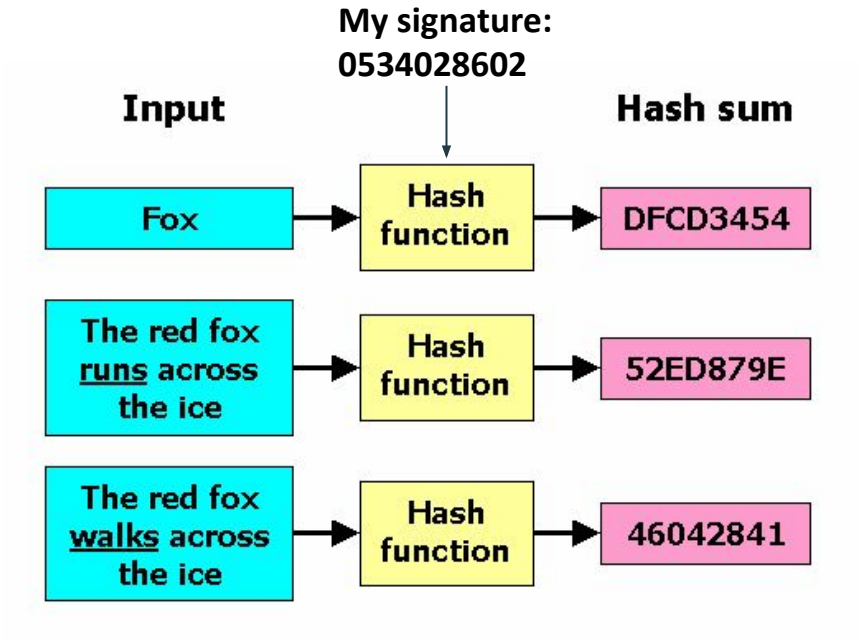
Cryptographic hash function

From	To	Amount	Proof
Alice	Bob	R100	252367
Bob	Jenny	R200	432166
Sally	Bob	R300	456772
Jenny	Alice	R150	357556

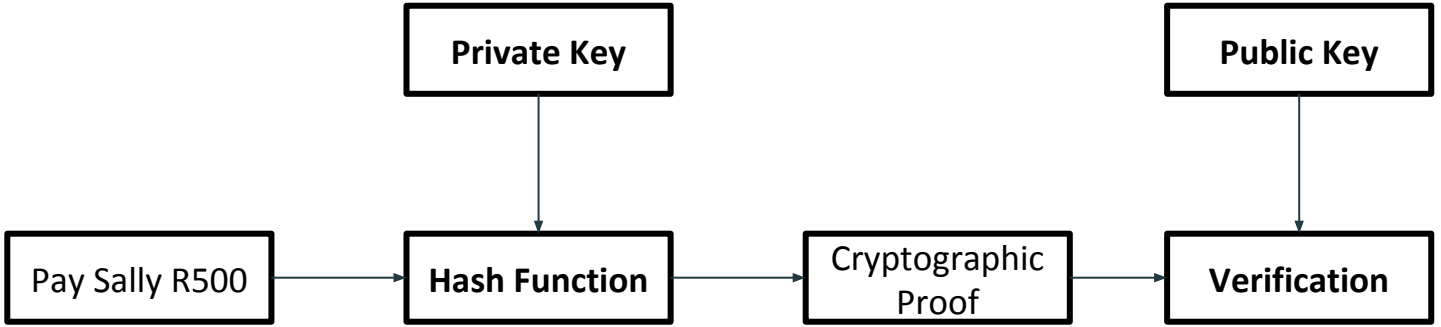
From	To	Amount	Proof	Verification
Alice	Bob	R100	252367	346711
Bob	Jenny	R200	432166	523098
Sally	Bob	R300	456772	556734
Jenny	Alice	R150	357556	869036

CryptoSoc

Cryptographic Hash Function



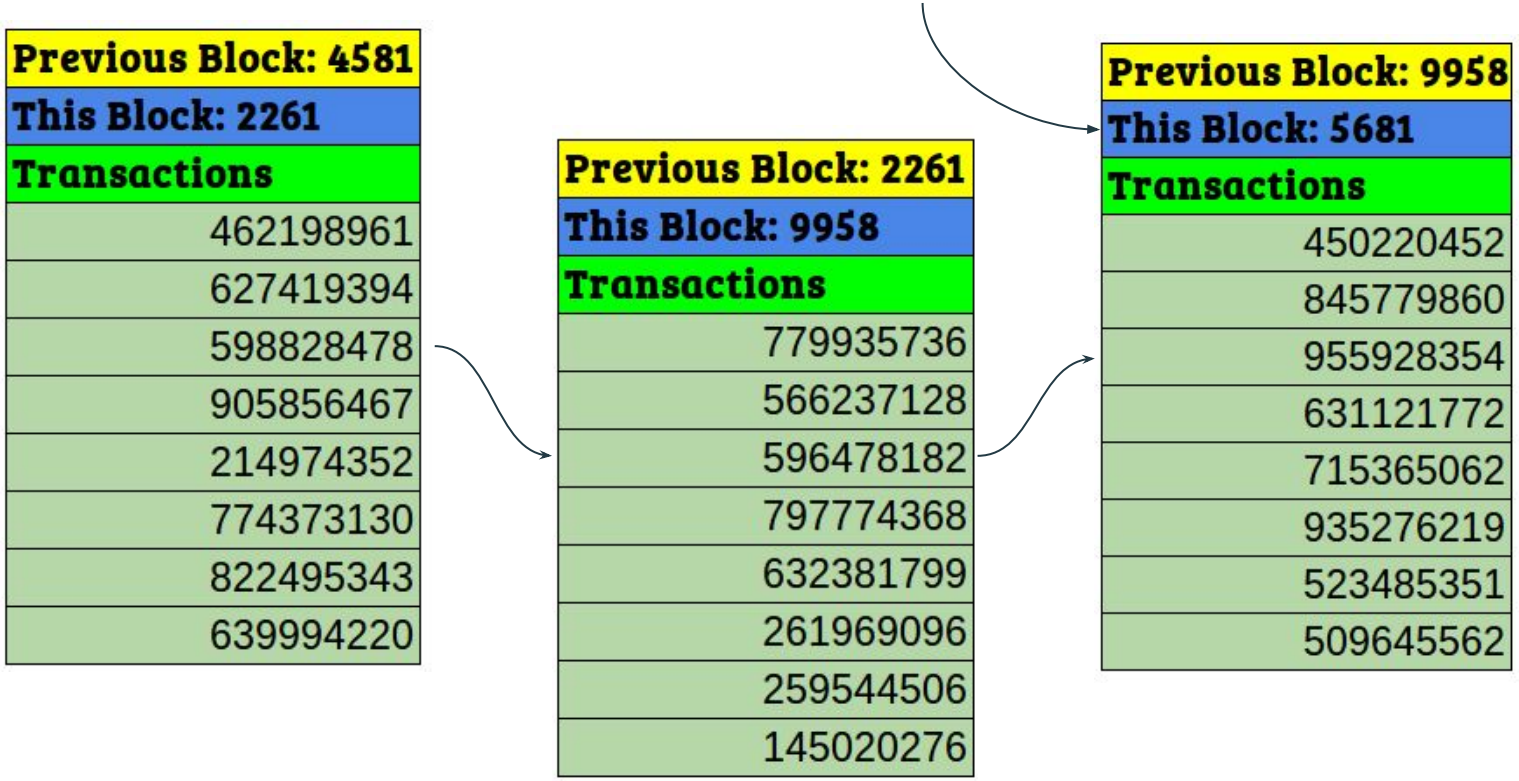
CryptoSoc


Private Key
Public Key
Pay Sally R500
Hash Function
Cryptographic Proof
Verification

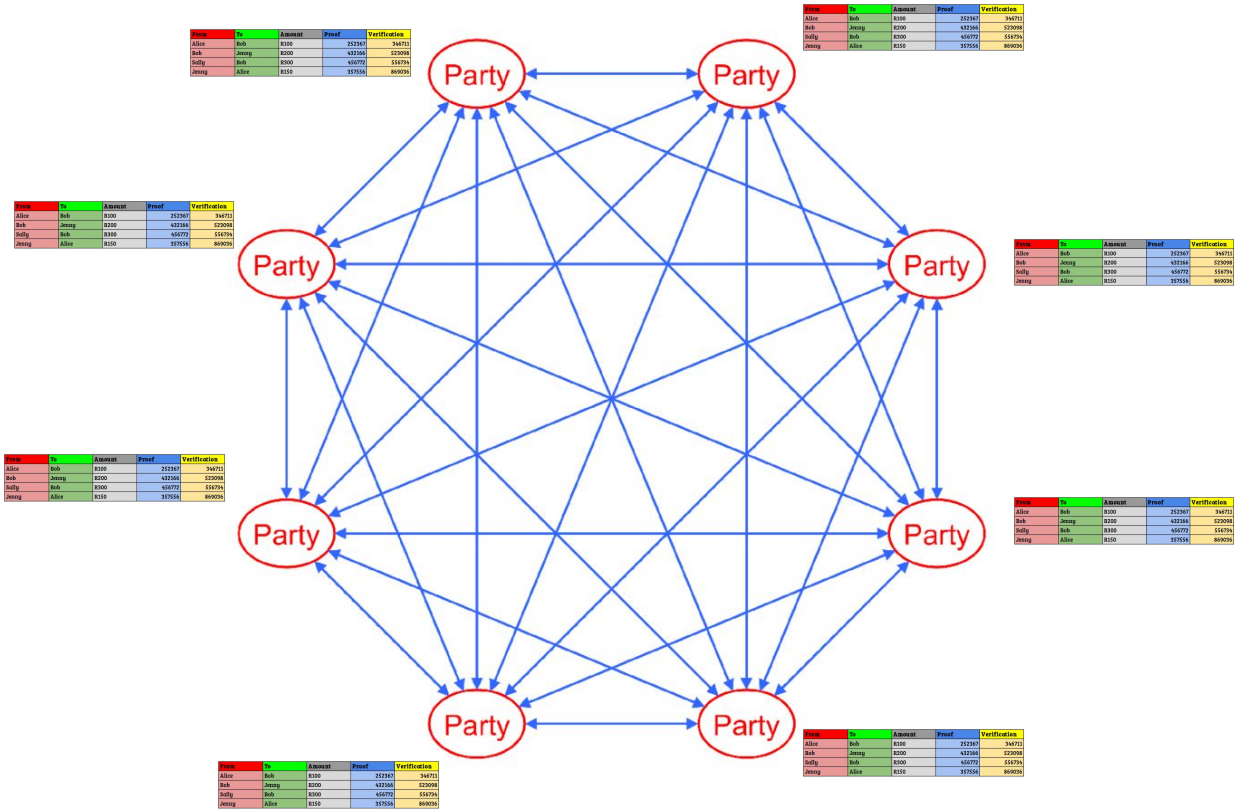
CryptoSoc

Blockchain

Known as the proof of work



CryptoSoc


From To Amount Proof Verification
Alice Bob R100 252367 346713
Bob Jenny R200 432166 523098
Sally Bob R300 456772 556734
Jenny Alice R150 357556 869036
Party
Party
Party
Party
Party
Party
Party
Party

CryptoSoc

Bitcoin

- Record transactions on a ledger that is copied to all participants
- Prove validity through cryptographic hashes
- Accept only the longest “proof of work”

CryptoSoc

Glossary

Cryptographic hash: a function that takes a message and a private key as input and returns a “signature”.

Public key: A public number that everyone can use to verify when something was signed by you.

Private key: A secret number that only you know so that you can verify your identity.

Digital signature: an irreproducible mark specific to a message.

CryptoSoc

Ethereum

- Blockchain is not just for money
- Programmable

CryptoSoc

Some Code

```
class Blockhain():
    def new_block(proof, previous_hash):
        "Adds a block to the blockchain"

    def new_transaction(sender, recipient, amount):
        "Submits a transaction to the network"

    def proof_of_work(last_proof):
        "Generate numbers for proof of work"

    def valid_proof(last_proof, proof):
        "Validate the proof of work"
```

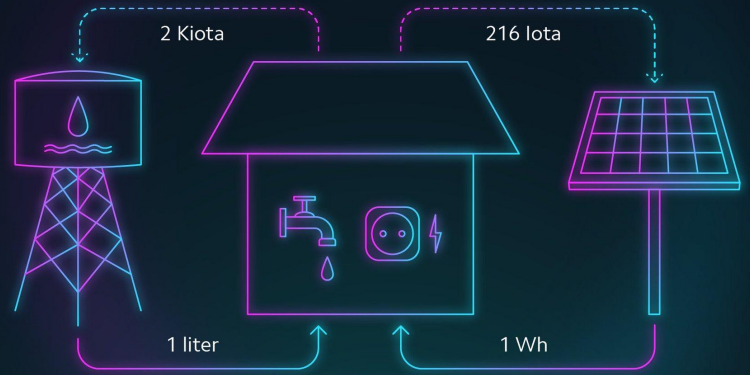
CryptoSoc

Applications

- Remove third parties
- Trustless applications
- Borderless
- Global
- Open

CryptoSoc

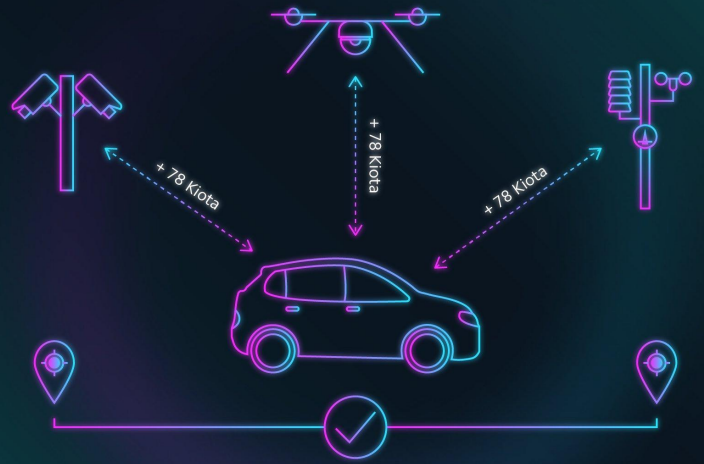
Real World Usecase
Water and Electricity per Unit



With IOTA's feeless transactions water and electricity can be instantly payed for per unit. No more annual subscriptions or middlemen.

IOTA

Real World Usecase
Optimized Routing



A self driving car will be able to collect data about traffic and weather from nearby drones, traffic sensors and weather sensors to optimize its route.

IOTA

CryptoSoc

Applications

- Governance
- Payments
- Asset management
- Insurance
- Networking
- Internet-of-things
- Supply chain
- Identity
- Voting

CryptoSoc

Fin

Slides and links on GitHub later today: https://github.com/cryptosoc

Feedback: https://bit.ly/2NNDiK6

CryptoSoc