Understanding the Bitcoin Protocol. Seriously.

IIC3253

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

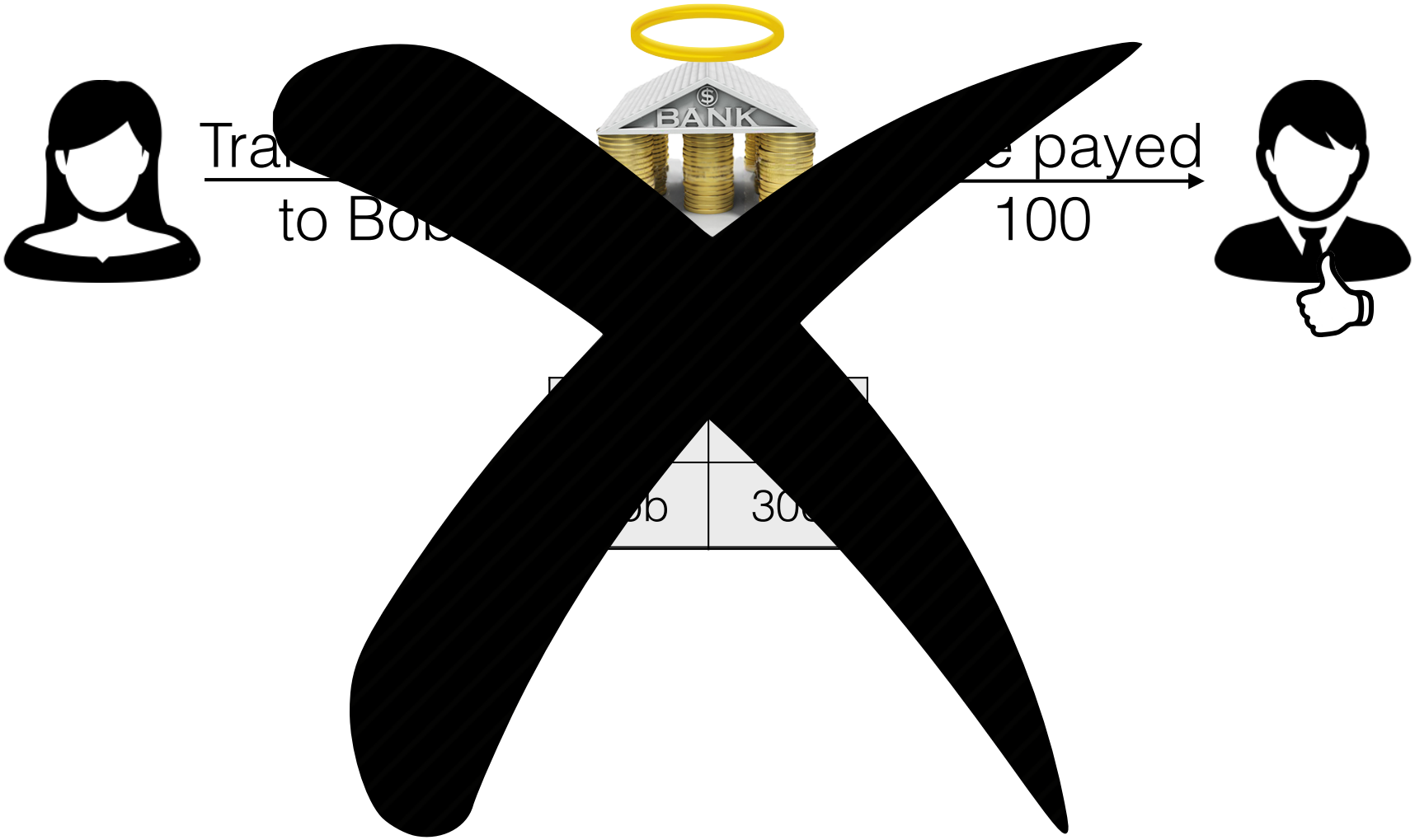
Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

P2P Electronic Cash: Main Challenges



P2P Electronic Cash: Main Challenges

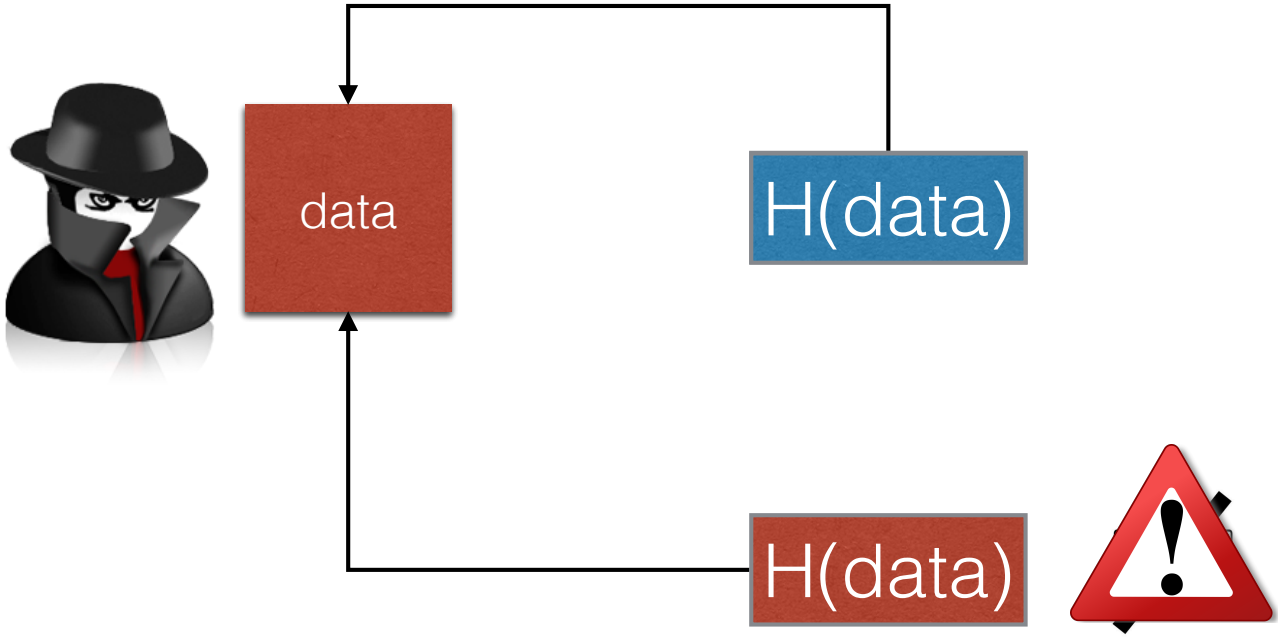
Here Bob, I'm paying you 100

Is this Alice?
Does Alice own 100?
Is she using these 100 elsewhere?

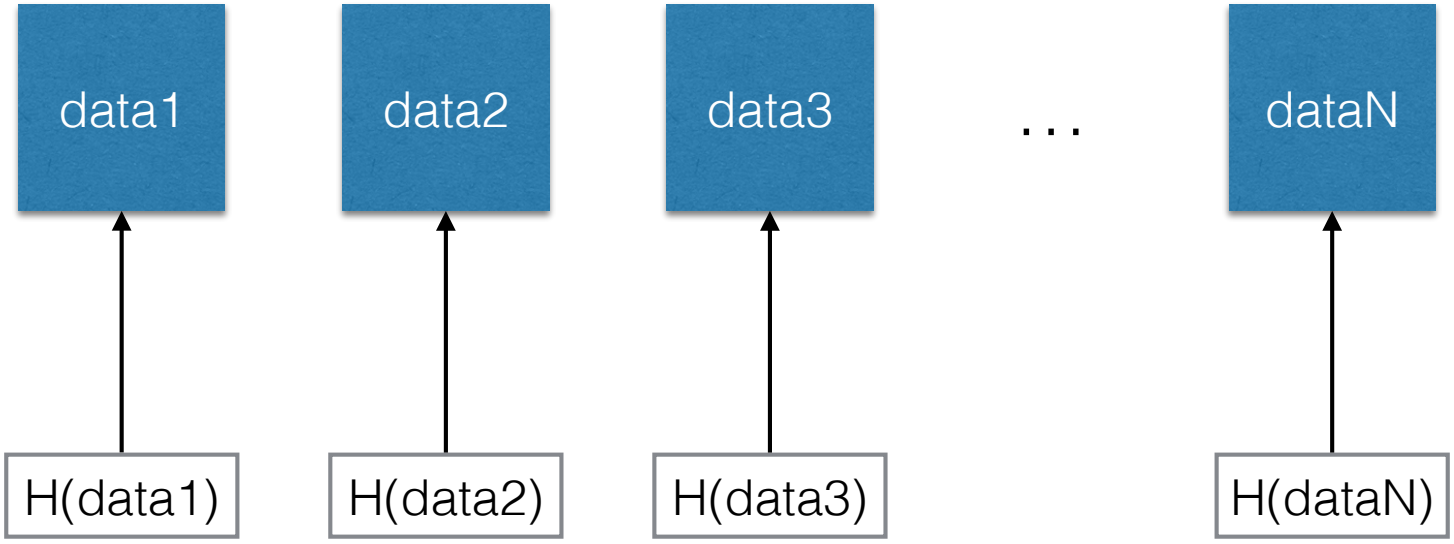
Also: where does currency come from?

Hash Pointers and Integrity

Hash Pointers and Integrity



What if data is added continuously?

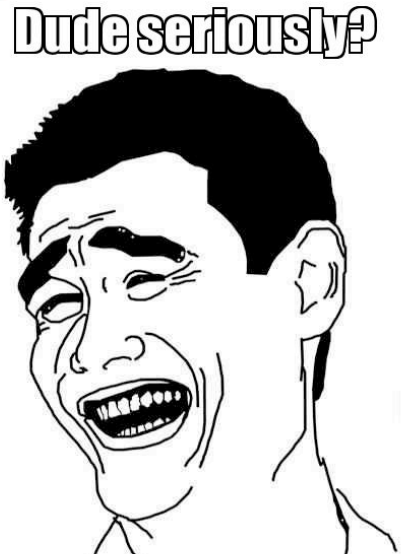


We need to store N hashes to
make sure no data has changed

Can we avoid hashing unbounded portions
of data while storing a single hash value?

Blockchain

WELCOME
TO THE
FUTURE



Our first Cryptocurrency

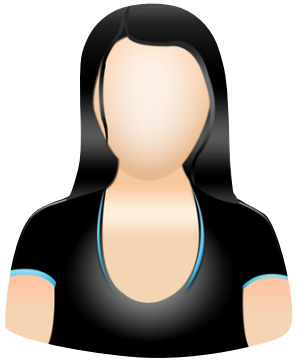
Alice

(PK_A, SK_A)

Bob

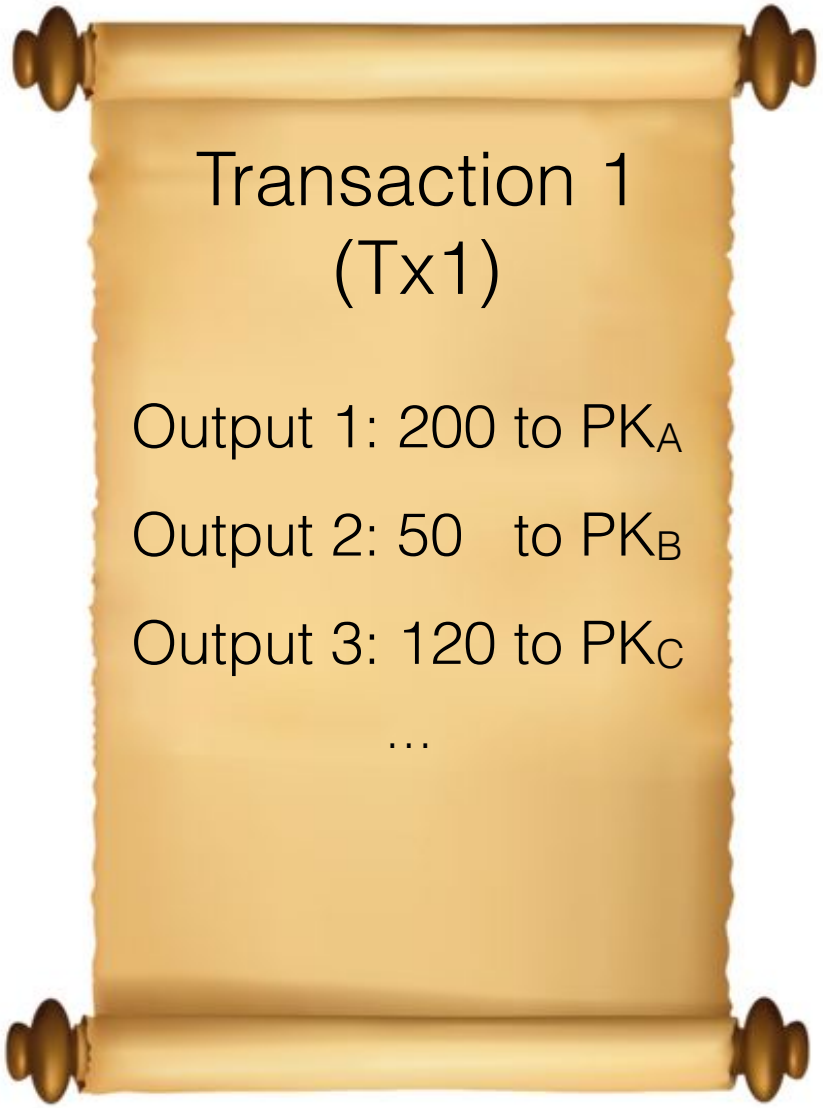
(PK_B, SK_B)

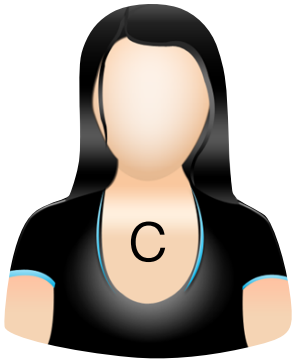
Carol



(PK_C, SK_C)

Our first Cryptocurrency

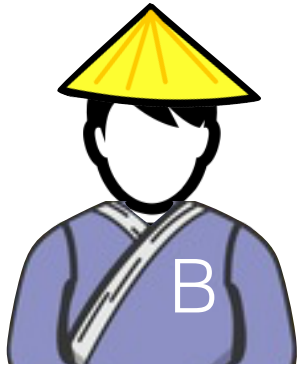




Input 1: H(Tx2), Output1
Output 1: 200 to PK_C

Signed with SK_B

Tx3



Every valid transaction *traces back* to the initial balance sheet

My public key is PK_B

Transaction 1
(Tx1)

~~Output 1: 200 to PK_A~~

Output 2: 50 to PK_B

Output 3: 120 to PK_C

...

*We assume everybody knows this transaction

Input 1: H(Tx1), Output 1
~~Output 1: 200 to PK_B~~

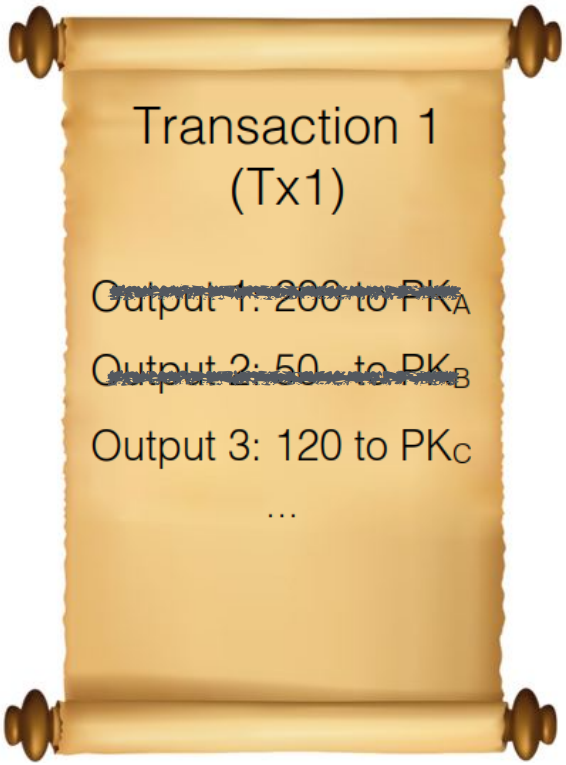
Signed with SK_A

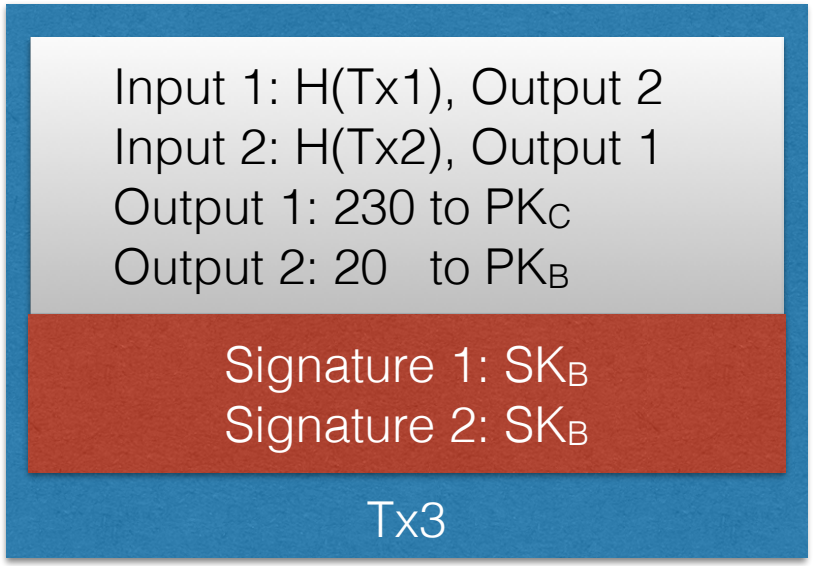
Tx2

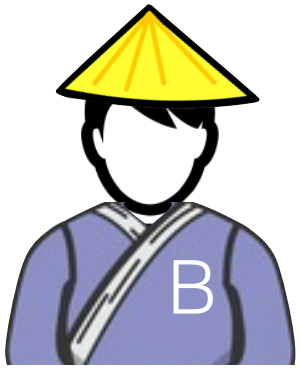
Bob, I want to pay you.
What's your public key?

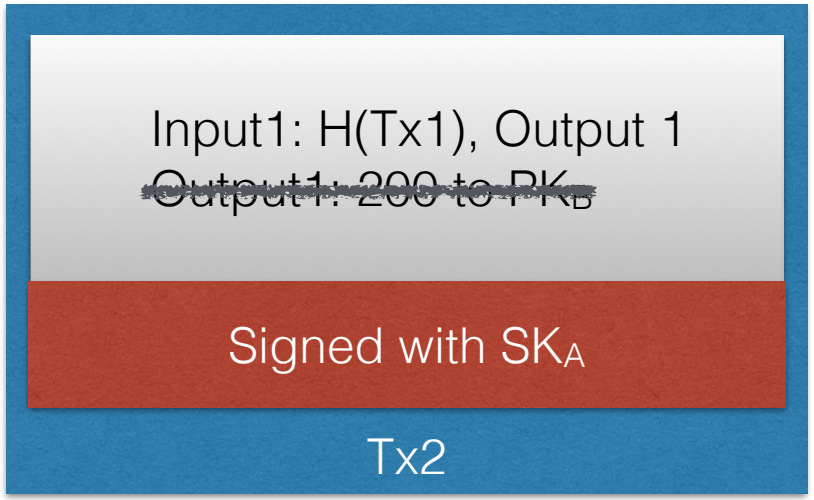
Do I have to spend complete outputs?

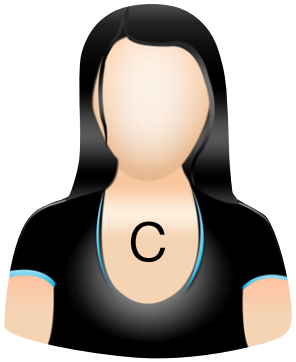
What if I want to pay a fraction of an output?











Input 1: H(Tx3), Output 1
Output 1: 230 to PK_D

Signed with SK_C

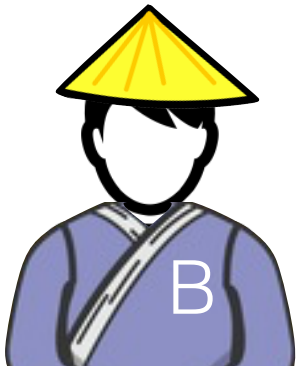
Tx4

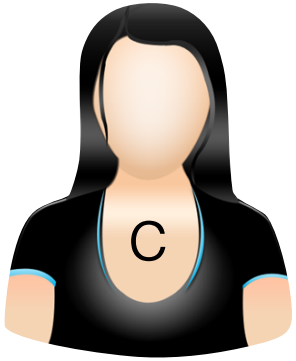
In general, every transaction can have multiple inputs and multiple outputs

Input 1: H(Tx1), Output 2
Input 2: H(Tx2), Output 1
~~Output 1: 230 to PK_C~~
Output 2: 20 to PK_B

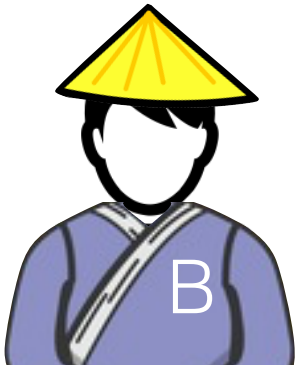
Signature 1: SK_B
Signature 2: SK_B

Tx3





I don't trust this system anymore, I'm out



We need to make sure that money is not spent twice

Input 1: H(Tx1), Output 1
Output 1: 200 to PK_C

Signed with SK_A

Tx2

Input 1: H(Tx1), Output 1
Output 1: 200 to PK_B

Signed with SK_A

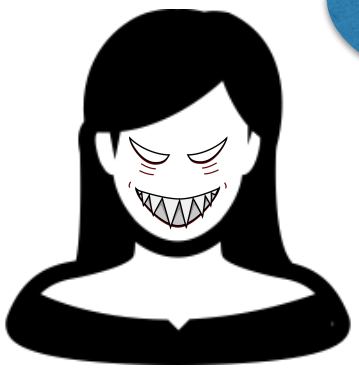
Tx2

~~Output 1: 200 to PK_A~~

Output 2: 50 to PK_B

Output 3: 120 to PK_C

...



Let me help

BankCoin

RULES

- *The initial transaction Tx1 contains the initial balances*
- *A transaction is a sequence of payments that trace back to outputs of Tx1.*
- *Valid transactions are published in a public ledger that is signed periodically by the bank.*

Input: H(Tx1), Output 1
Pay 10 to PK_B, pay 190 to PK_A

Tx2 Signed with PK_A

Input: H(Tx1), Output 3
Pay 20 to PK_B, pay 180 to PK_C

Tx3 Signed with PK_C

Block 1, Signed with SKB

Input: H(Tx2), output 1
Pay 9 to PK_C, pay 1 to PK_B

Tx4 Signed with SK_B

. . .

H(Block 1)

Block 2, Signed with SKB

Transaction 1 (Tx1)

Output 1: 200 to A

Output 2: 50 to Bob

Output 3: 120 to Carol

…

Cool, now I have 71 BankCoins!

B

Don't worry, you can check I never include double spends

And we are all very happy

There's no new block since last week!

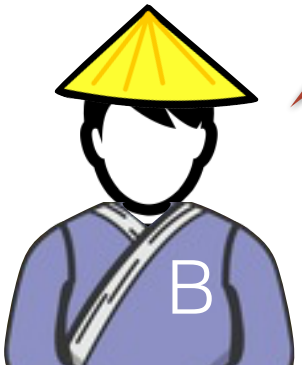
Hey people, bad news, someone stole my private key

Sorry guys, I had a technical problem

Just a week?! my transactions haven't been confirmed in over a month!

It's because I don't like you :)

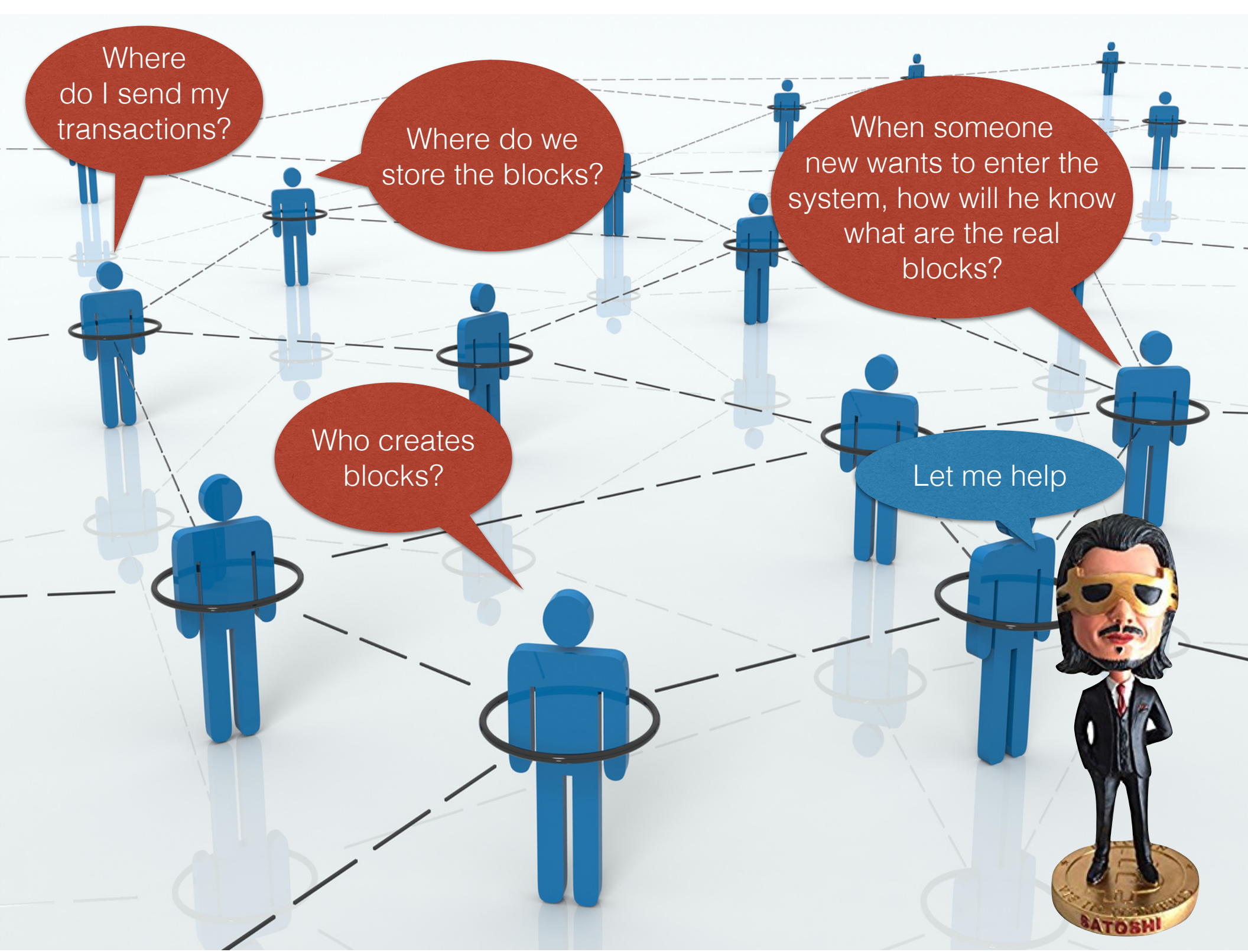
But you can pay me 5 Bcoins and I'll add your transactions



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network.

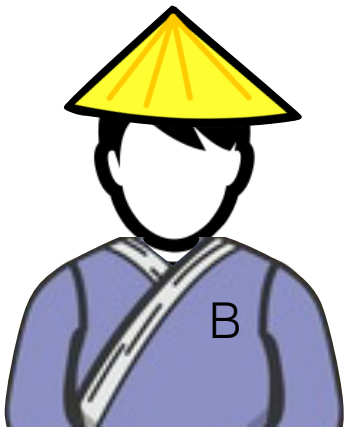

Where do I send my transactions?
Where do we store the blocks?
When someone new wants to enter the system, how will he know what are the real blocks?
Who creates blocks?
Let me help
SATOSHI

RULES

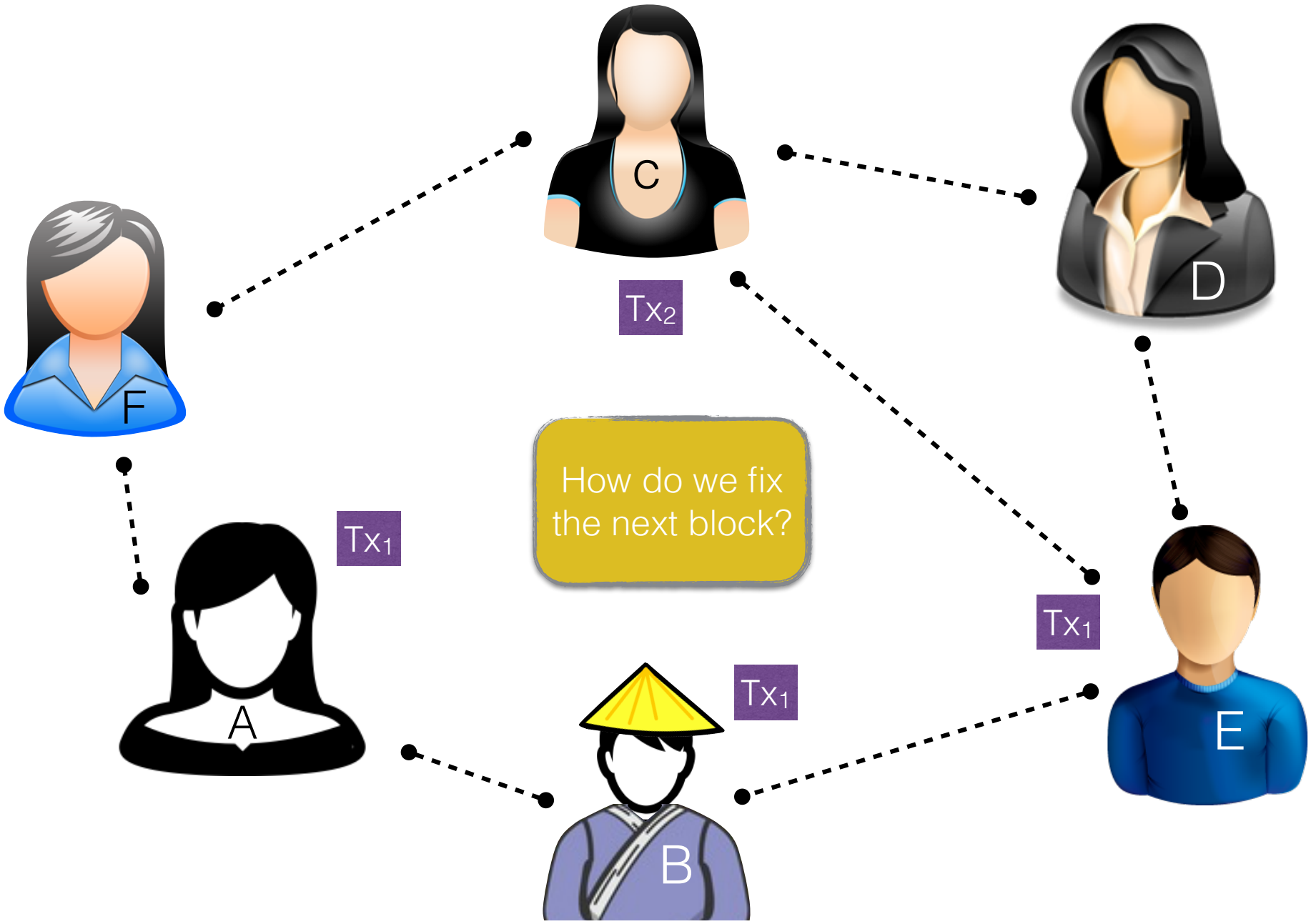
- *Everybody should store all blocks, no trust!*
- *you want to pay, just tell your neighbors*
- *If you receive a new transaction, broadcast it to your neighbors*
- *…*

More on that later…

How do I know when a transaction is valid? How are blocks generated?

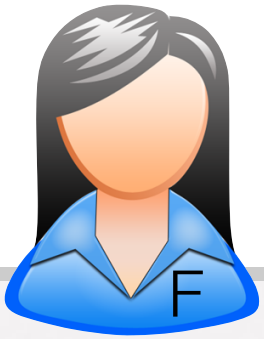


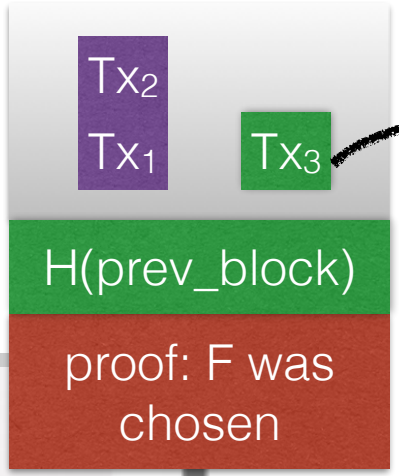
Let us assume for the moment that there is an initial transaction, like with BankCoin.


C
D
F
Tx2
How do we fix the next block?
Tx1
A
Tx1
Tx1
B
E

Assume that a participant is randomly chosen to generate the next block

Why should I do this correctly?



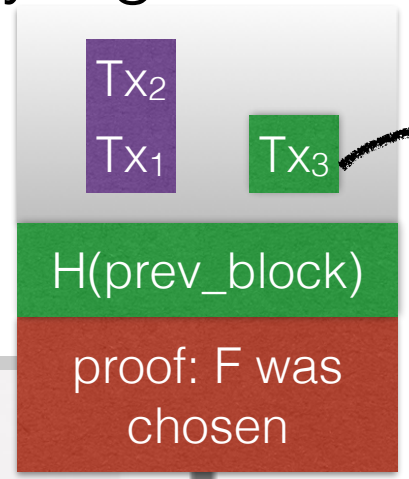


Coinbase transaction (~Fixed)

- …
- *When you form a block, you must broadcast it*
- *If you receive a block, you must check that it is correct and broadcast it*

Mhh.. If only there was a currency that we could use as incentive

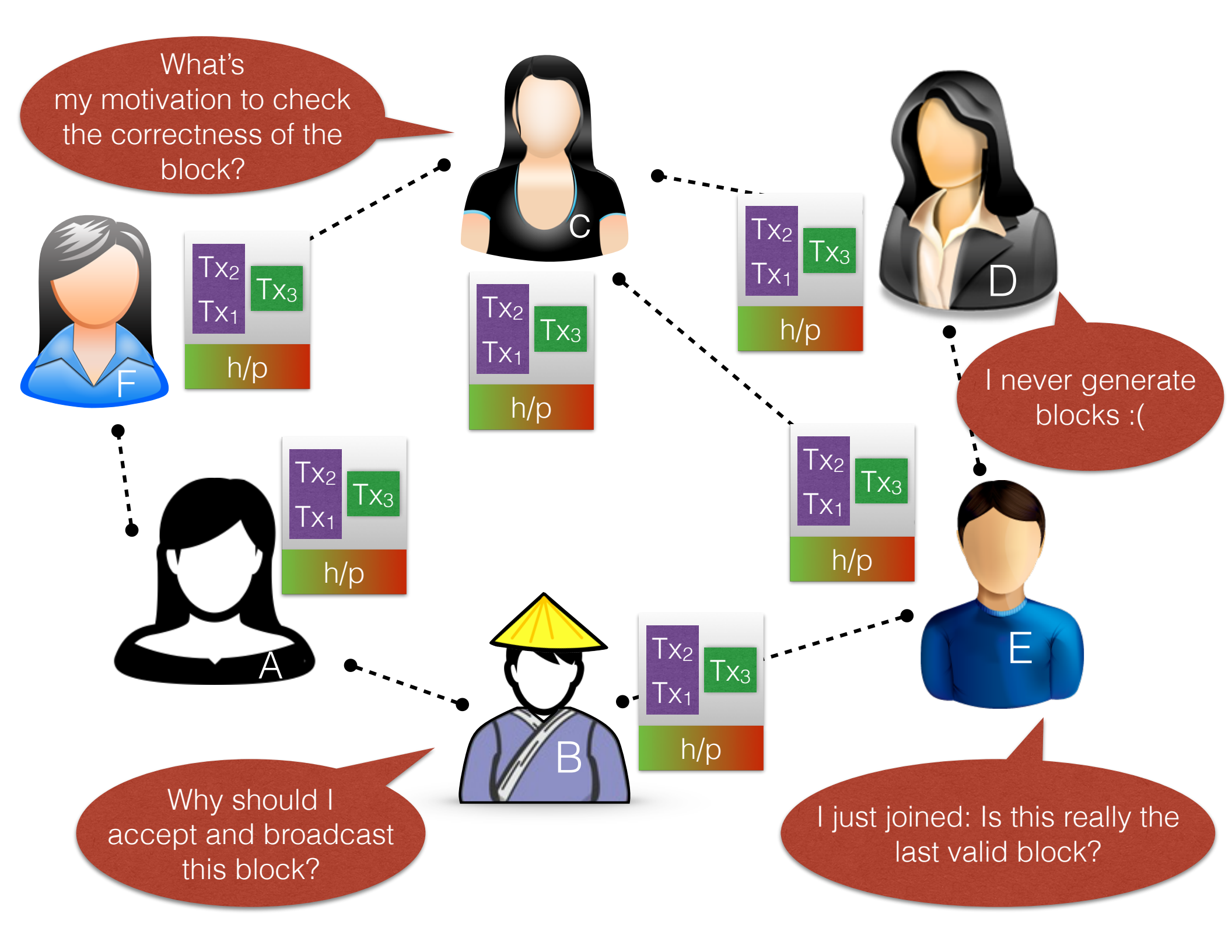
- The proof is correct
- All transactions are correctly signed
- The amounts are correct
- No double spends occur



Coinbase transaction (~Fixed)

- …
- *When you form a block, you must broadcast it*
- *If you receive a block, you must check that it is correct and broadcast it*

Mhh.. If only there was a currency that we could use as incentive


What's my motivation to check the correctness of the block?
C
D
F
A
B
E
Tx2 Tx1
Tx3
h/p
I never generate blocks :(
Why should I accept and broadcast this block?
I just joined: Is this really the last valid block?

Generating Blocks

A fair, verifiable source of randomness?

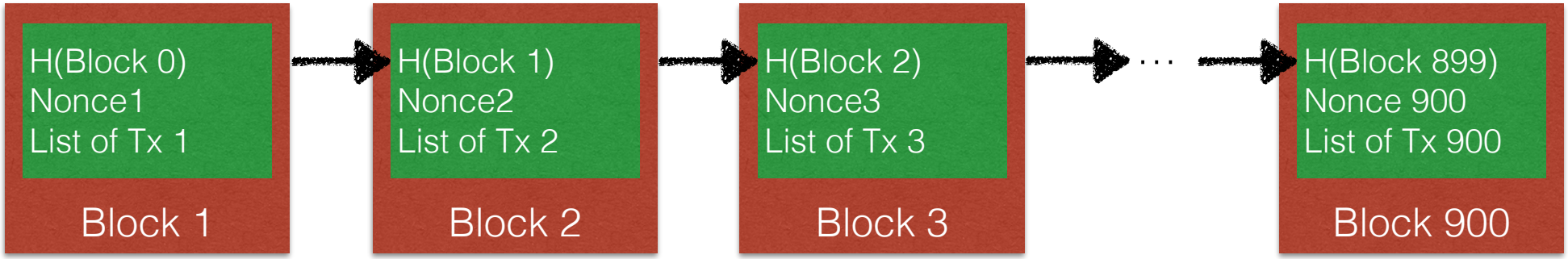
Randomness over what?

Then how do we decide who creates the next block?

Let’s get practical…

How do Blocks really look like?

The Bitcoin Blockchain



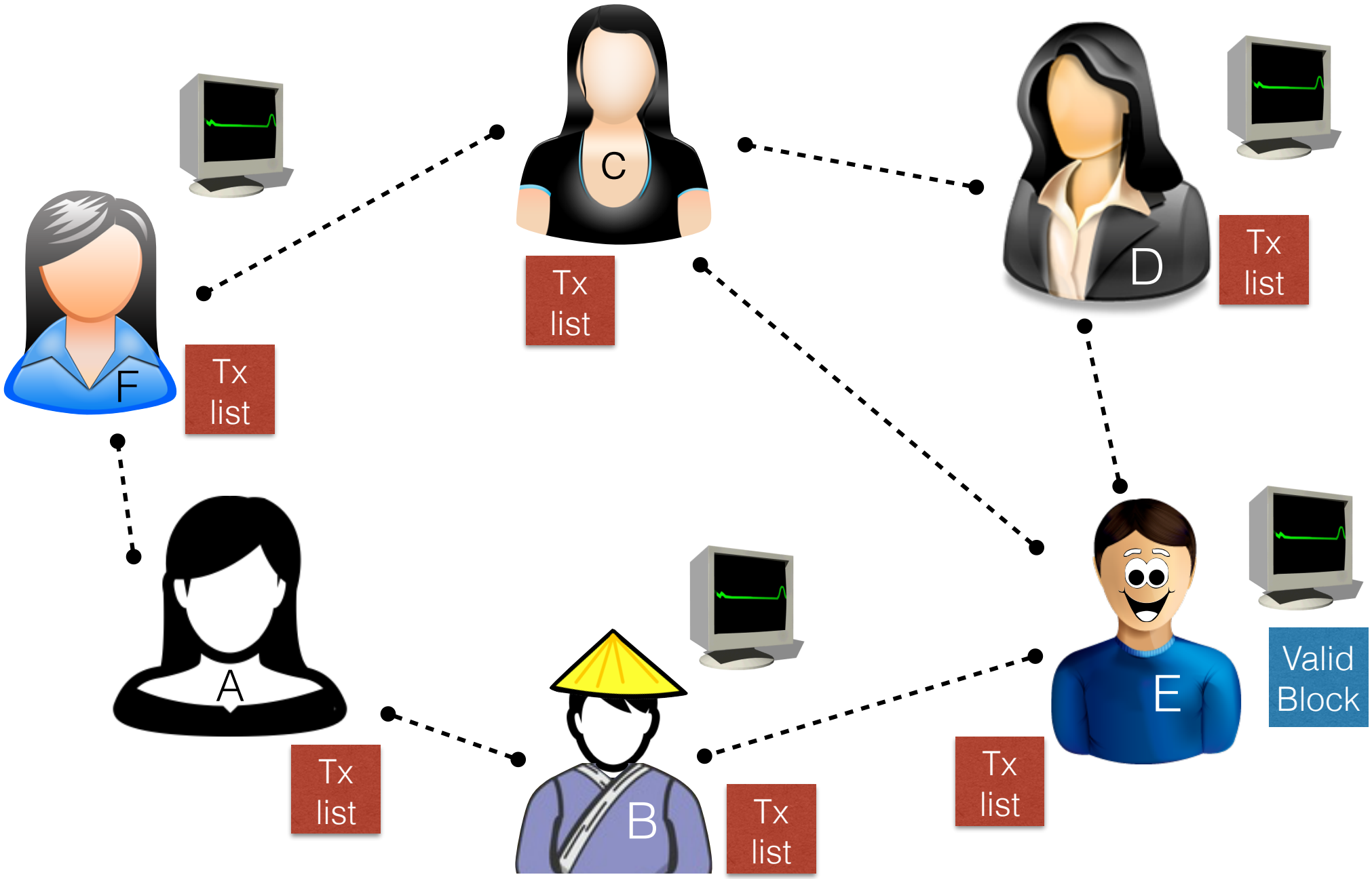
I'm still not chosen, this is not fair!

Not so complicated, huh?

How participants are chosen

The nonce is the key!

You can create a block if you can hash it and get a string starting with, let's say, fifteen zeros



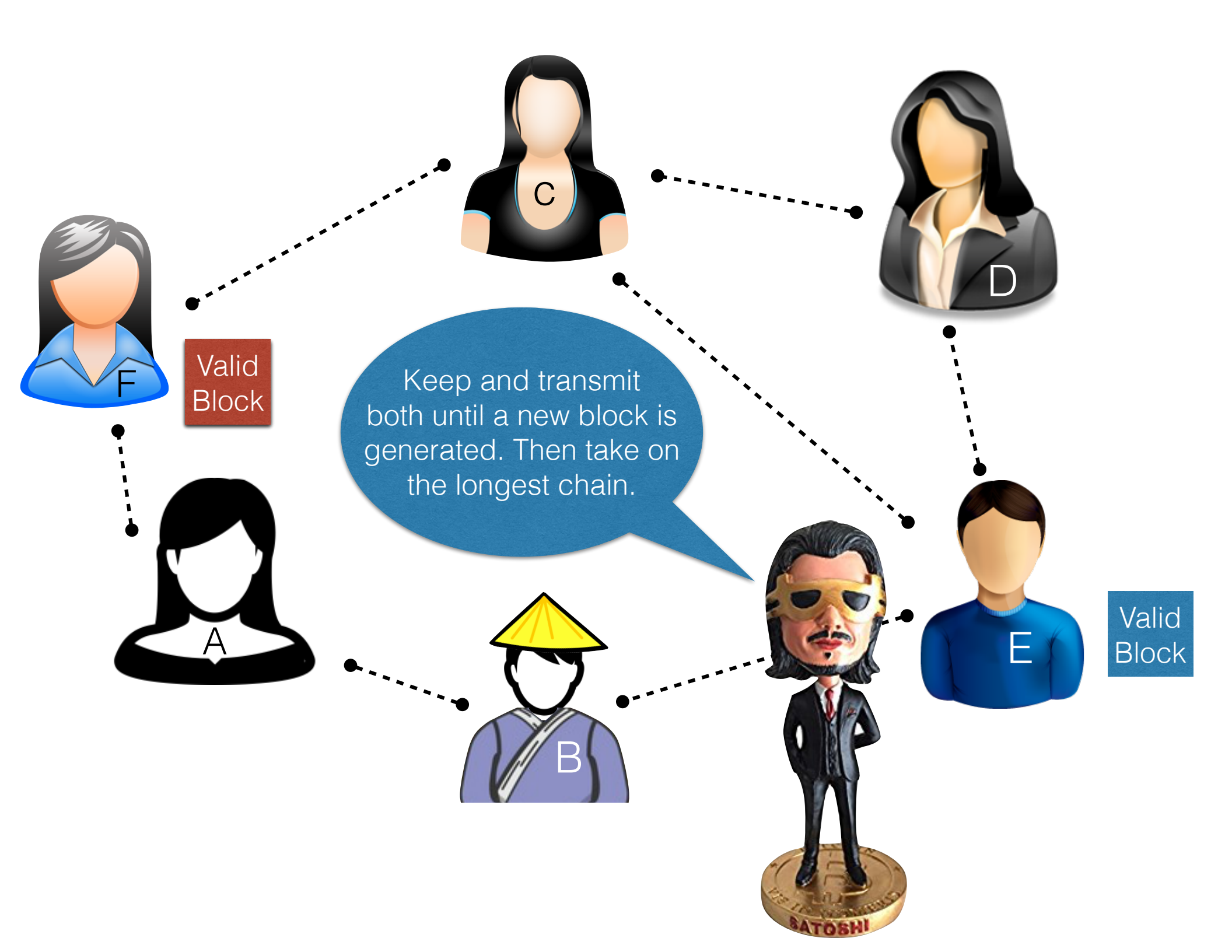
Bitcoin Mining

Bitcoin blocks are not actually signed

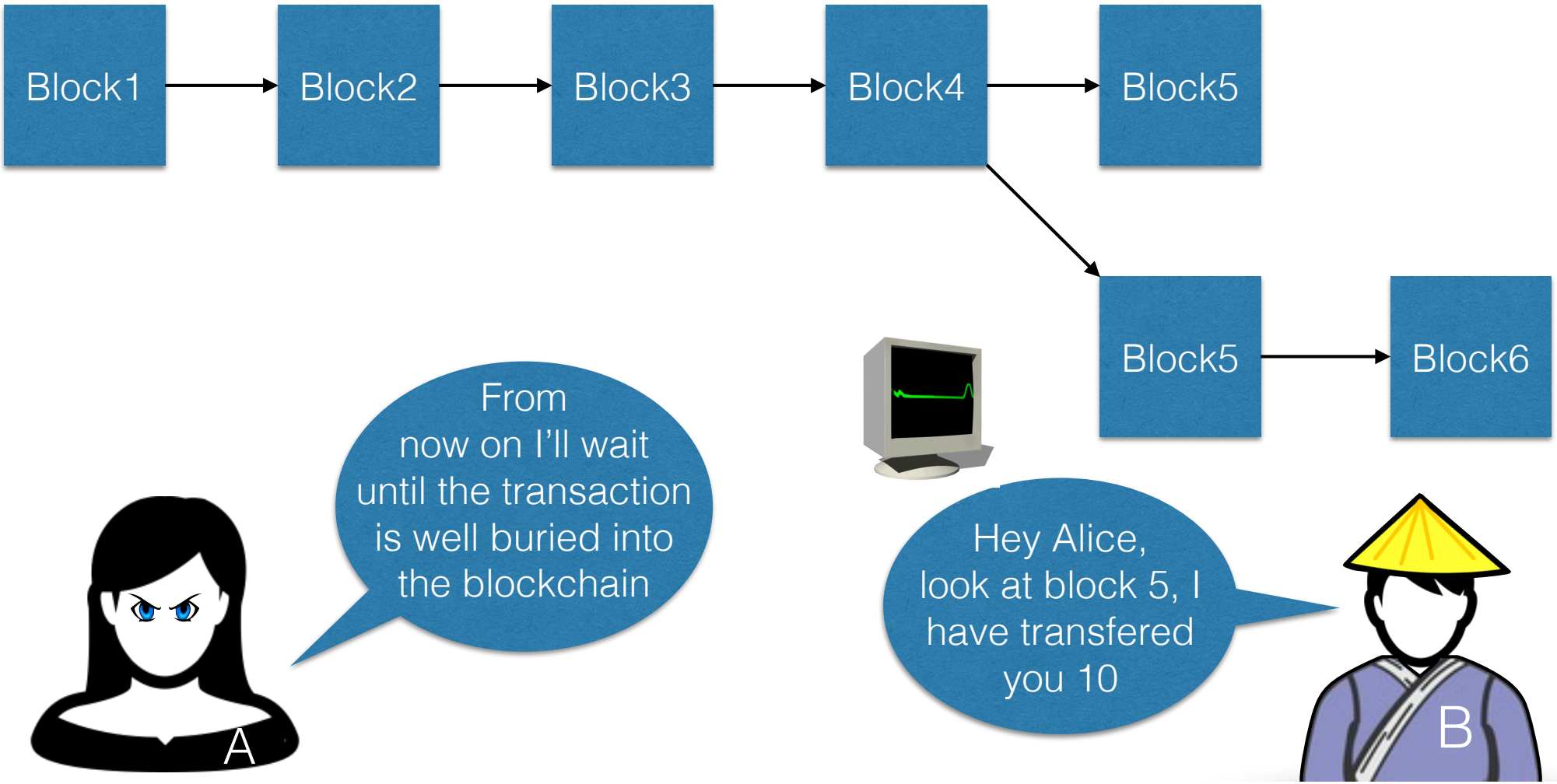
Valid blocks are simply generated by finding a nonce that makes the block “valid”

Having a coinbase transaction somehow *proves* that the recipient *worked* to find it

Proof of Work (PoW)


C
D
F
Valid Block
Keep and transmit both until a new block is generated. Then take on the longest chain.
A
B
E
Valid Block
SATOSHI

How to cheat?


Block1
Block2
Block3
Block4
Block5
Block5
Block6
From now on I'll wait until the transaction is well buried into the blockchain
A
Hey Alice, look at block 5, I have transfered you 10
B

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

The Bitcoin Blockchain

The Bitcoin Protocol

Mining

Questions

Done.