Bitcoin: decentralized electronic cash system

PhD Student - Giacomo Scornavacca

December 20, 2016

Table of Contents

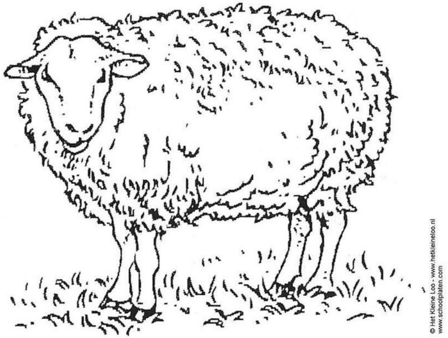
Money before Electronic Currencies

Money is a **medium of exchange** and it has value because people gives value to his properties:

- good divisibility
- good transportability
- hard duplicability
- hard traceability

Goods (e.g. a sheep):

- **bad** divisibility
- **bad** transportability
- **(???)** duplicability
- **(???)** traceability.



Precious metals (e.g. Gold):

- **possible** (but hard) divisibility
- **bad** transportability
- duplicability: only miners can exact new gold and increase inflation. But gold is rare.
- hard traceability.

Currency (1) (e.g. Gold Pound Coins).

In the history the first currency was minted by a Nation using some precious metal (e.g. Gold).

- **good** divisibility (rest)
- **good (?)** transportability
- duplicability: **theoretically** (legally) only the Nation can mint new moneys if provided of enough gold. But gold is rare.
- hard traceability.

Note that the value of the gold in the coin is a lower bound of the value of the coin.

Currency (2) (e.g. the Dollar until 1971).

The currency can be minted by a Bank using some cheap metal (or paper) but everyone in any moment can ask to the Bank to change the moneys in an equivalent amount of Gold (Gold Standard).

- **good** divisibility
- **good** transportability. Paper is better than Gold.
- duplicability: **Now the Bank can mint new moneys also if not provided of enough gold**.
- traceability: Banks have started to put numbers on the banknotes.

Note that the Gold Standard can be view as a lower bound to the value of the currency.

Currency (3) (e.g. all the Currency now).
The Currency was minted by a Bank using some cheap metal (or paper).

- **good** divisibility
- **good** transportability.
- duplicability: **the Bank can mint new moneys when it wants**.
- traceability: Banks continue to put numbers on the banknotes.

No precious metal, no gold standard, only trust in the Bank.

Electronic Currency with central authority (e.g. Paypal, Unicredit . . .) .
Currency is a simple number in a database.

- **Digital** divisibility
- **Digital** transportability.
- (?) duplicability.
- (?) traceability.

Let's implement a electronic currency to understand duplicability and traceability properties.

Preliminaries: RSA

Let $(\mathrm{PUK}, \mathrm{PRK})$ be a private/public key pair in the RSA algorithm. Each message m encrypted with the private key PRK can be decrypted using PUK and vice versa. Assuming polynomially bounded agents and "robustness" of RSA only the owner of private key PRK can be the sender of a message that can be opened by the public key PUK.

$$\mathrm{PUK}(\mathrm{PRK}(m)) = \mathrm{PRK}(\mathrm{PUK}(m)) = m$$

Normally only the **hash** of some message is "signed" using the RSA.

Preliminaries: Cryptographic Hash Function

$h : \{0,1\}^n \longrightarrow \{0,1\}^m$ where generally $m << n$

The properties (informally):

- it is deterministic so the same message always results in the same hash
- it is quick to compute the hash value for any given message
- it is infeasible to generate a message from its hash value except by trying all possible messages
- a small change to a message should change the hash value so extensively that the new hash value appears uncorrelated with the old hash value
- it is infeasible to find two different messages with the same hash value

Electronic payments with Central Authority (Bank) a simple implementation

Users ($\forall i \in$ Users, $(\mathrm{PUK}_i, \mathrm{PRK}_i)$)
Let's assume a secure channel between each User and the Bank
Cryptographic Hash Function $h()$.

Payment (operation):
i wants to give to j 100 euro to obtain a sheep.

- User i create a message $m = (\mathrm{PUK}_i, \mathrm{PUK}_j, 100)$ and send to the Bank $(m, \mathrm{PRK}_i(h(m)))$.
- The bank **checks** the validity of the signature and **checks** if i have 100 on his account.
- The bank notifies to j the output of the payment.

Now j knows if can deliver the sheep.

Properties of Electronic Currency

Properties of Electronic Currency with Central Authority:

- **Perfect** divisibility
- **Digital** transportability.
- duplicability: Only the Bank can create money.
- traceability: Only the Bank knows about payments.

Desired Properties of Electronic Money without Central Authority:

- **Perfect** divisibility
- **Digital** transportability.
- duplicability: No one can create money "for free".
- traceability: The payment system is anonymous.

A Peer-to-Peer Electronic Cash System - Satoshi Nakamoto, 2008

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.

Peer-to-peer Network

General model for peer-to-peer Networks:

- Number of users (nodes) are variable (they can enter and exit from the network).
- When a new user joins the network he asks to a DNS server a set of peers (edges to other nodes). Normally this peers are chosen uniformly at random.
- Users join and leave the network according some update rule.

Often peer-to-peer networks are modelled as Markovian Evolving Graphs . . .

Electronic Cash System: operations

Every user in the network need to perform the following operations:

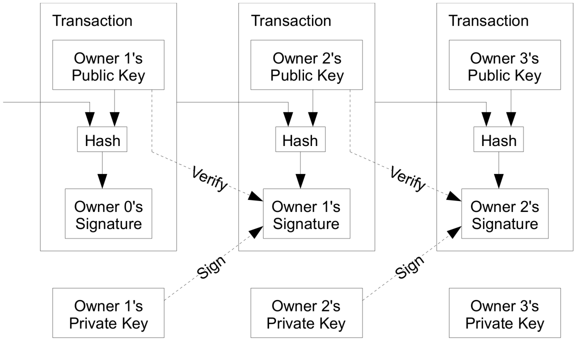
- Pay() # Coin transaction
- Check() # Check if a transaction has been confirmed

Required properties:

- Persistence (a confirmed transaction can not be undone)
- Liveness (a transaction eventually becomes confirmed)

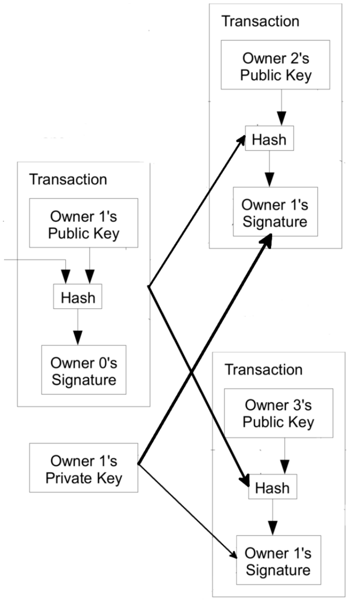
Transactions (single coin model)

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.



The Double-Spending problem

In a naive implementation of electronic coin as chain of digital signature **the payee can't verify that one of the owners did not double-spend the coin**. The following two transactions are both valid, but **they are not valid together**.
Often double-spends are intentional (we call these **double-spend-attacks**): In a transaction, an attacker pretends to transfer an output to a victim, only to double-spend the same output in another transaction back to itself.



The Double Spending solution

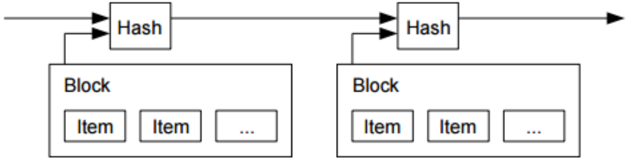
- We need a way for the payee to know that the previous owners did not sign any earlier transactions.
- The only way to confirm the absence of a transaction is to be aware of all transactions.
- To accomplish this without a trusted party, transactions must be publicly announced.
- The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

One-IP-address-one-vote is weak

The majority of nodes agreed transaction was the first received. A implementation of this solution, using classical Consensus protocol for Bizantine agents, is vulnerable to **Sybil attacks**: a malicious user can create multiple accounts in order to have the majority of the nodes in the network and control the majority of the votes.

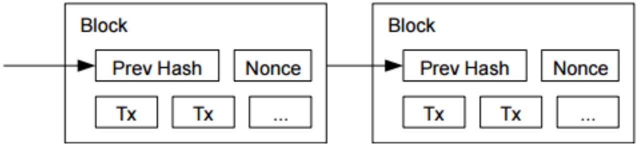
Timestamp Server

The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash [...] The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.



One-CPU-one-vote (Proof of Work)

The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, **the hash begins with a number of zero bits**. [...] Once the CPU effort has been expended to make it satisfy the proof-of-work, the **block cannot be changed without redoing the work**. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



The detailed Protocol

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Forks in the Block Chain

Nodes always consider the longest chain to be the correct one (consensus rule) and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first (**Fork**). In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

Electronic Cash System: operations (2)

Every user in the network need to perform the following operations:

- Pay(): create a transaction and broadcast it.
- Check(): look into the block chain for a transaction.
- Mine(): try put a set of transactions in the block chain.

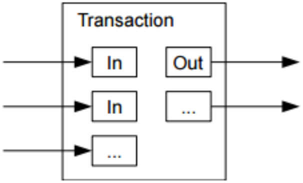
Incentive **and** Coin Mint

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. In this way we solved two problems:

- From a Game Theory prospective we ensure that users are encouraged to finds a proof-of-work.
- We decided how bitcoins are created.

Transactions (full model)

To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.

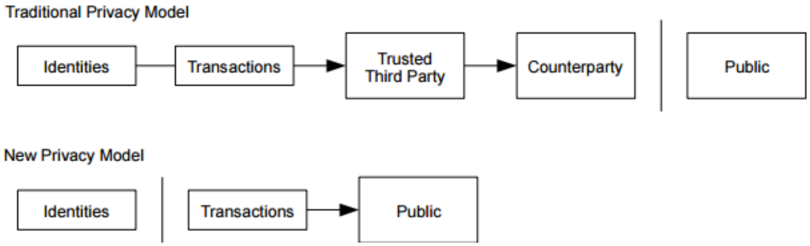


Incentive (2)

If the output value of a transaction is less than its input value, the difference is a **transaction fee** that is added to the incentive value of the block containing the transaction. **Once a predetermined number of coins have entered circulation**, the incentive can transition entirely to transaction fees and be completely inflation free.

Privacy (Pseudo anonymous system)

The traditional banking model achieves a level of privacy by **limiting access to information** to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone
else, but **without information linking the transaction to anyone**.



Questions:

- Is it possible trace the origin of some amount of bitcoins?
- What if two miners work together and they share their rewards? Do they gain something?
- "New transactions are broadcast to all nodes". Why is this a threat for the anonymity in a peer-to-peer network?
- What can happen if someone controls the half of the HASH power? How much it cost to have the half of the HASH power? Which is the total market value of bitcoin today?