Bitcoin: A Peer-to-Peer Electronic Cash System

Presenter: Hang Tong
Overview

- Motivation of Bitcoin
- Technology of Bitcoin: Blockchain
- Usage of Bitcoin
- Mathematical background of Bitcoin security
- How to scale up the system
The motivation of Bitcoin

- "Improve the existing Electronic Coin system"
  - Ecash [David Chaum, 1983]
  - Transaction with 3\textsuperscript{rd} party
    - Rely on 3\textsuperscript{rd} party
    - Subject to financial situation of bank

- Problems
  - 3\textsuperscript{rd} party may bankrupt
  - Vulnerable to the financial crisis (Remember “Lehman shock”)

Existing Electronic Coin System (Trust based model)
The motivation of Bitcoin: eliminate 3rd party

- "Create new currency w/o central organization!"

- How do we create currency?
  - Currency must be trusted by everyone spends it.

- Existing system
  - 3rd party guaranteed the value of currency.
  - How??
How do 3rd parties guarantee the value?

- Trust of source: transaction record.
- They know ALL the transaction history.
  - Prevent mainly 2 things.
    1. Double-Spending Problem:
       To copy and pay same coin for different person.
    2. Coin robbery:
       To change the history of transaction and get someone’s coin.

Existing Electronic Coin System
(Trust based model)
Double-Spending Problem

- They know ALL the transaction history.
  - Prevents mainly 2 things.
    1. Double-Spending Problem
       To spend same coin for different payment.

Your successive transactions have same coin IDs. Payment Invalid.

I wanna do
Me -> A: 5 CHF
Me -> B: 5 CHF
at the same time with same coin, man
They know ALL the transaction history.

Prevents mainly 2 things.

1. Double-Spending Problem
2. Coin robbery:
   To change the history of transaction and get someone’s coin

I wanna change
D -> C: 100 CHF
Into
D -> Me: 100CHF
But I can’t...damn!

A -> B: 5 CHF
D -> C: 100 CHF
A -> C: 7 CHF
...
How do we eliminate the 3rd party?

- They know ALL the transaction history.
  - Prevents mainly 2 things.
    1. Double-Spending Problem
    2. Coin robbery:
       To change the history of transaction and get someone’s coin

- How do we substitute 3rd party?
  -> Blockchain! [Nakamoto, 2009]
Blockchain

- **Block**: A group of transactions
- **Chain**: A sequence of Blocks
- Everyone trusts the longest chain to be true history
Blockchain: in P2P network

- Everyone trusts the longest chain to be true history
  - Free to extend the block
  - Once you get behind, it is really hard to catch up to the top block -> why??
Proof of work

- Meant to be a time-consuming math problem -> require about 10 minutes to solve

- Block is created only after finding the nonce (integer) s.t.
  \[ \text{hash}(\text{nonce} + \text{other data}) = 000\ldots05\text{fs2ce91a} \]

- If someone tries to change Tx1
  -> they have to find another corresponding nonce
  -> same work must be done for subsequent blocks
Blockchain: How does it get added?

Transaction

Unjustified Transactions

Current blocks

Finding a nonce
Block & Transaction creation

Transaction

Unjustified Transactions

Current blocks

New block With tx4

New block With tx4

New block With tx5

Finding a nonce
Block & Transaction creation

Transaction

Unjustified Transactions

Current blocks

New block With tx4

New block With tx4

New block With tx5

"I found!"
Block & Transaction creation

Transaction

Unjustified Transactions

Current blocks

Block candidates

New block
With tx4

New block
With tx5

Block

Block

Block
Block & Transaction creation

Transaction

Unjustified Transactions

Current blocks

Block candidates

"Is nonce correct?" (easy work)

"Let’s check the validity of each transactions"
Block & Transaction creation

Transaction

Unjustified Transactions

Current blocks

Block candidates

New block
With tx4

“I’m sure it’s correct, huh”
“Correct!”
“Correct!”
“Correct!”
“I don’t give a shit.”
Block & Transaction creation

Transaction

Unjustified Transactions

Current blocks

Block candidates

4/5 agreed.

“I’m sure it’s correct, huh”

“Correct!”

“Correct!”

“Correct!”

“I don’t give a shit.”
Block & Transaction creation

Transaction

Unjustified Transactions

Current blocks

New block is added.
New block is added.
The creator can get a bitcoin.
This incident will be written on the new block.
Block & Transaction creation

1.12.2014

First name Surname

Every node should fetch the longest block.
Blockchain security

- Let us recall…
  how does blockchain cope with…
1. Double-Spending Problem ??
2. Coin robbery ??
Blockchain against Double-Spending Problem

1. Double-Spending Problem
   - Each Transaction shows Coin’s hash
   - Blockchain only accepts unique coin hash.
   - If same hash found in 2 transactions, either one will be valid.

In more detail…
- Coin’s hash: hash of previous tx
- Sender ID: Sender’s public key
- Receiver ID: Digital signature from him
2. Coin robbery
   - He might change the transaction arbitrarily.
     -> Actually, he can.
   - However…
     A lot of Proof of Work are waiting for him.

I can modify the transaction like:

Before:
Alice -> Bob: 100 CHF

After:
Alice → Me: 100 CHF
2. Coin robbery
   - He might change the transaction arbitrarily.
     -> Actually, he can.

   - However…
     A lot of Proof of Work are waiting for him.

   - The modification for block makes hash completely different.
     -> He needs to find new nonce accordingly.
     -> 10 minutes work on average.

I can modify the transaction like:

Before:
Alice -> Bob:
100 CHF

After:
Alice -> Me:
100 CHF
Can attacker catch up with the top block?

Attacker is modifying

Honest nodes are creating
Can attacker catch up with the top block?

- Attacker is modifying
- Honest nodes are creating
Can attacker catch up with the top block?

Attacker is modifying

Honest nodes are creating

Block → Block → Block → Block → Block
Can attacker catch up with the top block?

He cannot catch up as long as majority of nodes are honest!
Is it really impossible for attacker to modify the history?

- $p$: $\text{Prob}(\text{Honest nodes can proceed to the next block})$
- $q$: $\text{Prob}(\text{Attacker nodes can proceed to the next block})$
- $q_i$: $\text{Prob}(\text{the attacker eventually can catch up from the position of } i)$
Let us focus on $q_i$

$\rightarrow \text{Gambler’s Ruining Problem (Binomial Random Walk)}$

The gambler starts with money $i$ to reach $N$. If $i=0$, gambler lose. (cannot play anymore)

$p$: $Prob$(Honest majority can proceed to the next block)

$q$: $Prob$(Attacker majority can proceed to the next block)

$q_i$: $Prob$(the attacker eventually can catch up from the position of $i$)

$(q_0 = 0, q_N = 1)$
Mathematical proof

- \( q_i \): recursion equation

\[
q_i = q \times q_{i-1} + p \times q_{i+1}
\]

\[
\vdots
\]

\[
\ldots p \neq \frac{1 - \left(\frac{p}{q}\right)^i}{\frac{q}{q}}
\]

\[
\therefore q_i = \begin{cases}
\frac{1 - \left(\frac{p}{q}\right)^i}{1 - \left(\frac{p}{q}\right)^N} & \text{if } \ldots p = q \end{cases}
\]

\[
\vdots
\]

\[
\frac{i}{N}
\]
Mathematical proof

- Recall:
  \( q_i : Prob( \text{the attacker eventually can catch up from the position of } i ) \)

\[
\begin{align*}
\lim_{i \to \infty} q_i &= \begin{cases} 
1 & \ldots p \leq q \\
\left( \frac{q}{p} \right)^i & \ldots p > q
\end{cases}
\end{align*}
\]

- We assume \( p > q \).
  Someone in majority can create block earlier.

The time to solve proof of work

vs

10min vs 1 day
Summary

- **Block**
  - Nonce, Transactions

- **Blockchain**
  - Sequence of blocks (current length: 513,552 blocks)
  - Length makes tampering difficult
    -> incentive is paid for the creator of new block.
    (amount: 12.5 BTC =~106,416 CHF per block!!)
Disk space problem

- Block size easily gets super huge...
  - # transactions = 304,134,203
  - 1 transaction size = 250 ~ 500 B [https://blockchain.info/]
  - 250B * 304,134,203 = 76TB !!!
  - How do we compress the transaction?
Disk space problem

- Block size easily gets super huge...
  - # transactions = 304,134,203
  - 1 transaction size = 250 ~ 500 B [https://blockchain.info/]
  - 250B * 304,134,203 = 76TB !!!
  - How do we compress the transaction?
    -> Just remove it!
Some transactions: unnecessary!
- Only if the same coins are spent by another latest transactions.
- Merkle Tree “transforms” sequential Tx into Root Hash
- If the coin in Tx0, 1, 2 is spent in another newly created block…
Some transactions: unnecessary!
- Only if the same coins are spent by another latest transactions.

Merkle Tree “transforms” sequential Tx into Root Hash

If the coin in Tx0, 1, 2 is spent in another newly created block...
- Tx0, 1, 2 gets removed.
  Hashcode still remains
Conclusions

- **Bitcoin: History of Bitcoin transaction**
  - Block: Multiple transactions + previous hash + nonce
  - Source of Trust: Majority agreement with Proof of Work

- **Tampering of history:**
  It is difficult if the attacker is a few blocks behind. It is easy to change just one blocks behind. (z = 1)

- **Vulnerability to mining pool**
  These days mining is so competitive that they form some groups (pool).
  What happens if they occupy half of the p2p voting?
The life of Blockchain

1. Transaction is made and cast to pool, broadcast

2. Each nodes collects arbitrary transactions to create a block

3. Whoever has created firstly broadcasts about it

4. All the other nodes check the block to calculate hash (block + found_nonce) and check it’s not already spend

5. Block will be added to the existing chain.
Let $N = i + Z$
(Attacker is Z block behind)

Recall:
$q_i: \text{Prob}(\text{the attacker eventually can catch up from the position of } i)$

$$\lim_{i \to \infty} q_i = \begin{cases} 
\frac{1 - \left(\frac{p}{q}\right)^i}{1 - \left(\frac{p}{q}\right)^i + Z} \to 1 & \text{... } p < q \\
\frac{1 - \left(\frac{p}{q}\right)^i}{1 - \left(\frac{p}{q}\right)^i + Z} \to \left(\frac{q}{p}\right)Z & \text{... } p > q \\
\frac{i}{i + Z} \to 1 & \text{... } p = q
\end{cases}$$
Mathematical proof

- $q_i$ can be calculated by the sum of neighbors.

- $q_i = q \cdot q_{i-1} + p \cdot q_{i+1}$
  
  $$(p + q)q_i = q \cdot q_{i-1} + p \cdot q_{i+1}$$

- $q_{i+1} - q_i = \frac{p}{q} (q_i - q_{i-1}) = \left(\frac{p}{q}\right)^i q_1$ ($\therefore q_0 = 0$)

\[
q_i = \begin{cases} 
1 - \left(\frac{p}{q}\right)^i & \ldots p \neq q \\
1 - \left(\frac{p}{q}\right)^N & \ldots p = q \\
i \over N & 
\end{cases}
\]
Motivation: Reversible? Non-reversible?

- Reversible transactions = transactions with Third-party
  - Reversing: High costs!
  - -> limit the fluidity of transactions
  - (They might accept transactions from 5CHF… and so on)

- Non-reversible transactions = what bitcoin tries to achieve
  - We cannot cancel the payment
  - Instead, the money should be absolutely sent to payee.
  - How do we pay without third-party?
Transaction: who guarantees the value?

- Bitcoin: the history of transaction
  - Source of trust:
    - Shared history with everyone in P2P.

- Cf. Money: the specific note
  - Source of trust:
    - Government or central bank (whoever guarantees its value)

- How should the history be maintained securely?
  - Of course not in .csv, .xlsx like the figure…
  - -> too simple, too easy to modify
Transaction: How should it be maintained?

- Bitcoin: the history of transaction

Focus:
- Who owned Bitcoin?
- Who payed Bitcoin to whom?

Key component
- Hash
- Digital Signature

Satoshi Nakamoto [2009]
Transaction: How should it be maintained?

- Digital Signature recap

- Sender:
  signature
  \[= \text{encrypt}\_\text{with}\_\text{prvkey}(\text{hash}(\text{document}))\]

- Receiver:
  \[\text{hash}' = \text{decrypt}\_\text{with}\_\text{pubkey}(\text{signature})\]
  \[\text{hash}'' = \text{hash}(\text{document})\]
  If \text{hash}' \neq \text{hash}'':
  there was some unintended change.

https://commons.wikimedia.org/wiki/File:Digital_signature_schema.png
Transaction: How should it be maintained?

- **Bitcoin**: the history of transaction
  - Must not be changed.

- **Signature** is crucial
  - To clarify who had the coin

- **Example**:
  - Owner1(payer) pays to Owner2(payee).
  - Hash = hashfunc(previous transaction + payee’s public key);
  - Signature = encrypt_with_prikey(Hash);

- Signature =~ payment

Satoshi Nakamoto [2009]
**Transaction: How should it be maintained?**

- **Signature/pros**
  - Successfully proves owner’s identity
  - If owner X tries to change the history:
    - change owner2 into owner X
    - owner2’s pub key != owner X’s pub key
    - the hash will get changed

- **Signature/cons**
  - Double-spending problem
    - What happens if owner1 has paid same coin to others before?
    - We need the way to verify the previous transaction.
  - How to prove the rightness?

---

Satoshi Nakamoto [2009]
Overview

- The motivation of Bitcoin
- The usage of Bitcoin
  - Transaction
  - Chain of transaction (Proof of Work)
  - The life of blockchain
- The way of scaling up
- The mathematical proof of resistance to tampering
Overview

- The motivation of Bitcoin
- The usage of Bitcoin
  - Transaction
  - Chain of transaction (Proof of Work)
  - The life of block chain
- The way of scaling up
- The mathematical proof of resistance to tampering
Block creation = Bitcoin creation

- Chain of block is the crucial part of bitcoin.
  - To make it difficult to change the history.

- Giving bitcoin is a strong incentive to create blocks.
  - Current reward: 12.50 BTC (≈ 106,416 CHF!!)
  - Also for discouraging the attacker to change the history.

---- separate ---

- Attacker with super strong CPU power has 2 options:
  - Change the history as s/he wishes to get unlimited money.

- Honestly using CPU to generate new coins
Chain of block is the crucial part of bitcoin.
- To make it difficult to change the history.

Giving bitcoin is a strong incentive to create blocks.
- Current reward: 12.50 BTC (=~ 106,416 CHF!!)
- Also for encouraging attacker not to change the history.

Attacker with super strong CPU power has 2 options:
- Change the history as s/he wishes to get unlimited money.
  -> The coin will be useless.
- Honestly using CPU to generate new coins -> profitable option
Space analysis: challenge

- **Statistics** (at 11 March 2018)
  - # blocks = 513,243
  - # transactions = 304,134,203
  - sizeof(1 transactions) = 250 ~ 500 B

  -> sizeof(blockchain) >= 304,134,203 * 250 = 76TB ???

- **In practice**
  - sizeof(blockchain) = 160,571 MB

  Why so different??
Is it really impossible for attacker to modify the history?
Mathematical proof

- Is it really impossible for attacker to modify the history?

Block → Block → Block → Block

Attacker is modifying

Honest nodes are creating
Mathematical proof

- Is it really impossible for attacker to modify the history?

![Diagram of blockchain with attacker modifying and honest nodes creating blocks]
Mathematical proof

- Is it really impossible for attacker to modify the history?

```
Block → Block → Block → Block

Honest nodes are creating

Block

Attacker has created!
```
Mathematical proof

- Is it really impossible for attacker to modify the history?

Block → Block → Block → Block

Honest nodes are creating

Block

Attacker has created!
Mathematical proof

- Is it really impossible for attacker to modify the history?

- It is virtually impossible. Let’s do the math!
Contact information and credits

ETH Zurich
Organisational unit
Street address
Postcode City

www.ethz.ch/en.html

Publisher: Organisational unit of ETH Zurich
Design: Designer Name
Images: Credit (slide xx), Credit (slide xx)

© ETH Zurich, December 2013