Bitcoin

Orfeas Stefanos Thyfronitis Litos

Special thanks to Dionysis Zindros

University of Edinburgh – Informatics School 28/11/2017
Lecture aims

- What is bitcoin
- Addresses, keys
- Transactions, change
- Bitcoin graph, edges, nodes, balances, owners, utxo, coinbase
- Wallets
- Mining, consensus, blockchain, genesis
- Proof-of-work, difficulty, confirmations, miner payment, fees
- Bitcoin value
Bitcoin

- Digital currency
- Allows sending money online
Bitcoin advantages

- Instantaneous money transfer (< 1 sec)
  - Instead of 1 - 2 days for local bank transactions
  - or 20 days for international bank transactions
- Fast transaction confirmation (10 min)
- Security through cryptographic and mathematical properties
  - Instead of security against forgery through chemical/physical properties
How can I get bitcoin?

- In the same way you get pounds!
- You can **work** and get paid in bitcoin
- You can **sell** products and services for bitcoin
- You can **exchange** pounds for bitcoin
  - With exchange services
    - https://cex.io
  - With somebody else in person
    - https://localbitcoins.com
    - A friend with bitcoin that is willing to exchange
Bitcoin network

- All nodes connected to common p2p network
- Every node runs a bitcoin implementation (bitcoind, bcoin, etc.)
- Most implementations are open source
- Anyone can freely join the network
- Nodes do not have to trust the network! Everybody assumes that neighbours may lie
Keys

- Bitcoin uses an elliptic curve (secp256k1) for its public key cryptography
- A user can create a keypair (P, x)
  - P: public key
  - x: private key
- An **address** can be generated from the public key
- We **receive** coins with the address
- We **spend** coins with the private key
Keys example

Private key:
5JXesisRRU2Z7HMmwMpNtoiYk1QDMVjV3HLoYMd1PTKEkJhJT1z

Public key:
045a5f526dfe5d5995bf95f1229e70e21818190883c40ab3590458476ad34aaae59bc772b98a587035b452638b59238e2a39e954b43ab7a4f32408664d36ec1575

Address: 133GT5661q8RuSKrrv8q2Pb4RwSpUTQU1Z
Keys example

Private key:

5JXesisRRU2Z7HMmwMpNtoiYk1QDMVjV3HLoYMd1PTKEkJhJT1z

Public key:

045a5f526dfe5d5995bf95f1229e70e21818190883c40ab3590458476ad34aaaae59bc772b98a587035b452638b59238e2a39e954b43ab7a4f32408664d36ec1575

Address: 133GT5661q8RuSKrrv8q2Pb4RwSpUTQU1Z
Keys example

Private key:
5JXesisRRU2Z7HMmwMpNtoiYk1QDMVjV3HLoYMd1PTKEkJhJT1z

Public key:
045a5f526dfe5d5995bf95f1229e70e21818190883c40ab3590458476ad34aaae59bc772b98a587035b452638b59238e2a39e954b43ab7a4f32408664d36ec1575

Address: 133GT5661q8RuSKrrv8q2Pb4RwSpUTQU1Z

Always one
Transactions

- The basic structure in bitcoin is a transaction (tx)
- A transaction transfers money from somebody to somebody else
Alice £15 tx Bob £15
Public Transactions

- All transactions are published!
- Everybody can see all transactions
- Users create a new address whenever they receive bitcoin
- Reusing addresses would be bad for privacy
Alice 36 mBTC tx Bob 36 mBTC
tx

1FdtUtK5vZxwo8jzjzid5EwGAB7paqX4n

36 mBTC

128MZKqUsvg2kYQ5LCVDx8Mdn8xrijzQY

36 mBTC
\[ \text{txid} = \text{SHA256}^2 (\text{tx}) \]

- 36 mBTC from Alice to tx
- 36 mBTC from tx to Bob
The transaction graph

- Payments are done through **linking** transaction nodes
- Money is a **chain of transactions**
Unspent money

- Spendable money are **outgoing unlinked edges**
  - unspent transaction outputs - utxo
How do I ask for money?

- I generate a private key and its corresponding public key and address
- I send the **address** to the **payer**, e.g. through chat
- I watch the **network** for a transaction that pays me
What money do I own?

- The ones that are still in the UTXO, thus are still unspent
  - Otherwise I have transferred ownership to someone else
- On the outgoing edge there is a public key for which I hold the private key
- To calculate my balance, I sum the outgoing values

How do I ask for money?

- I generate a private key and its corresponding public key and address
- I send the address to the payer, e.g. through chat
- I watch the network for a transaction that pays me
How do I spend money?

- I find a transaction that has a \textit{UTXO} of which I am the \textit{owner}
- I create a \textit{new transaction}
- \textbf{With one incoming and one outgoing edge}
- I connect the \textit{incoming edge} of the \textit{new} transaction with the \textit{old UTXO}
- Now the old UTXO is not a UTXO anymore – it was just spent
- The \textit{outgoing edge} of the \textit{new tx} is unconnected – it is the \textit{new UTXO}
- I specify the \textit{value} and the \textit{owner} (address) of the new outgoing edge
tx

1 BTC

Alice
1 BTC

Alice

utxo
1 BTC
Alice

tx ————>  tx
1 BTC
Alice

not utxo anymore

1 BTC
Bob

new utxo
Alice signs

No one else can forge this signature
Transaction broadcasting

- **Broadcast**: When I create a transaction, I send it to all my neighbours.
- **Relay**: When I receive a transaction from a neighbour, I check if it is valid and then send it to the rest of my neighbours.
- In a few moments, the whole network learns about a new transaction.
One transaction – many inputs

- I can spend money from many UTXOs in one transaction
One transaction – many outputs

- I can pay multiple recipients with one transaction

```
2 BTC
Wage
```

```
1 BTC
Rent
```

```
0.5 BTC
Electricity
```

```
0.5 BTC
Internet
```
One transaction – many outputs

- ...or keep change in case of a small transaction
- I give change to myself, not the seller
Kirchhoff’s Law

\[ \forall \ tx \in \txs:\]
\[ \sum_{i \in \text{in}(tx)} w(i) \geq \sum_{o \in \text{out}(tx)} w(o) \]
Kirchhoff’s Law

\[ \forall tx \in \text{txs}: \sum_{i \in \text{in}(tx)} w(i) \geq \sum_{o \in \text{out}(tx)} w(o) \]

- All transactions ever
- Outgoing value
- Incoming value
Transaction validity

- To ensure the validity of a newly received transaction
- I already know some valid transactions with UTXOs
- I verify Kirchhoff’s law
- I verify the digital signature
- I verify that the inputs connect to already known valid UTXOs
  - This verifies that money is spent exactly one time
Wallet

- A bitcoin *private key set*
- Usually an application
- Mobile or Desktop
Desktop wallet - Electrum

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
<th>Amount</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-09-30</td>
<td>03:57</td>
<td>+2000.0</td>
<td>2257.59289</td>
</tr>
<tr>
<td>2013-09-02</td>
<td>15:48</td>
<td>+257.59289</td>
<td>257.59289</td>
</tr>
<tr>
<td>2013-07-17</td>
<td>20:17</td>
<td>-1000.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2013-07-17</td>
<td>19:51</td>
<td>+1000.0</td>
<td>1000.0</td>
</tr>
<tr>
<td>2013-07-05</td>
<td>12:51</td>
<td>-240.75443</td>
<td>0.0</td>
</tr>
<tr>
<td>2013-07-05</td>
<td>12:43</td>
<td>-142.98671</td>
<td>240.75443</td>
</tr>
<tr>
<td>2013-06-28</td>
<td>20:40</td>
<td>+240.75443</td>
<td>383.74114</td>
</tr>
<tr>
<td>2013-05-31</td>
<td>13:17</td>
<td>+110.94938</td>
<td>142.98671</td>
</tr>
<tr>
<td>2013-05-27</td>
<td>11:58</td>
<td>+32.03733</td>
<td>32.03733</td>
</tr>
<tr>
<td>2013-05-09</td>
<td>01:33</td>
<td>-100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2013-04-21</td>
<td>20:08</td>
<td>+100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>2012-11-09</td>
<td>21:13</td>
<td>-2328.65664</td>
<td>0.0</td>
</tr>
<tr>
<td>2012-10-26</td>
<td>04:49</td>
<td>+2051.2</td>
<td>2328.65664</td>
</tr>
<tr>
<td>2012-10-10</td>
<td>23:14</td>
<td>-220.0</td>
<td>277.45664</td>
</tr>
<tr>
<td>2012-10-10</td>
<td>20:32</td>
<td>+220.0</td>
<td>497.45664</td>
</tr>
<tr>
<td>2012-10-10</td>
<td>18:58</td>
<td>-583.53221</td>
<td>277.45664</td>
</tr>
<tr>
<td>2012-10-10</td>
<td>18:49</td>
<td>+500.0</td>
<td>868.98885</td>
</tr>
<tr>
<td>2012-10-10</td>
<td>17:27</td>
<td>-360.0</td>
<td>360.98885</td>
</tr>
<tr>
<td>2012-09-30</td>
<td>15:53</td>
<td>-1545.00547</td>
<td>3428.98885</td>
</tr>
<tr>
<td>2012-09-29</td>
<td>00:07</td>
<td>+1545.00547</td>
<td>4965.99432</td>
</tr>
<tr>
<td>2012-09-25</td>
<td>03:45</td>
<td>+277.45664</td>
<td>3428.98885</td>
</tr>
</tbody>
</table>

Balance: 2.02604 mBTC
Mobile wallet - Android

<table>
<thead>
<tr>
<th>Currency</th>
<th>Rate</th>
<th>Balance</th>
<th>Received</th>
<th>Both</th>
<th>Sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNY</td>
<td>416.78</td>
<td>465.2653</td>
<td>Apr 6 ← 1719Pmoehr5Ck1dX6mQ9zYj4nTPnGdf5- + 0.0050</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DKK</td>
<td>328.56</td>
<td>366.7824</td>
<td>Apr 5 ← Beer with Lisa + 0.0050</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EUR (default)</td>
<td>49.90</td>
<td>55.7050</td>
<td>Apr 5 → 1Q4H8CY4FpnJ93SPbdz4Cqg7v714KXae- - 3.5005</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GBP</td>
<td>40.74</td>
<td>45.4794</td>
<td>Apr 4 ← Donation + 0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HKD</td>
<td>506.94</td>
<td></td>
<td>Apr 3 ← 1FUuQeguKnVFavXYqKwYB7g4YXJ4REKjh + 0.05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Use at your own risk. Read the safety notes.
**Double spending**

- Two transactions that spend the same output are named **double spend**
- Kirchhoff’s law holds for both transactions
- All signatures are valid
Double spending attack

- Eve buys shoes from Bob
- She simultaneously double spends to herself
- She takes the shoes and leaves
- Bob learns about the double spend later
The arrow of time

- We need to put transactions in some order
- We must be able to answer: Did tx A happen before tx B?
- The answer should be common for everyone in the network
- Global agreement on a common truth is named consensus
  - This is where the bitcoin novelty is
Block

- Contains many transactions
- It cannot contain double spends, that is txs that spend the same output
- Each transaction can appear **only once** in a block
Block

- The network is set to create **one block every 10 minutes**
- A newly created block contains the **most recent transactions** that did not exist in previous blocks
- Blocks are **broadcast** and **relayed** in the network, just like transactions
- The SHA256$^2$ of the block is the **block id**
- A transaction in a valid block is called **confirmed**
confirmed transaction
blockid = SHA256(tx, tx, tx, tx, tx, tx)
Blockchain

- Each block refers to its previous block
- It contains a pointer to the blockid of its parent
- A later block cannot contain a double spend of a previous one
- ...or a transaction that appeared in a previous block
- This connected list is called blockchain
Blockchain

- Each block refers to its **previous** block
- It contains a pointer to the blockid of its **parent**
- A later block cannot contain a double spend of a previous one
- ...or a transaction that appeared in a previous block
- This connected list is called **blockchain**
Blockchain

- Achieves **consensus**
- Tx A **precedes** tx B if A is contained in a previous block from B
- If we want to ensure that a transaction will not be double spent, we have to wait for it to be confirmed
Who creates blocks?

- **Anybody** can create a block
- The system is free for everybody
- Each block must contain a **proof of work SHA256**
- The proof of work has such a **difficulty** that the **entire network** create **1 block every 10 minutes on expectation**

\[
E(\text{block generation time}) = 10 \text{ min}
\]
Mining

- The process of block creation is called **mining**
- There are **many miners** who try to mine blocks
- Each miner has a **small probability** to extract a particular block
- When a miner successfully mines a block, she **broadcasts** it
- The rest of the nodes **relay** it
Bitcoin proof of work

\[ \text{SHA256}^2(\text{txs} \ || \ \text{nonce} \ || \ \text{parent-blockid}) < \varepsilon \]
Genesis block

- The **first** block in the blockchain is the genesis block
- It is **hard-coded** in the bitcoin software
- Every valid blockchain begins from the genesis
  - It is the **base** of the **induction** for block validity confirmation
Genesis block

- Contains the text “The Times 03/Jan/2009 Chancellor on brink of second bailout for banks”
- This proves that the block was created on or after 3 January 2009
- We also know that it was made on or before 3 January 2009 because we observed it on the network
- Thus it was made on 3 January 2009
Chancellor on brink of second bailout for banks
Blockchain forks

- Often two valid blocks may be mined simultaneously
- This creates a **blockchain fork**
Blockchain fork

- A blockchain fork is a problem because it prevents us from maintaining the arrow of time
- We have the same problem as ordering transactions
- Which of the two blocks is the most recent valid block?
- What happens if two rival blocks contain double spends?
Algorithm for resolving rival blockchains

- We observe two rival blockchains on the network
- The valid blockchain is the one with the maximum height
- If the two blockchains have the same height, we choose one at random.
- The chosen block is the one on top of which we mine and/or trust for transaction confirmation
Double spending

- To successfully double spend, I have to create a malicious parallel blockchain equal or longer than the honest one.
Double spending difficulty

- Double spending requires **great computational power**
- The malicious actor must control more computational power than the rest of the network
- Otherwise the probability of catching up with the honest blockchain falls **exponentially** with the length of the honest blockchain
- He can achieve it if he controls more than 50% of all the computational power in the world
- This is called a **51%-attack**
What can a malicious miner achieve?

- Can he double spend?
  - ?
- Can he prevent money from being spent?
  - ?
- Can he spend our money?
  - ?
What can a malicious miner achieve?

- Can he double spend?
  - Yes – he makes a parallel blockchain containing the malicious transaction

- Can he prevent money from being spent?
  - Yes – he makes a parallel blockchain that does not contain the undesired transaction

- Can he spend our money?
  - No – he doesn't have our private keys!
Mining incentives

- A miner is rewarded in 2 ways:

1. With all the remaining money from the transactions she confirms:

   \[
   \text{fees} = \sum_{tx \in \text{block}} \left[ \sum_{i \in \text{in}(tx)} w(i) - \sum_{o \in \text{out}(tx)} w(o) \right]
   \]
Mining incentives

- A miner is rewarded in 2 ways:

2. With **one** coinbase transaction she is allowed to put in the block with a value of **12.5 BTC**
Mining incentives

- A miner is rewarded in 2 ways:

2. With **one** coinbase transaction she is allowed to put in the block with a value of 12.5 BTC
Coinbase transaction

- A coinbase transaction is the only one that can have an incoming edge with no beginning
- It is the induction base for transactions validity confirmation
- Exactly one coinbase transaction is allowed in each block
- The coinbase value must be 12.5 BTC
- This is the only way to create bitcoin
Bitcoin value

- Extreme variance
- 23/11/2017: 1 BTC = £6,130
- 2016: 1 BTC = £590
- Max 2013: 1 BTC = £750
- Min 2013: 1 BTC = £45
- 2012: 1 BTC = £6
- 2010: 1 BTC = £0.05
- 22/5/2010: First purchase with bitcoin
22 May 2010: One pizza for 10,000 BTC
What we learned

- What is bitcoin
- Addresses, keys
- Transactions, change
- Bitcoin graph, edges, nodes, values, owners, utxo, coinbase
- Wallets
- Mining, consensus, blockchain, genesis
- Proof-of-work, difficulty, confirmations, rewards, fees
- Bitcoin value