CPSC 426/526
Blockchain & BitCoin

Ennan Zhai
Computer Science Department
Yale University
Final Exam

• Former Three questions:
  - Lec 13, 16, 17, 18, and 21
  - Similar to the midterm

• The last question:
  - Select one from the five alternative questions
  - I.e., Lec 11, 14, 15, 19, and 20
  - If you answer more than one, I will use the highest one as your credit for the last question
Lecture Roadmap

- Blockchain Basics
- Case Study: BitCoin
- Case Study: CoinShuffle
- Summary of our course
The Blockchain

BitCoin $\neq$ Blockchain
The Blockchain

A: $10
B: $2
C: $5
D: $7
The Blockchain

A: $10
B: $2
D: $7
C: $5
The Blockchain

A: $10  
B: $2  
D: $7  
C: $5

A -> B: 3

A: 10 - 3 = 7
B: 2 + 3 = 5
The Blockchain

A: $7

D: $7

A → B: 3

A: 10 - 3 = 7

B: 2 + 3 = 5

B: $5

C: $5

A → B: 3

A: 10 - 3 = 7

B: 2 + 3 = 5
The Blockchain

A: $7
B: $5

A -> B: 3
A: 10 - 3 = 7
B: 2 + 3 = 5

D: $7
C: $5
The Blockchain

- Blockchain is used to decentralize the log:

  A: $7
  B: $5
  C: $5
  D: $7

  A -> B: 3
  A: 10 - 3 = 7
  B: 2 + 3 = 5
The Blockchain

- Blockchain is used to decentralize the log:
  - Decentralization
  - Public accountability
  - Efficiency
The Blockchain

- Blockchain is used to decentralize the log:
  - Decentralization
  - Public accountability
  - Efficiency
The Blockchain

- Each block contains multiple transactions
- Each user locally maintains a ledger
- All ledgers should have the same data
The Blockchain

Log (or Ledger)

• Each hash identifies the entire prefix of the log
The Blockchain

A: $7  B: $5

A -> B: 4

D: $7  C: $5
The Blockchain

A: $3

B: $9

D: $7

C: $5
The Blockchain

A: $3
B: $9
C: $5
D: $7

A -> B : 4
The Blockchain

A: $3

B: $9

D: $7

C: $5

A→B : 4

B→D : 1
The Blockchain

A: $3

B: $8

C: $5

D: $8

A -> B : 4
B -> D : 1
The Blockchain

A: $3
B: $8
C: $5
D: $8

I am the leader

A -> B : 4
B -> D : 1
The Blockchain

A: $3
B: $8
C: $5
D: $8

New Block

A -> B : 4
B -> D : 1
The Blockchain

A: $3
B: $8
C: $5
D: $8

A -> B: 4
B -> D: 1

New Block
The Blockchain

• Blockchain can be used to decentralize any centralized service:
  - Making them decentralized (without single-point-fault)
  - Public accountability
The Blockchain

• Blockchain can be used to decentralize any centralized service:
  - Making them decentralized (without single-point-fault)
  - Public accountability

• We still have two problems:
  - How to achieve consensus?
  - How to preserve the privacy?
Lecture Roadmap

• Blockchain Basics
• Case Study: BitCoin
• Case Study: CoinShuffle
• Summary of our course
Deployment of BitCoin Nodes

- Blockchain is used to decentralized bank:
  - Each user has several wallets (public keys)
  - They sign the money transaction using the private key
How to compute BitCoin?

If B’s initial value is 0, then B is 4-1+1-2+1=3
How to compute BitCoin?

<table>
<thead>
<tr>
<th>Previous output (index)</th>
<th>Amount</th>
<th>From address</th>
<th>Type</th>
<th>ScriptSig</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x38f77560ca...1</td>
<td>8</td>
<td>1P9SgqujFWgWVAnZBFwimNPV7LmaaJpgTj</td>
<td>Address</td>
<td>304502200078d1c47ec4e152b4d04eac4a73afe731476606392a06584e1a4f300332ef3c4bb1</td>
</tr>
<tr>
<td>0x912295f5a8...1</td>
<td>0.03</td>
<td>58Mv65wV1E5kCVHFShwUTU6zytyVFKM5F</td>
<td>Address</td>
<td>3045022004e877f5ca3783e165052e44788dd4769b655cb4d127b4e024c8624b4c4f2d7e71</td>
</tr>
<tr>
<td>0x5837d9f8e8...15</td>
<td>1</td>
<td>1G4f0i2wAq1ECoDawq5tvUTB2PwLr2</td>
<td>Address</td>
<td>3044022075d236d4a500846677721065f16e9646dd4f5b7e3f33f1535458eabf3b7f22d2b4a7</td>
</tr>
<tr>
<td>0x69d1cd1c2ac...1</td>
<td>150</td>
<td>1LPbL6jN0SmGq7h6BGZzwbo6bX29h9YWzC7</td>
<td>Address</td>
<td>3046022100a65a1a8889a4e5e2a9a5ba38750304ab81a15938e0d7e67685694791b5245699</td>
</tr>
<tr>
<td>0x7674a521c...1</td>
<td>0.55357267</td>
<td>16Kn6XppHUbjmgY8QpMyv9JXIN9A5Xvb</td>
<td>Address</td>
<td>3045022100ec761d1a62e38d6462e16f1d114f0f195d26f5705038871a3b18b663f1277f00</td>
</tr>
<tr>
<td>0x544027a0e69...0</td>
<td>0.03270607</td>
<td>1LjsQx1g6c757578AnUemj6YQe6CTw54QN</td>
<td>Address</td>
<td>3045022100859d2ed74793e8658a0423e0163d04257e6490b9d1618886b065ca7b43816faab</td>
</tr>
</tbody>
</table>

**Inputs**

<table>
<thead>
<tr>
<th>Index</th>
<th>Redeemed at input</th>
<th>Amount</th>
<th>To address</th>
<th>Type</th>
<th>ScriptPubKey</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8baaca27d158...</td>
<td>0.0107174</td>
<td>F7BgcQbyWTRWEMUKNzLdjkbsQT9K96m</td>
<td>Address</td>
<td>OP_DUP OP_HASH160 9ab2d82e0c0a63d8a36575a3128f15e89f7274e394 OP_EQUALVERIFY OP_CHECKSIG</td>
</tr>
<tr>
<td>1</td>
<td>1bb973b44c8e...</td>
<td>139.605567</td>
<td>NT2FMa1l1NjCZytd4kgXRZPfi56ZPGZ</td>
<td>Address</td>
<td>OP_DUP OP_HASH160 eb471d7a903535e9c9e1f2f9f20eaadad8479af OP_EQUALVERIFY OP_CHECKSIG</td>
</tr>
</tbody>
</table>

**Outputs**

139.6
How to compute BitCoin?
How to compute BitCoin?

A→B : 4
A→D : 1
E→D : 3

B→C : 1
A→B : 1
C→D : 3

B→D : 2
A→B : 1
A→C : 1

A→B : 4
C→D : 1
Who should generate a new block to include these two transactions?
They need to compete, and the winner can earn money.
How to compute BitCoin?

\[ X = \text{SHA256}(H + \text{salt}) \]

\[ X \text{ should be ‘0000....’} \]
How to compute BitCoin?

X = SHA256(H + salt)
X should be ‘0000....’
SHA256("The quick brown fox jumps over the lazy dog")
0x d7a8fbb307d7809469ca9abcb0082e4f8d5651e46d3cdb762d02d0bf37c9e592
SHA256("The quick brown fox jumps over the lazy dog.")
0x ef537f25c895bfa782526529a9b63d97aa631564d5d789c2b765448c8635fb6c

X = SHA256(H + salt)
X should be ‘0000....’
SHA256("The quick brown fox jumps over the lazy dog")
0x d7a8fbb307d7809469ca9abcb0082e4f8d5651e46d3c3db762d02d0bf37c9e592
SHA256("The quick brown fox jumps over the lazy dog.")
0x ef537f25c895bfa782526529a9b63d97aa631564d5d789c2b765448c8635fb6c

X = SHA256(H + salt)
X should be ‘0000....’
How to compute BitCoin?

A→B : 4
A→D : 1
E→D : 3

B→C : 1
A→B : 1
C→D : 3

B→D : 2
A→B : 1
A→C : 1

Salt=8

A→B : 4
C→D : 1
How to compute BitCoin?

Salt = 8
Proof of Work

• BitCoin uses the proof of work to achieve many goals:
  - Generating additional money
  - Achieving consensus while tolerating malicious users
  - A great incentive mechanism
Proof of Work

• Occasionally, more than one block will be solved at the same time, leading to several possible branches
Proof of Work

• We should build on top of the first one you received.
• Others may have received the blocks in a different order, and will be building on the first block they received
Example

I found a block extending the blockchain

I found a block extending the blockchain
Example
Example

I found a new block that extends the blockchain
Example

Oops! This new block extends ANOTHER chain

I found a new block that extends the blockchain

Great! This new block extends the chain
Proof of Work

• We do not need to worry about the branch problem:
  - You always immediately switch to the longest branch
  - The math makes it rare for blocks to be solved at the same time, and even more rare for this to happen multiple times
  - The end result is the block chain quickly stabilizes
Proof of Work

• We do not need to worry about the branch problem:
  - You always immediately switch to the longest branch
  - The math makes it rare for blocks to be solved at the same time, and even more rare for this to happen multiple times
  - The end result is the block chain quickly stabilizes

• ~10 minutes to generate a new block
• Your transactions are confirmed after 6 blocks
Proof of Work

• We do not need to worry about the branch problem:
  - You always immediately switch to the longest branch

  Miners in BitCoin can earn a lot of money!

• ~10 minutes to generate a new block
• Your transactions are confirmed after 6 blocks
Miner’s life
Recall: The Blockchain

• Blockchain can be used to decentralize any centralized service:
  - Making them decentralized (without single-point-fault)
  - Public accountability

• We still have two problems:
  - How to achieve consensus?
  - How to preserve the privacy?
Lecture Roadmap

• Blockchain Basics
• Case Study: BitCoin
• Case Study: CoinShuffle
• Summary of our course
Linkability of Pseudonyms
Linkability of Pseudonyms

[Meiklejohn et al., IMC'13]
Setup

Alice

Bob

Carol

Dave

$(sk_A, vk_A)$

$(sk_B, vk_B)$

$(sk_C, vk_C)$

$(sk_D, vk_D)$
Step1: Announcement

A' ← AddrGen();

(ek_B, dk_B) ← EncGen(); B' ← AddrGen();

(ek_C, dk_C) ← EncGen(); C' ← AddrGen();

(ek_D, dk_D) ← EncGen(); D' ← AddrGen();
Step1: Announcement

A' ← AddrGen();
(ek_B, dk_B) ← EncGen(); B' ← AddrGen();
(ek_C, dk_C) ← EncGen(); C' ← AddrGen();
(ek_D, dk_D) ← EncGen(); D' ← AddrGen();

Input Addresses

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A:</td>
<td>B1</td>
</tr>
<tr>
<td>B:</td>
<td>B1</td>
</tr>
<tr>
<td>C:</td>
<td>B1</td>
</tr>
<tr>
<td>D:</td>
<td>B1</td>
</tr>
</tbody>
</table>
Step 1: Announcement

\[
\begin{align*}
A' &\leftarrow \text{AddrGen}(); \\
(ek_B, dk_B) &\leftarrow \text{EncGen}(); B' &\leftarrow \text{AddrGen}(); \\
(ek_C, dk_C) &\leftarrow \text{EncGen}(); C' &\leftarrow \text{AddrGen}(); \\
(ek_D, dk_D) &\leftarrow \text{EncGen}(); D' &\leftarrow \text{AddrGen}(); \\
\end{align*}
\]

\[
\begin{align*}
\text{Sign}(sk_A; A) \\
\text{Sign}(sk_B; ek_B, B) \\
\text{Sign}(sk_C; ek_C, C) \\
\text{Sign}(sk_D; ek_D, D)
\end{align*}
\]

Input Addresses

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: (B_1)</td>
<td>(B_1)</td>
</tr>
<tr>
<td>B: (B_1)</td>
<td>(B_1)</td>
</tr>
<tr>
<td>C: (B_1)</td>
<td>(B_1)</td>
</tr>
<tr>
<td>D: (B_1)</td>
<td>(B_1)</td>
</tr>
</tbody>
</table>

ek: encryption key
dk: decryption key
sk: signing key
Step 2: Shuffling

\[ A' \]

\[ e_{k_B} \]

\[ e_{k_C} \]

\[ e_{k_D} \]
Step 2: Shuffling
Step 2: Shuffling

The diagram illustrates the process of shuffling in a decentralized system. Each participant (A', B', C') exchanges messages (A' -> B' -> C') with different encryption keys (e_k_B, e_k_C, e_k_D) to secure the communication. The keys are used to encrypt and decrypt messages, ensuring the confidentiality of the communication between the participants. This method is inspired from the concepts of decentralized systems and speedup techniques.
Step 2: Shuffling
Step 3: Verification

<table>
<thead>
<tr>
<th>Input Addresses</th>
<th>Output Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: B1</td>
<td>B': B1</td>
</tr>
<tr>
<td>B: B1</td>
<td>D': B1</td>
</tr>
<tr>
<td>C: B1</td>
<td>A': B1</td>
</tr>
<tr>
<td>D: B1</td>
<td>C': B1</td>
</tr>
</tbody>
</table>

\(\sigma_A\) \(\sigma_B\) \(\sigma_C\) \(\sigma_D\)
Step 3: Verification

<table>
<thead>
<tr>
<th>Input Addresses</th>
<th>Output Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: 3B1</td>
<td>B': 3B1</td>
</tr>
<tr>
<td>B: 3B1</td>
<td>D': 3B1</td>
</tr>
<tr>
<td>C: 3B1</td>
<td>A': 3B1</td>
</tr>
<tr>
<td>D: 3B1</td>
<td>C': 3B1</td>
</tr>
</tbody>
</table>

Input Addresses

\[ \sigma_A \]
\[ \sigma_B \]
\[ \sigma_C \]
\[ \sigma_D \]

Output Addresses

\[ \sigma_A \]
\[ \sigma_B \]
\[ \sigma_C \]
\[ \sigma_D \]

(no signature for A)

Input Addresses

\[ \sigma_B \]
\[ \sigma_C \]
\[ \sigma_D \]

Output Addresses

\[ \sigma_B \]
\[ \sigma_C \]
\[ \sigma_D \]

Blame phase
Blame Process

A' -> A' -> B' -> B' -> E' -> C' -> D' -> Outputs

Outputs:
- D': B1
- B': B1
- E': B1
- C': B1

Other cases are discussed in the paper in detail.
# CoinShuffle

**Input Addresses**

<table>
<thead>
<tr>
<th></th>
<th>A: $3</th>
<th>B: $0</th>
<th>C: $1</th>
<th>D: $3</th>
</tr>
</thead>
</table>

**Output Addresses**

|   | B': $2 | D': $2 | A': $2 | C': $2 |

**Transactions**

- A → B : 2
- D → C : 1
- E → B' : 1
- B' → D' : 2
- C' → F : 1
Recall: The Blockchain

- Blockchain can be used to decentralize any centralized service:
  - Making them decentralized (without single-point-fault)
  - Public accountability

- We still have two problems:
  - How to achieve consensus?
  - How to preserve the privacy?
The Blockchain

BitCoin ≠ Blockchain
The Blockchain

BitCoin ≠ Blockchain

- Bank
- DNS service
- Uber

- AirBnb
- Food provenance
How to decentralize app via blockchain?

Log (or Ledger)

- What data we want to put as “transaction”
- The data is what we want to audit
Lecture Roadmap

- Blockchain Basics
- Case Study: BitCoin
- Case Study: CoinShuffle
- Summary of our course
UseNet (~1980)

PageRank

P2P lookup service

Attacks in P2P systems

Trust/Reputation

Google file system & Xen

MapReduce

Firewall and NATs

BigTable

P2P Computing

Cloud Computing
Thank you for taking my course