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
- Blockchain app & Smart Contract
- Summary of our course
BitCoin ≠ 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

D: $7

A -> B : 3

A: 10 - 3 = 7

B: $2

A -> B: 3

B: 2 + 3 = 5

C: $5
The Blockchain

A: $7

B: $5

D: $7

C: $5

A → B: 3

A: 10 - 3 = 7

B: 2 + 3 = 5
The Blockchain

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:
  
  1. **Decentralization**
  2. **Public accountability**
  3. **Efficiency**

\[ A \rightarrow B : 3 \]

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

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

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

A -> B : 3

A: $7
B: $5

D: $7
C: $5
The Blockchain

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

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

A -> B : 3

Ledger
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)

Will disk space become a burden?
Transactions are hashed in a Merkle Tree.
If we suppose blocks are generated every 10 minutes, then 4.2MB per year.
The Blockchain

Log (or Ledger)

- Each hash identifies the entire prefix of the log
Transactions in the Blockchain

A: $7
B: $5

A -> B : 4

D: $7
C: $5
Transactions in the Blockchain

A: $3

B: $9

D: $7

C: $5
Transactions in the Blockchain

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

A -> B : 4
Transactions in the Blockchain

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

A → B : 4
B → D : 1
Transactions in the Blockchain

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

A -> B : 4
B -> D : 1
I am the leader

A: $3

B: $8

C: $5

D: $8

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

Consensus
New Block Generation

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

New Block

A -> B : 4
B -> D : 1
New Block Generation

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
- Blockchain app & Smart Contract
- 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
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>ebf77560ca...1</td>
<td>8</td>
<td>3P9SgqzjFWg7VWAnZBFwimNPV7Lmu3PfgTj</td>
<td>Address</td>
<td>304502200078238c48ed1522d40eace4a73ae8fc3</td>
</tr>
<tr>
<td>b912994fa5...1</td>
<td>0.03</td>
<td>1GMke65wV1E5kCVHFSiyUTU6rd4yVF4M5F1</td>
<td>Address</td>
<td>3045022004e877e5c3a783f165052647c478dadd</td>
</tr>
<tr>
<td>58379494685...15</td>
<td>1</td>
<td>1G4hFvM2uAPFECdawg5tvUTBE2Pvxl42</td>
<td>Address</td>
<td>3044022075d236f4a90047667772106f146c96</td>
</tr>
<tr>
<td>69d1edc1c2ac...1</td>
<td>150</td>
<td>1LPQVnJSMggp8bQBpGzwo3bdY2Ghm9YWcC7</td>
<td>Address</td>
<td>3046022100a65a18889a4e5aee5a5ba387503</td>
</tr>
<tr>
<td>7b674a521c...1</td>
<td>0.55357267</td>
<td>16Kb6XppHbUbgmYQDprVynx9jNE9Azu5Xycb</td>
<td>Address</td>
<td>3045022100ceb76e61be62d38846262f1d11</td>
</tr>
<tr>
<td>5440973a30e90...0</td>
<td>0.03276928</td>
<td>1LmsDx1g6cc757z8AnUuemj6YQg5CTw54QJN</td>
<td>Address</td>
<td>304502210058b92ecd47493e8a489c4cc10615</td>
</tr>
</tbody>
</table>

<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>0.01071174</td>
<td>7TBgcQbyWTWzEMUKNzLdjk8icsQ79K96m</td>
<td>OP_DUP OP_HASH160 9ab2d20e0a63dea3657c33128e15d2746956184866</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1bb973b4ccce8</td>
<td>139.605567</td>
<td>NT2zFMa11NjCZyd4kgXRZPI6s6ZPGZ</td>
<td>OP_DUP OP_HASH160 eb471d7a903e958eb94e1f2f2a0eaddad8479af</td>
<td></td>
</tr>
</tbody>
</table>
How to compute BitCoin?
How to compute BitCoin?

...
Who should generate a new block to include these two transactions?

A -> D : 1
E -> D : 3

A -> B : 1
C -> D : 3

A -> B : 1
A -> C : 1

A -> B : 4
C -> D : 1
They need to compete, and the winner can earn money.

A->D : 1
E->D : 3

A->B : 1
C->D : 3

A->B : 1
A->C : 1

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

How to compute BitCoin?
How to compute BitCoin?

X = SHA256(H + salt)
X 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 d7a8fbb307d7809469ca9acb0082e4f8d5651e46d3c4b762d02d0bf37c9e592
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 d7a8fbb307d7809469ca9abcb0082e4f8d5651e46d3cde762d02d0bf37c9e592
SHA256("The quick brown fox jumps over the lazy dog.")
0x ef537f25c895bfa782526529a9b63d97aa631564d5d789c2b765448c8635fb6c

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

Salt=8

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
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

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
  - 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

Miners in BitCoin can earn a lot of money!
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
- Blockchain app & Smart Contract
- Summary of our course
Linkability of Pseudonyms
Linkability of Pseudonyms

[Meiklejohn et al., IMC’13]
Setup

Alice

Bob

Carol

Dave

$(sk_A, vk_A) \quad (sk_B, vk_B) \quad (sk_C, vk_C) \quad (sk_D, vk_D)$
Step 1: Announcement

\[
A' \leftarrow \text{AddrGen}();
\]

\[
(ek_B, dk_B) \leftarrow \text{EncGen}(); B' \leftarrow \text{AddrGen}();
\]

\[
(ek_C, dk_C) \leftarrow \text{EncGen}(); C' \leftarrow \text{AddrGen}();
\]

\[
(ek_D, dk_D) \leftarrow \text{EncGen}(); D' \leftarrow \text{AddrGen}();
\]
Step1: Announcement

\[ A' \leftarrow \text{AddrGen}(); \]
\[ (\text{ek}_B, \text{dk}_B) \leftarrow \text{EncGen}(); B' \leftarrow \text{AddrGen}(); \]
\[ (\text{ek}_C, \text{dk}_C) \leftarrow \text{EncGen}(); C' \leftarrow \text{AddrGen}(); \]
\[ (\text{ek}_D, \text{dk}_D) \leftarrow \text{EncGen}(); D' \leftarrow \text{AddrGen}(); \]

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

ek: encryption key
dk: decryption key
sk: signing key
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();

Sign(sk_A; A)

Sign(sk_B; ek_B, B)

Sign(sk_C; ek_C, C)

Sign(sk_D; ek_D, D)

Input Addresses

<table>
<thead>
<tr>
<th>A</th>
<th>B'</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B'</td>
</tr>
</tbody>
</table>

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

\[ \text{Step 2: Shuffling} \]

\[ \begin{align*}
&A' \\
&\text{ek}_B \\
&\text{ek}_C \\
&\text{ek}_D
\end{align*} \]
Step 2: Shuffling

The diagram illustrates the process of shuffling with the following steps:

1. **Step 1:** Each participant (A', B', C', D') generates a key (ek_B, ek_C, ek_D) for the shuffle.

2. **Step 2:** The keys are used to shuffle the participants' outputs (A', B').
Step2: Shuffling
Step 2: Shuffling

1. Inspired from Nessent [B?ibbs and ordf[qqSffi02

Outputs

D': $1
B': $1
A': $1
C': $1
### 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>

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

![Bitcoin logo](bitcoin.png)
### Step 3: Verification

**Input Addresses** | **Output Addresses**
--- | ---
A: B | B': B
B: B | D': B
C: B | A': B
D: B | C': B

**Input Addresses** | **Output Addresses**
--- | ---
A: B | B': B
B: B | D': B
C: B | C': B
D: B | E': B

(no signature for A)

---

**Input Addresses** | **Output Addresses**
--- | ---
A: B | B': B
B: B | D': B
C: B | C': B
D: B | E': B

**Blame phase**
Blame Process

The diagram illustrates the blame process involving several individuals and their interactions. The process begins with an initial input (A') leading to subsequent steps involving B', C', and E'. The outputs are categorized into D': B1, B': B1, E': B1, and C': B1. The diagram also shows individuals labeled dk_A, dk_B, dk_C, and dk_D, each possibly representing different roles or stages in the blame process.
CoinShuffle

A → B : 2
D → C : 1

A: $4
B: $0
C: $1
D: $3

E → B' : 1
B' → D' : 2
C' → F : 1

Input
Addresses
A: ₿$2
B: ₿$2
C: ₿$2
D: ₿$2

Output
Addresses
A': ₿$2
B': ₿$2
D': ₿$2
C': ₿$2
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
- **Blockchain app & Smart Contract**
- Summary of our course
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
Smart Contract

A -> B : 3

Contract:
If xx
then yy
Smart Contract

A->B : 3

Account Balance

<table>
<thead>
<tr>
<th>Account</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>25</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>15</td>
</tr>
<tr>
<td>F</td>
<td>16</td>
</tr>
<tr>
<td>G</td>
<td>23</td>
</tr>
<tr>
<td>H</td>
<td>42</td>
</tr>
<tr>
<td>I</td>
<td>10</td>
</tr>
</tbody>
</table>

If this, then that

Contract:
If xx
then yy
Example

- You are planning to ship a laptop to your friend Bob
  - You trust Bob, but you do not trust trucker Tom
  - Tom will carry your laptop
  - Tom does not trust since maybe you will not pay him
Example

- You are planning to ship a laptop to your friend Bob
  - You trust Bob, but you do not trust trucker Tom
  - Tom will carry your laptop
  - Tom does not trust since maybe you will not pay him

You and Tom have to sign a contract.
Example

- We can use smart contract:
  - You and Time define all the rules in code
  - You make a payment for shipment to smart contract on a day of loading.
  - It holds payment till shipment delivery is confirmed by Bob.
  - Smart contract releases the payment and money is transferred to Tom automatically.
Another Example

Doctor informs patient that they need to exercise.
Another Example

Doctor informs patient that they need to exercise

Patient agrees to exercise regime
Another Example

Doctor informs patient that they need to exercise

Patient agrees to exercise regime

A “HealthCoin” is placed – a smart contract – is placed in the patient's wallet (with demurrage)

A ledger records all changes
Another Example

Doctor informs patient that they need to exercise

Patient agrees to exercise regime

A “HealthCoin” is placed – a smart contract – is placed in the patient's wallet (with demurrage)

As an individual performs agreed on actions, health coins change (either go up or down) – tracked by wearable
Another Example

1. Doctor informs patient that they need to exercise.
2. Patient agrees to exercise regime.
3. A "HealthCoin" is placed — a smart contract is placed in the patient's wallet (with demurrage).
4. As an individual performs agreed on actions, health coins change (either go up or down) — tracked by wearable.
5. If the individual completes they are rewarded or penalized if they fail.

A ledger records all changes.

Shared Ledger
Lecture Roadmap

• Blockchain Basics
• Case Study: BitCoin
• Case Study: CoinShuffle
• Blockchain app & Smart Contract
• Summary of our course

Trust/Reputation
Thank you for taking my course