MONASH University

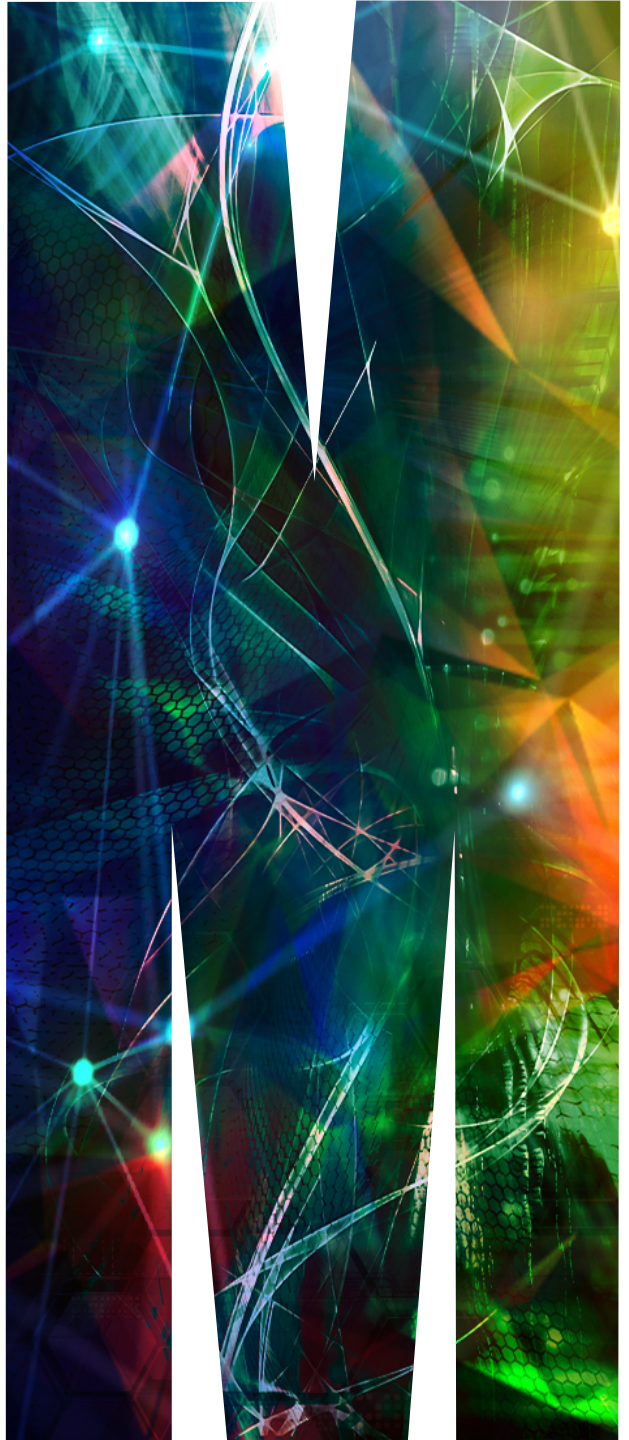
FIT5214: Blockchain

Lecture 12: Payment Channels

Lecturer: Rafael Dowsley

rafael.dowsley@monash.edu

https://dowsley.net



Unit Structure

- Lecture 1: Introduction to Blockchain
- Lecture 2: Bitcoin
- Lecture 3: Ethereum and Smart Contracts
- Lecture 4: Proof-of-Work (PoW)
- Lecture 5: Attacks on Blockchains
- Lecture 6: Class Test/Alternatives to PoW
- Lecture 7: Proof-of-Stake (PoS)
- Lecture 8: Privacy
- Lecture 9: Byzantine Agreement
- Lecture 10: Algorand
- Lecture 11: Blockchain Network
- Lecture 12: Payment Channels

Unit Structure

- Lecture 1: Introduction to Blockchain
- Lecture 2: Bitcoin
- Lecture 3: Ethereum and Smart Contracts
- Lecture 4: Proof-of-Work (PoW)
- Lecture 5: Attacks on Blockchains
- Lecture 6: Class Test/Alternatives to PoW
- Lecture 7: Proof-of-Stake (PoS)
- Lecture 8: Privacy
- Lecture 9: Byzantine Agreement
- Lecture 10: Algorand
- Lecture 11: Blockchain Network
- Lecture 12: Payment Channels

Learning outcome:

Have basic understandings on how to use layer-2 protocols to scale blockchain.

I will also talk about the exam, give a recap and answer your questions.

Last Reminder

Your feedback is extremely important to us! You have a chance to provide:

(1) formal feedback about your learning experience

SETU: Go directly to https://monash.bluera.com/monash or follow the link from Moodle sidebar/Moodle page of Week 12

(2) your recognition to a teaching staff/unit

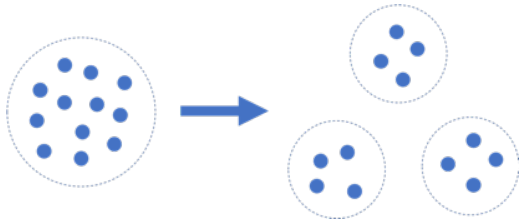
Teaching Award Nomination: https://www.intranet.monash/it/education/ed-quality/awards

MONASH University

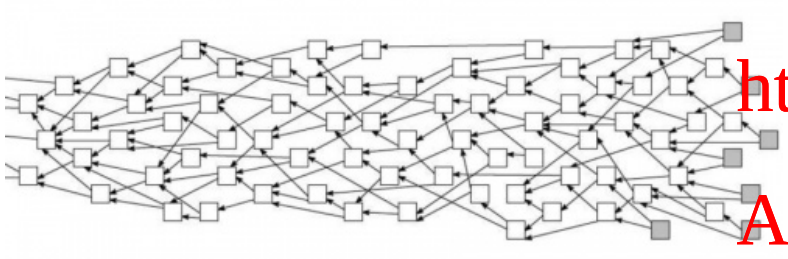
Problem - scalability

Bitcoin does not scale: it can only processe 7 tx per second

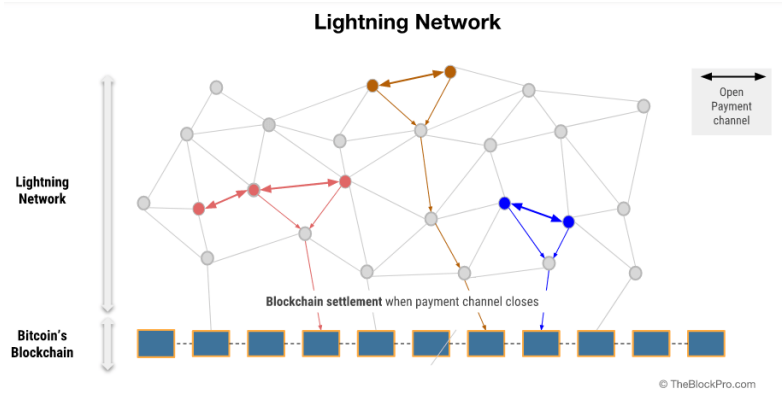
Improving scalability



Database partitioning
e.g. sharding



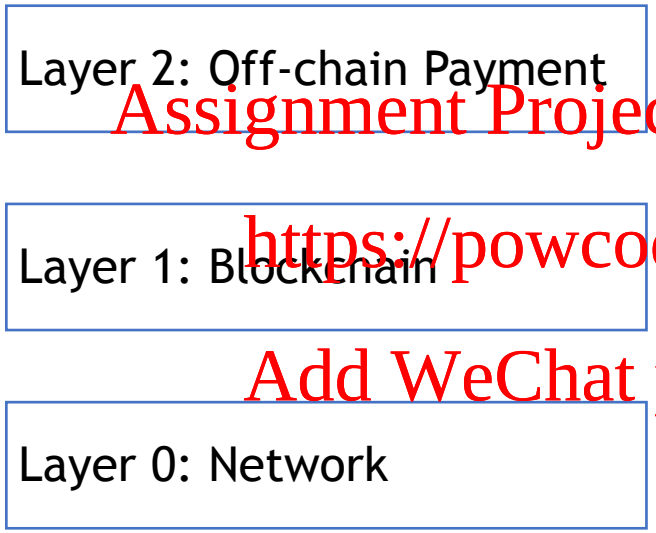
New data structure
e.g. Directed Acyclic Graph



Off-chain payment
e.g. Lightning network

MONASH University

Layers of blockchain



Layer-2 protocol

Alice and Bob can trade without having to go through the blockchain.

They only need to record a settlement (a summary of their transactions) on the chain.

Threshold signature

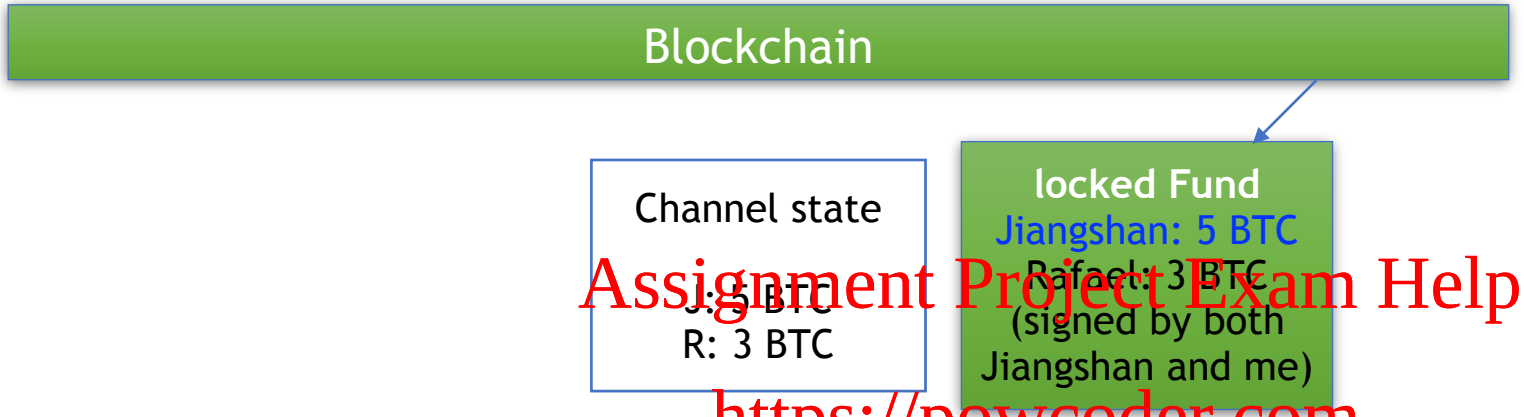
K out of N parties need to work together to create a valid signature.

Example application:

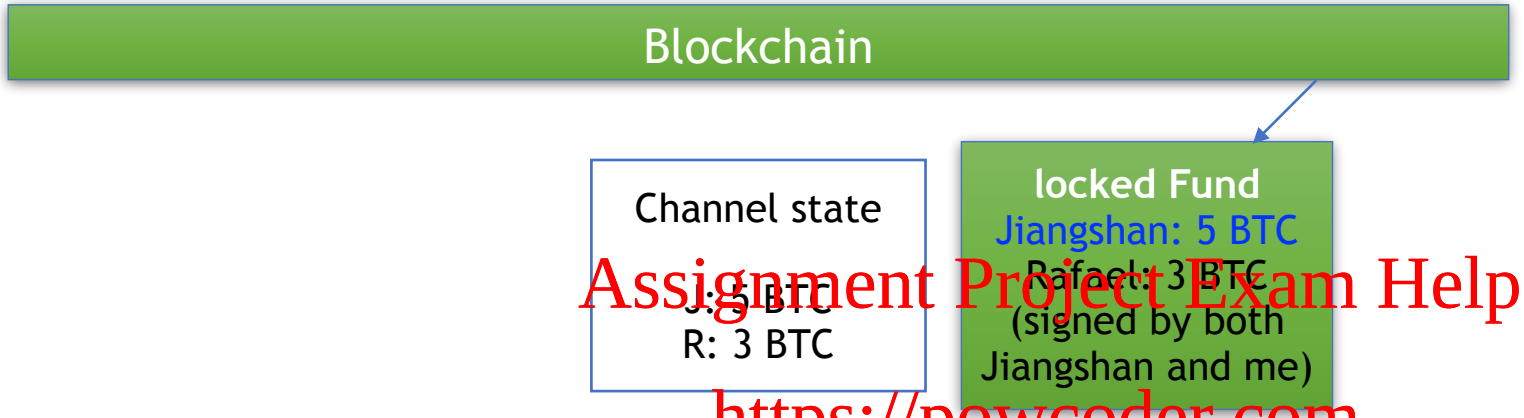
- Jointly owned Bitcoin account
- Two-factor authentication wallet
- Backup of keys

Payment channel (basic idea)

Payment channel (basic idea)

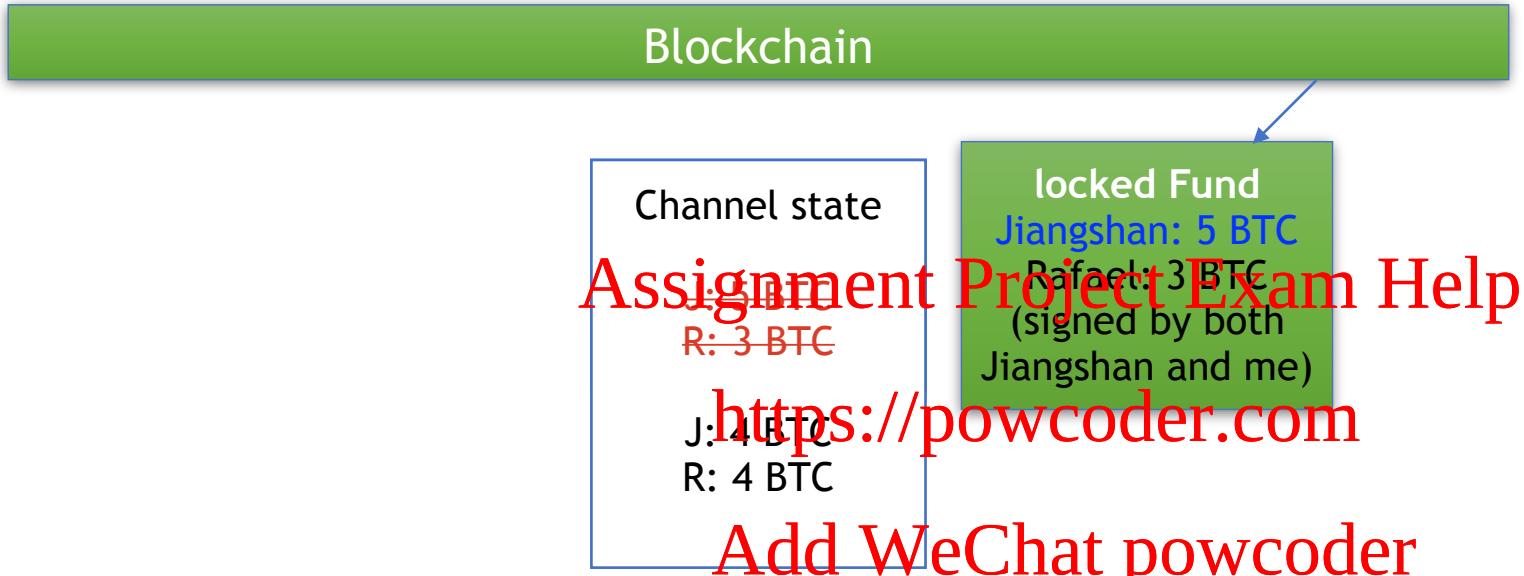


Payment channel (basic idea)



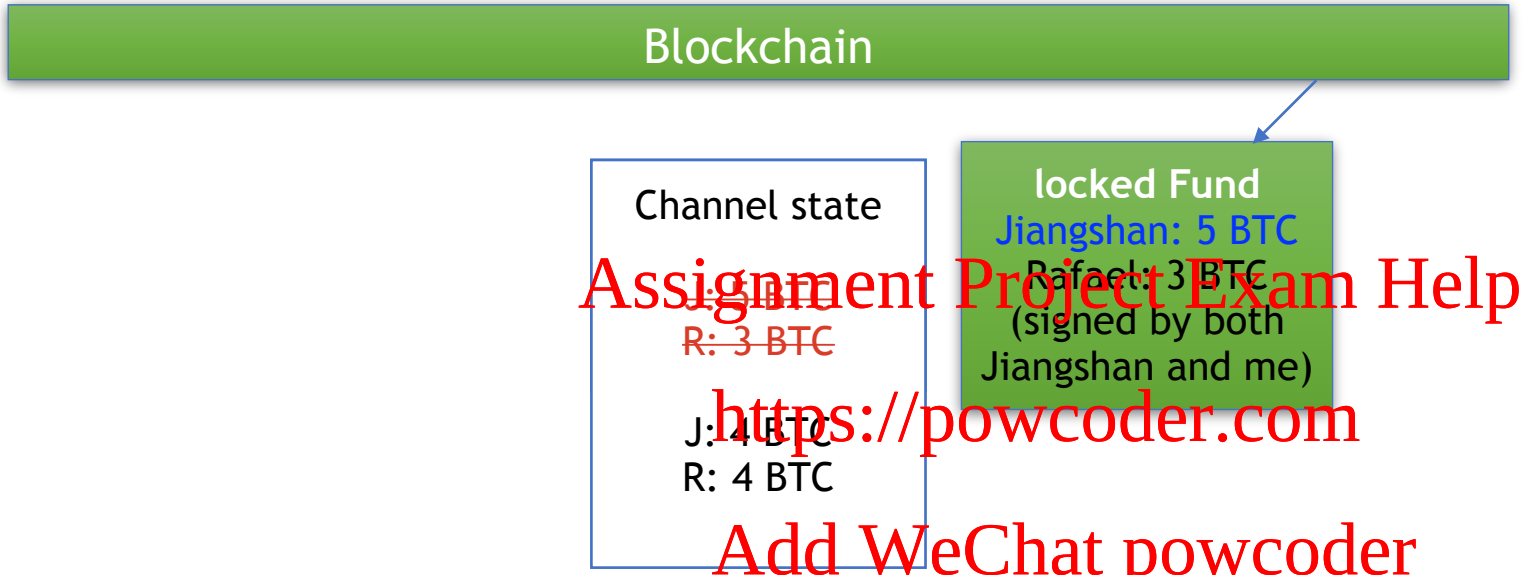
State 1: Transfer 1 BTC to R (channel state signed by J)

Payment channel (basic idea)



State 1: Transfer 1 BTC to R (channel state signed by J)

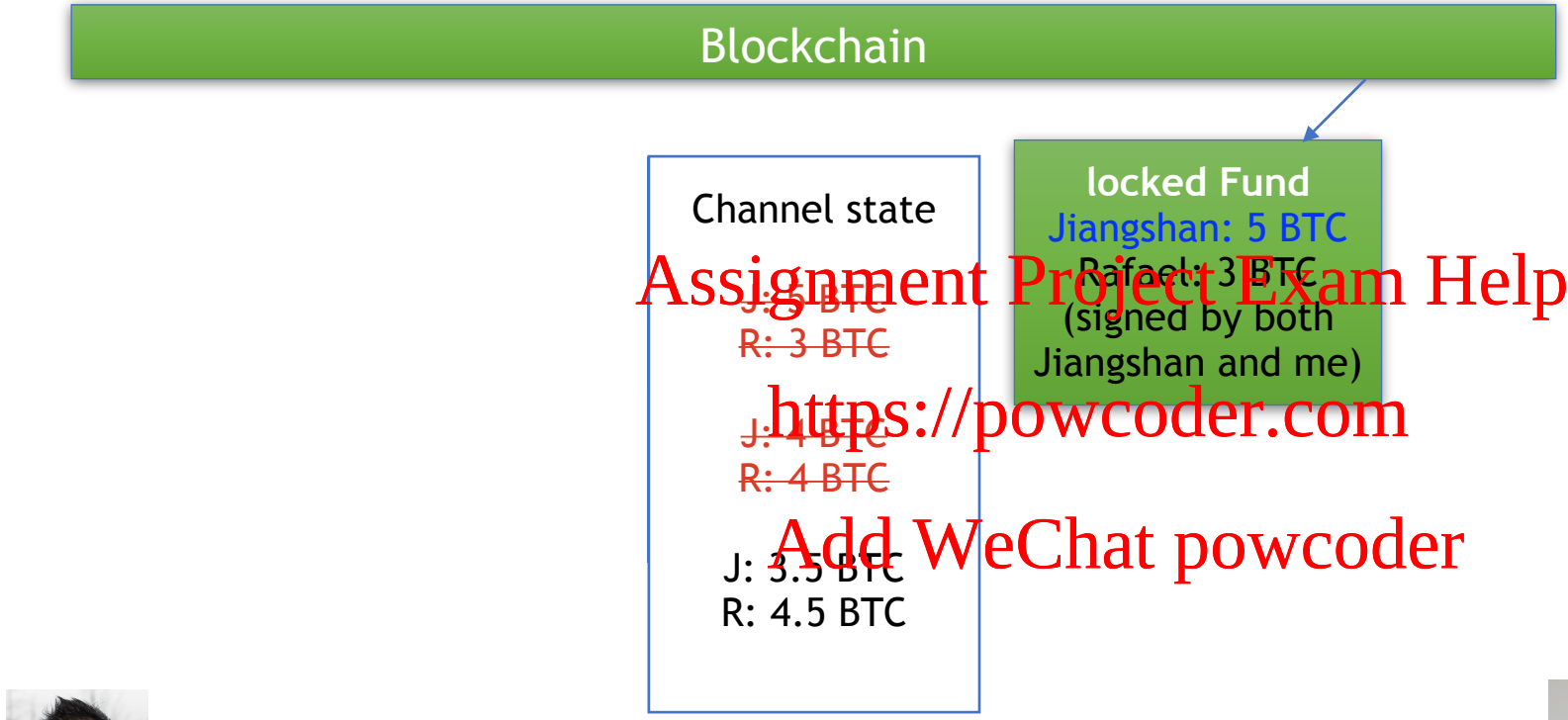
Payment channel (basic idea)



State 1: Transfer 1 BTC to R (channel state signed by J)

State 2: Transfer 0.5 BTC to R (channel state signed by J)

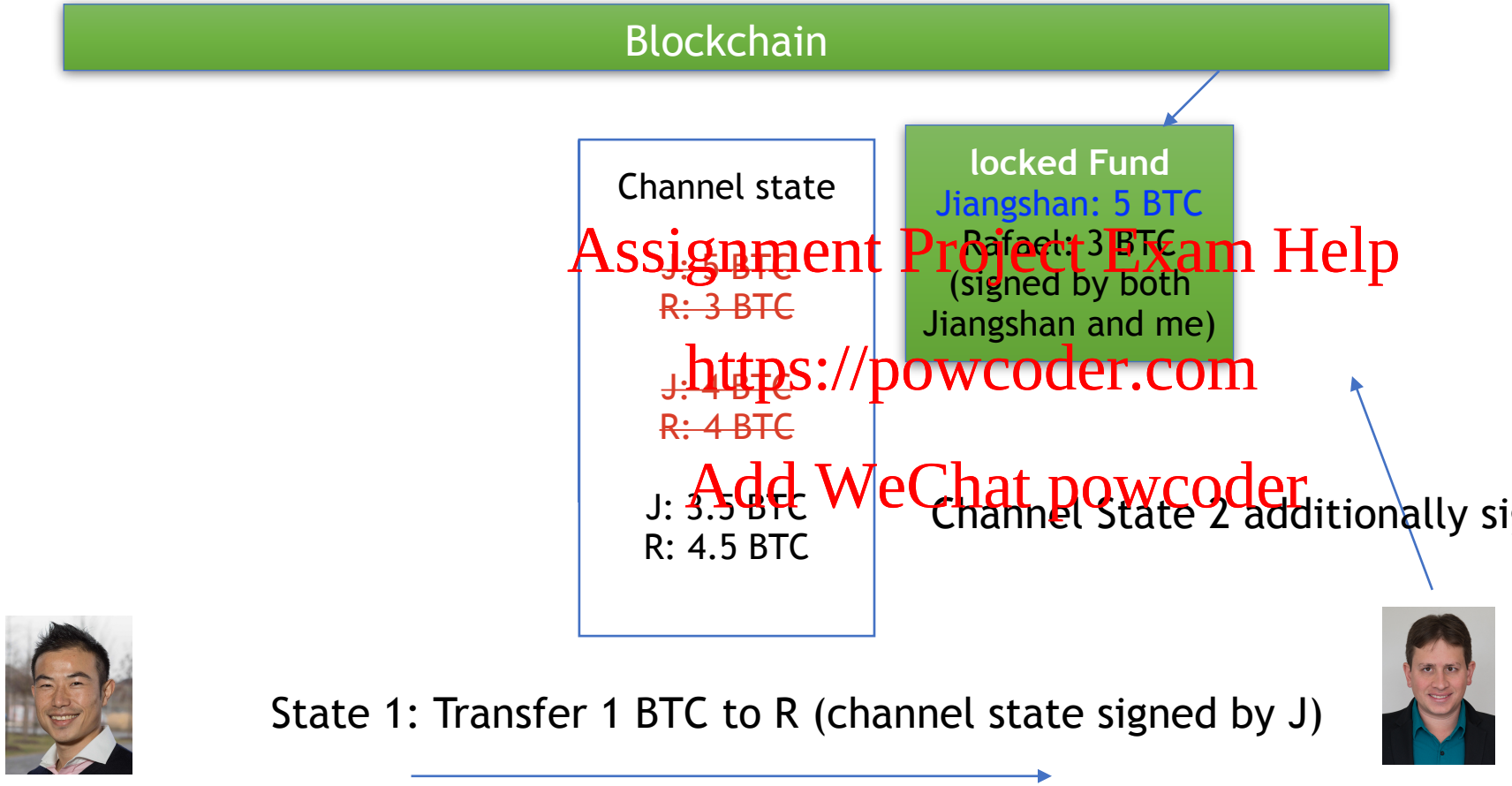
Payment channel (basic idea)



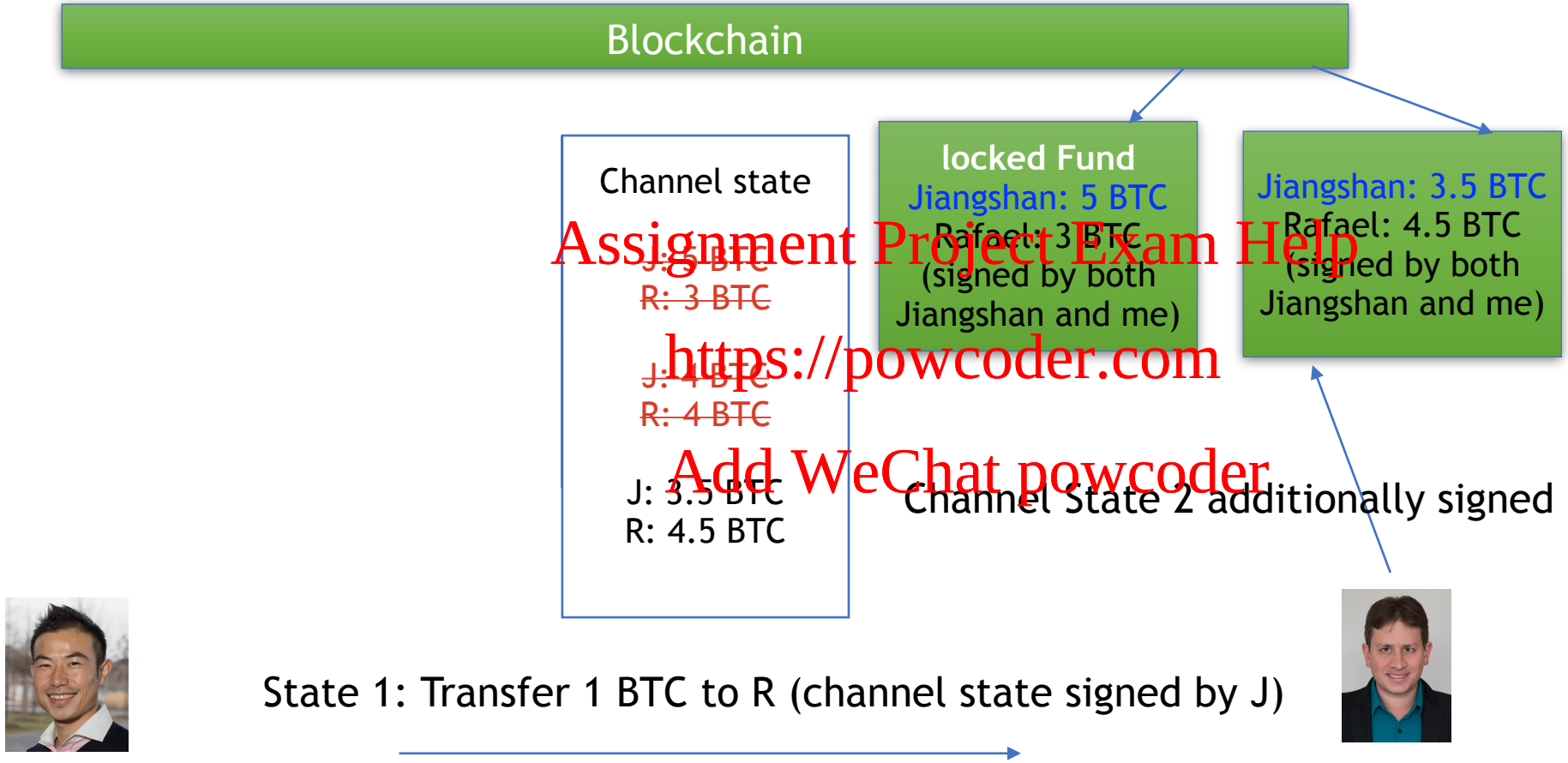
State 1: Transfer 1 BTC to R (channel state signed by J)

State 2: Transfer 0.5 BTC to R (channel state signed by J)

Payment channel (basic idea)



Payment channel (basic idea)



State 1: Transfer 1 BTC to R (channel state signed by J)

State 2: Transfer 0.5 BTC to R (channel state signed by J)

MONASH University

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

I want to send money to Maxime, but just once. I don't want to open a channel as it requires operations on blockchain.

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

How to transfer the money without the need of trusting R or L?

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

How to transfer the money without the need of trusting R or L?

Group discussion:

How to reduce the risk?

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

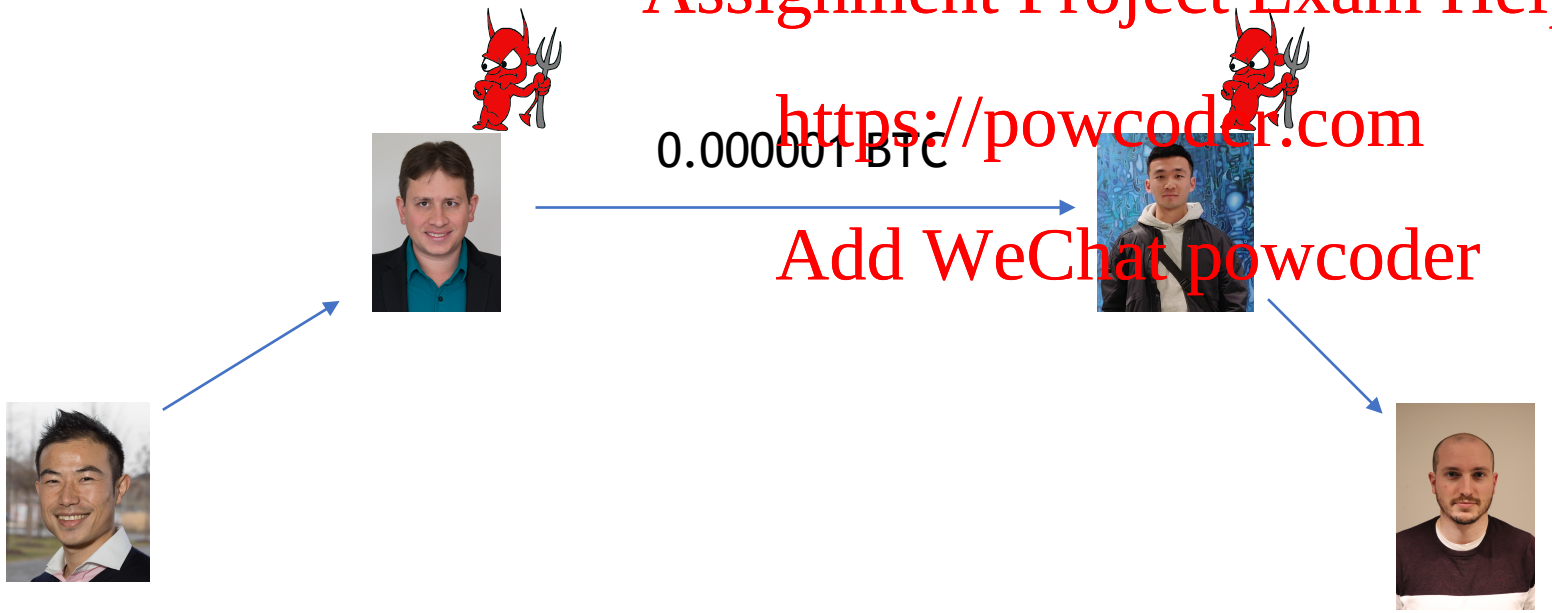
I want to send 10 BTC to Maxime.

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

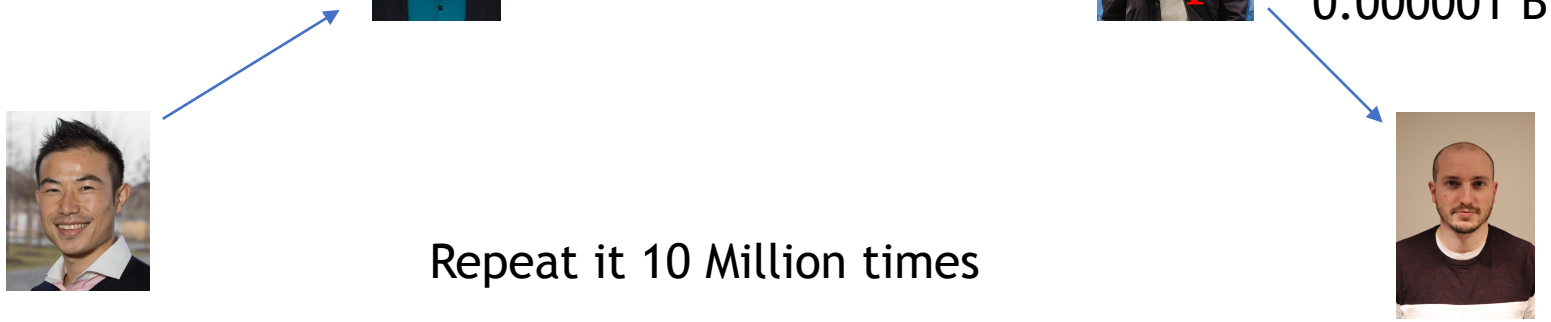


Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

Payment channel (basic idea)

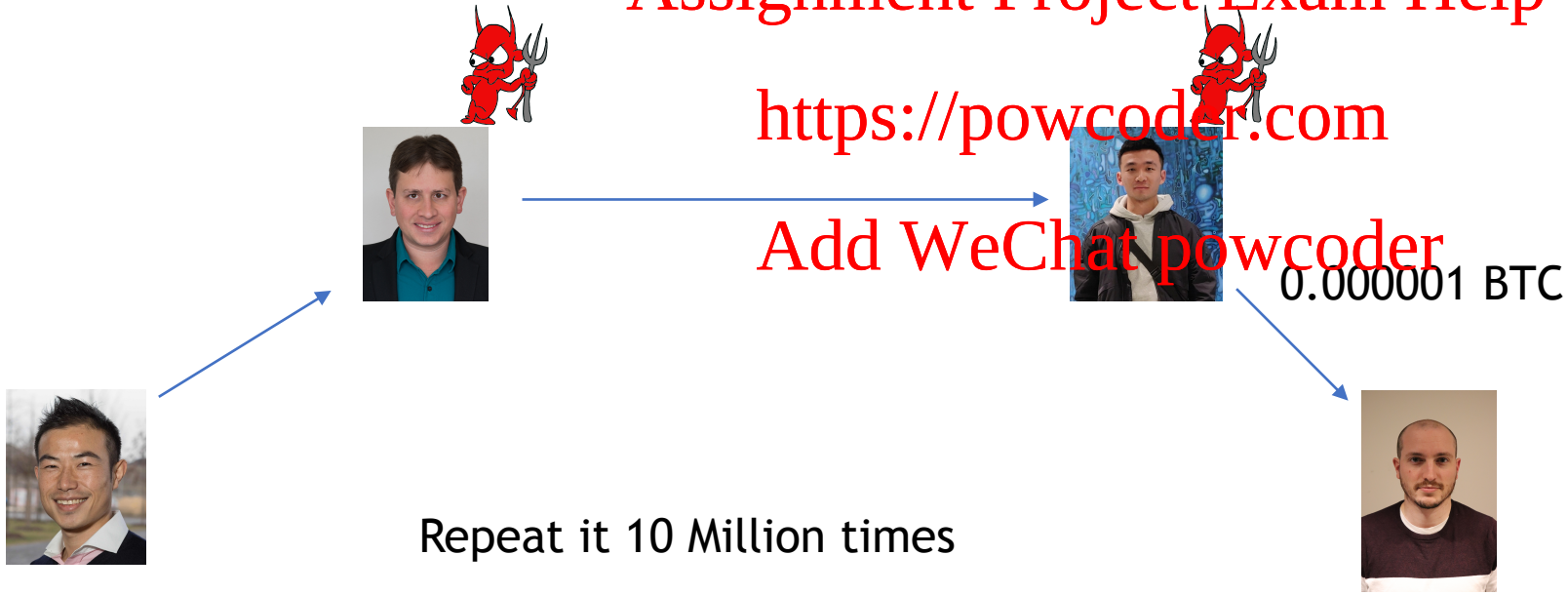
Every transaction to a new recipient require a new channel, and it may only be used once.



Repeat it 10 Million times

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.

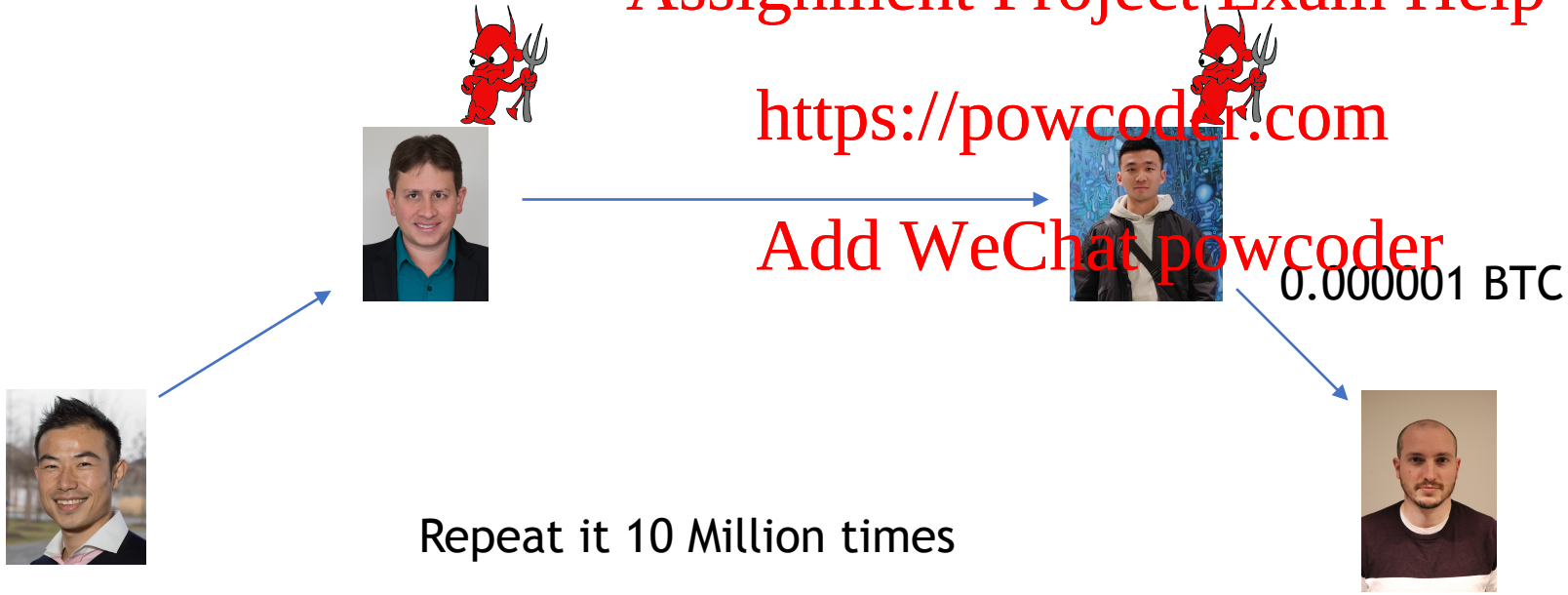


Repeat it 10 Million times

At most 0.000001 BTC is lost.

Payment channel (basic idea)

Every transaction to a new recipient require a new channel, and it may only be used once.



Repeat it 10 Million times

At most 0.000001 BTC is lost.

This reduces the potential damage, but not efficient.

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

Random r

h=H(r)

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

Random r

h=H(r)

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

r is the proof of successfully sent money to M.

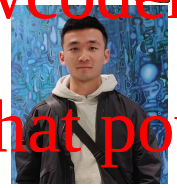
Random r

h=H(r)

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to R if you know r, within time t



r is the proof of successfully sent money to M.

Random r

h=H(r)

MONASH University

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to L if you know r, within time t

10 BTC to R if you know r, within time t

r is the proof of successfully sent money to M.

Random r

h=H(r)

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to L if you know r, within time t

10 BTC to R if you know r, within time t

10 BTC to M if you know r, within time t

r is the proof of successfully sent money to M.

Random r

h=H(r)

MONASH University

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to L if you know r, within time t

10 BTC to R if you know r, within time t

10 BTC to M if you know r, within time t

r is the proof of successfully sent money to M.

Release r and get 10 BTC

Random r

h=H(r)

MONASH University

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

Group discussion:

What can go wrong?

10 BTC to L if you know r, within time t

10 BTC to R if you know r, within time t

10 BTC to M if you know r, within time t

r is the proof of successfully sent money to M.

Release r and get 10 BTC

Random r

h=H(r)

MONASH University

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to L if you know r, within time t

10 BTC to R if you know r, within time t

10 BTC to M if you know r, within time t

r is the proof of successfully sent money to M.

Release r and get 10 BTC

Random r

h=H(r)

M may release r right before the time t finishes.
R and L have no time to get money!

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to L if you know r, within time t

10 BTC to R if you know r, within time t

10 BTC to M if you know r, within time t

r is the proof of successfully sent money to M.

Release r and get 10 BTC

Random r

h=H(r)

M may release r right before the time t finishes.
R and L have no time to get money!

MONASH University

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

Random r

h=H(r)

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

Random r

h=H(r)

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

r is the proof of successfully sent money to M.

Random r

h=H(r)

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to R if you know r, within X blocks

r is the proof of successfully sent money to M.

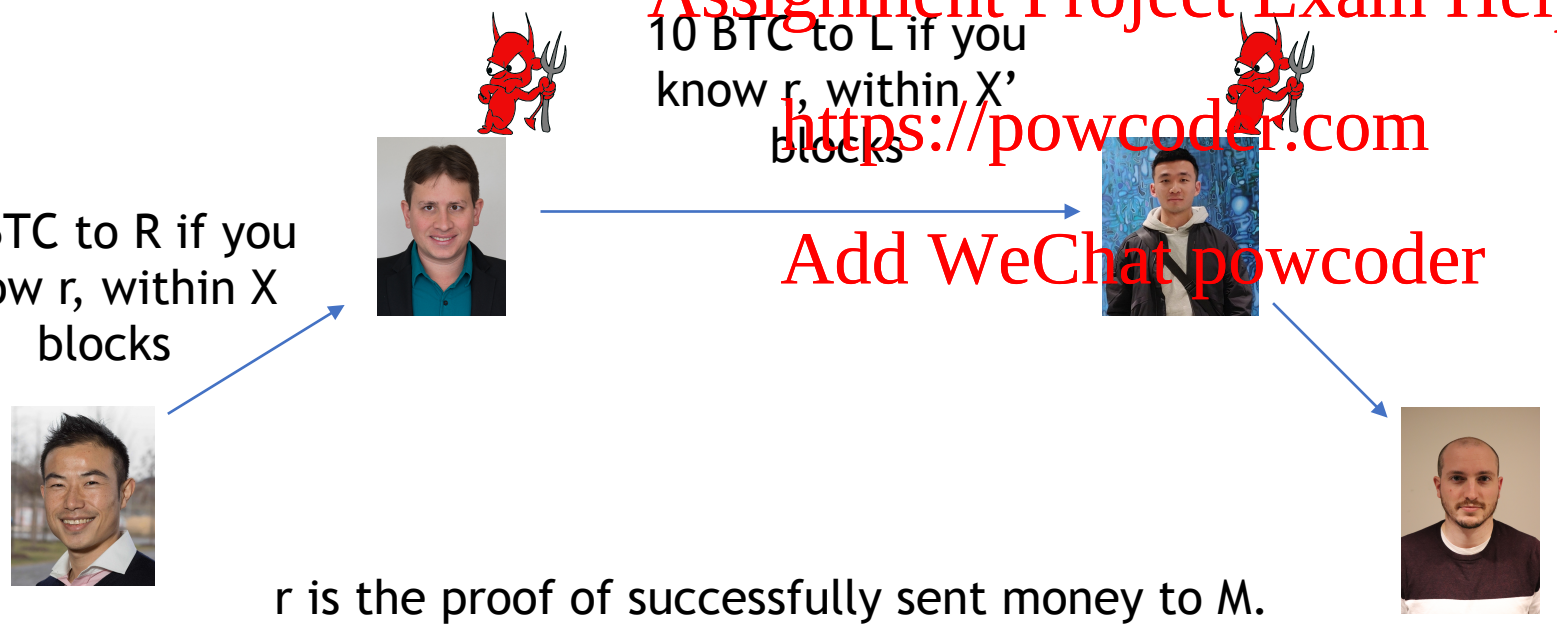
Random r

h=H(r)

MONASH University

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.



Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to L if you know r, within X' blocks

10 BTC to R if you know r, within X blocks

10 BTC to M if you know r, within X'' blocks

r is the proof of successfully sent money to M.

Random r

h=H(r)

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to L if you know r, within X' blocks

10 BTC to R if you know r, within X blocks

10 BTC to M if you know r, within X'' blocks

r is the proof of successfully sent money to M.

Release r and get 10 BTC

Random r

h=H(r)

Hash time locked contract

Every transaction to a new recipient require a new channel, and it may only be used once.

10 BTC to L if you know r, within X' blocks

10 BTC to R if you know r, within X blocks

10 BTC to M if you know r, within X'' blocks

Release r and get 10 BTC

r is the proof of successfully sent money to M.

Random r

h=H(r)

Time of generating Min(X-X', X'-X'') blocks >> the time to run the protocol

MONASH University

Exam…

- ★ 50 marks in total
- ★ 14 questions in total
  - ★ 10 multiple choice questions with multiple answers (3 marks each).
  - ★ 4 short answers questions (5 marks each).
- ★ Covering topics from **Week 1** to **Week 12.**

Exam…

★ Multiple choice questions with multiple answers:

Which of the following protocols use Proof-of-Work?

- Algorand
- Bitcoin
- Ethereum
- Ouroboros Praos

Exam...

★ Multiple choice questions with multiple answers:

Which of the following protocols use Proof-of-Work?

- Algorand
- Bitcoin
- Ethereum
- Ouroboros Praos

Every right answer that is selected will give positive marks, every wrong answer that is selected will deduct marks (overall the grade of each question is between 0.0 and 3.0).

Exam…

★ Short answers questions:

A standard ETH transfer requires a gas limit of _____ units of gas? (Just type the numerical answer)

21000

Recap

- Lecture 1: Introduction to Blockchain
  - Basic concepts about blockchain
- Lecture 2: Bitcoin
  - Crypto primitives (hash functions, birthday attack, Merkle trees, signature scheme)
  - How Bitcoin works
  - Hard/soft forks
  - Blocks, transactions, fees
  - Wallets
  - UTXO
- Lecture 3: Ethereum and Smart Contracts
  - Ethereum and smart contracts
  - Ethereum consensus, uncle blocks, different rewards
  - EVM, solidity, etc.
  - Ether, Gas, transactions, etc.
  - Different attacks

Recap

- Lecture 4: Proof-of-Work
  - Bitcoin consensus
  - Design flaws and simple attacks
  - Formal properties of PoW consensus
- Lecture 5: Attacks on Blockchains
  - Selfish mining attack
  - Eclipse attack
  - 51% attack
- Lecture 6: Alternatives to PoW
  - Proof-of-Elapsed-Time
  - Proof-of-Capacity
  - Proof-of-Personhood

Recap

- Lecture 7: Proof-of-Stake
  - Basics of PoS
  - Attacks on PoS
  - Ouroboros Praos
  - Key-Evolving Signature Schemes
  - Verifiable Random Function
- Lecture 8: Privacy
  - Privacy properties
  - Bitcoin privacy
  - (Linkable) Ring signature
  - CryptoNote/Monero
  - 0-mixin attack
  - Passive/active inference attacks

Recap

- Lecture 9: Byzantine Agreement
  - Byzantine generals problem
  - Different network conditions
  - PBFT and quorum
- Lecture 10: Algorand
- Lecture 11: Blockchain Network
  - How to discover new peers
  - How to connect to new peers
  - Different P2P protocols
- Lecture 12: Payment Channels

Questions?

Questions?

Best wishes in your exam!