SHANDONG UNIVERSITY

Analysis of the Blockchain Protocol with Long Delays

Puwen Wei [1], Quan Yuan [1], Yuliang Zheng [2]

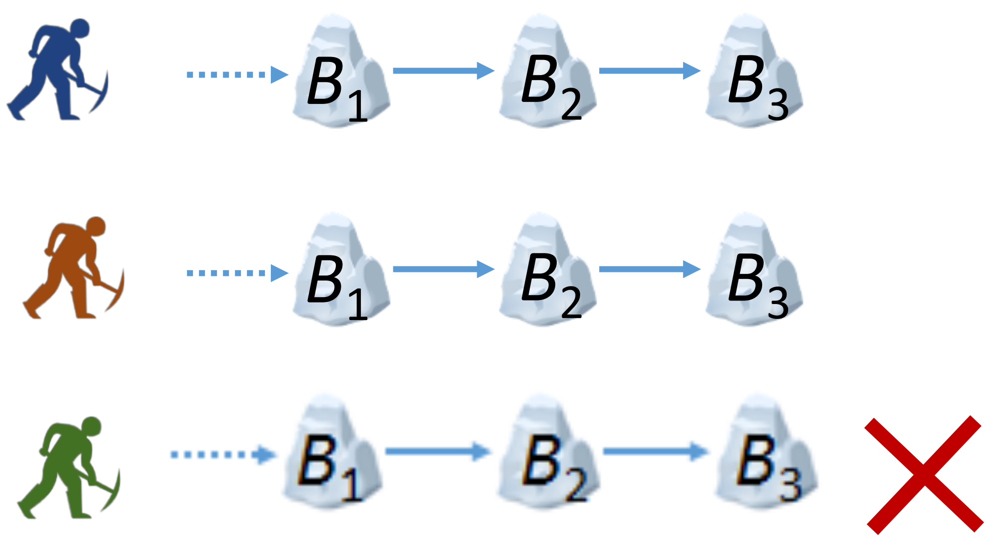
1. Shandong University
Key Lab of Cryptologic Technology and Information Security, Ministry of Education
2. University of Alabama at Birmingham

Nakamoto's blockchain

- Bitcoin introduced by Nakamoto in 2008
  - Decentralized payment system
    - Ledger maintained by the public in a decentralized manner
    - Attractive properties
      - Decentralization, Pseudonymity, Robustness …

Nakamoto's blockchain

- Blockchain
  - Chain-structured ledger maintained by all the participants (miners)
    - Blocks can only be added to the end of the chain
  - Basic security requirement
    - All the miners maintain the same record
    - Achieve **consensus** in the permissionless setting

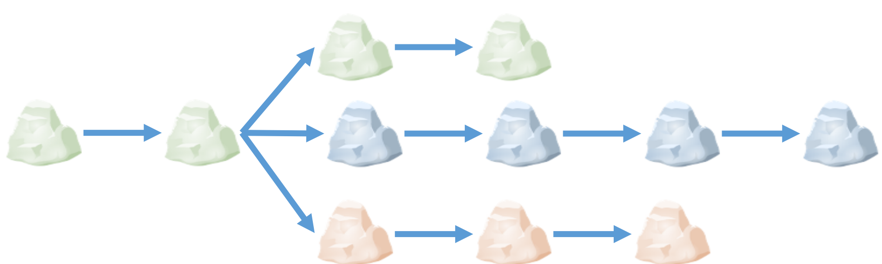


permissionless
anyone can join (or leave) the protocol execution

Nakamoto's blockchain

- Proof of work (POW)
  - Solve a "cryptographic puzzle"
    - Integrity：More difficult for the adversary to modify the chain
    - Synchronism：help the distributed miners to synchronize
      - Slowdown the generation of blocks
      - Longest chain rule

$$H(h||m||?) < D$$



Bitcoin Backbone Protocol [GKL15]

blockchain C=$(B_0, B_1, \ldots, B_l)$

block $B_i = (h_{i-1}, m_i, r_i, h_i)$

$h_i = H(h_{i-1}||m_i||r_i)$, s.t. $h_i < D$

Nakamoto's blockchain

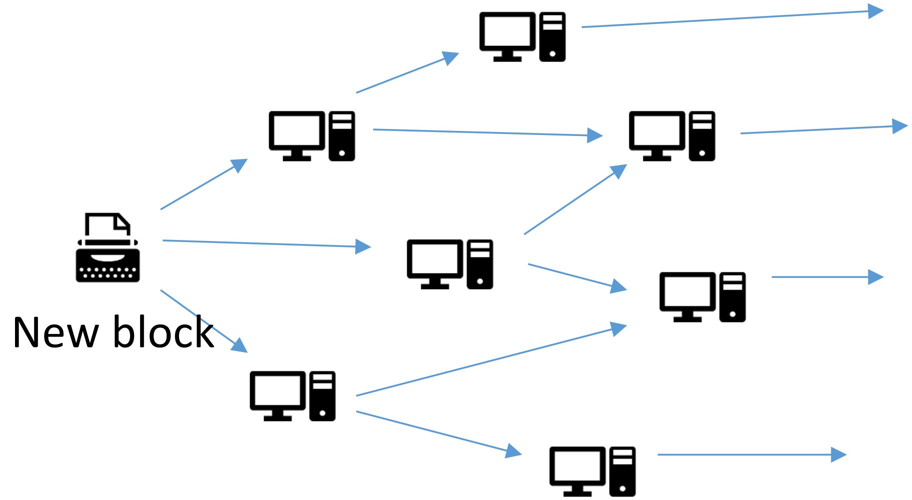
Common prefix | **Chain growth** | **Chain quality**

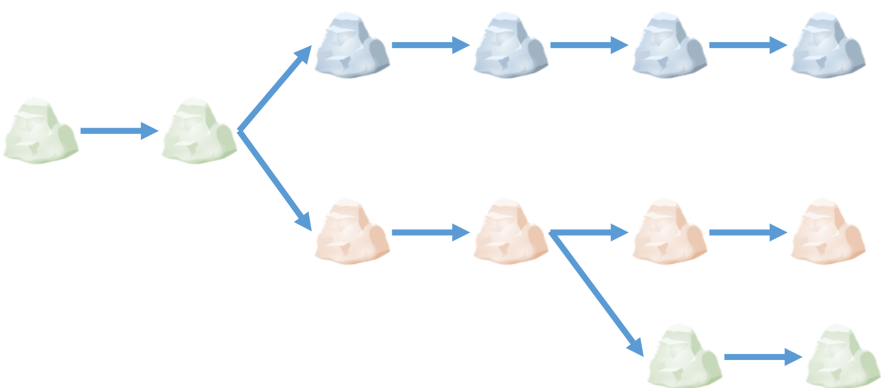
- Security
  - Garay, Kiayias and Leonardos [GKL15] provide a rigorous analysis of blockchain protocol
    - Synchronous model
  - Pass, Seeman and shelat [PSS17] analyze the security in an asynchronous network with a-priori bounded delay
    - Asynchronous model

Why consider the delay?

Blockchain protocol with delays

- Bitcoin P2P network
  - Delays are inevitable

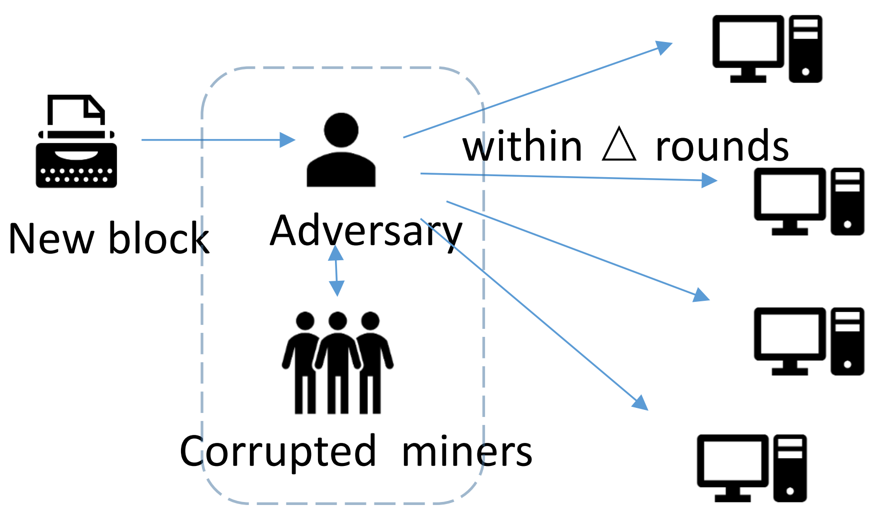




- The propagation delay in the network is the primary cause for blockchain forks [DW13]

Blockchain protocol with delays

- Adversary in [PSS17]
  - Responsible for the all message delivery
    - All the message can be delayed within Δ rounds
  - Has certain factions of hash power

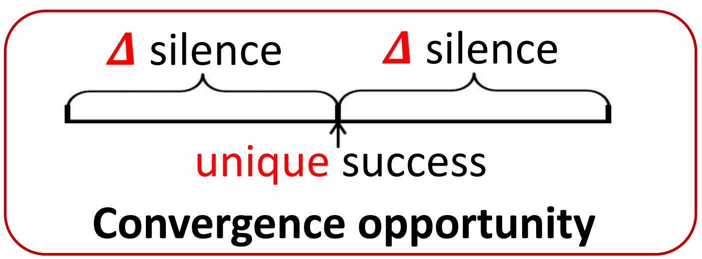


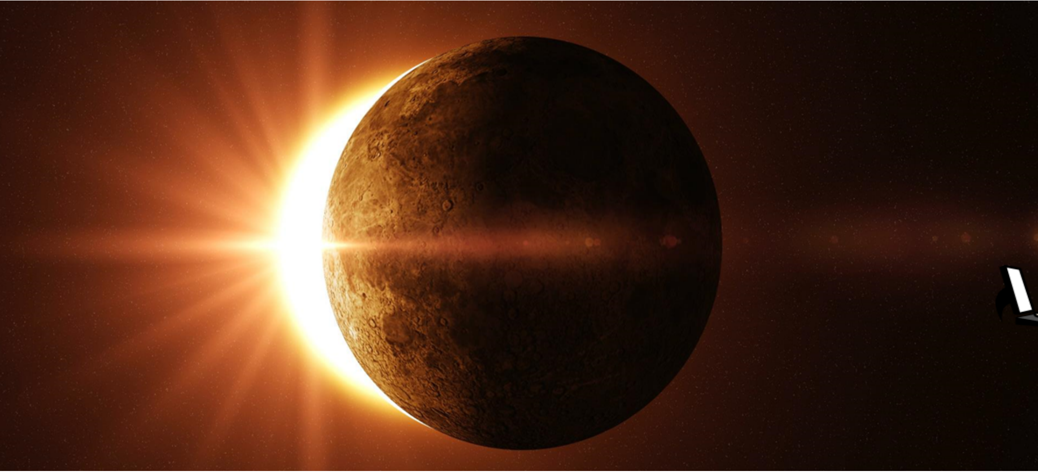
- Chain growth: $\frac{(1-\epsilon)f}{1+f\Delta}$, where $f \approx np$
- Consistency: T with probability $1 - negl(T)$
- Chain quality: $1-(1+\epsilon)\frac{tp(1+f\Delta)}{f}$

- **Limitation: $\Delta \ll O(1/np)$**

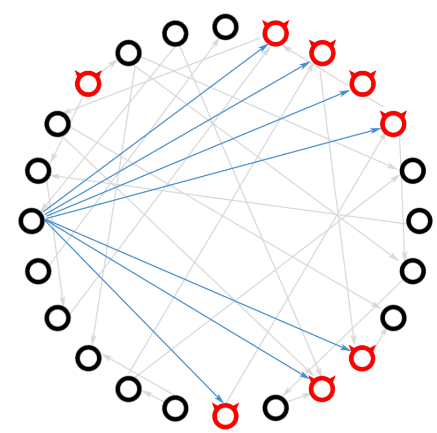
The proof holds for a relatively small delay only

n: the number of miners
p: the probability that a miner succeeds in mining a block at a round





- In the real world, long delays, say $\Delta \geq 1/np$, could be caused easily!
  - "bad" asynchronous networks, equipment failure,…
  - malicious attacks
    - eclipse attacks [HKZG15], which allow an adversary to control 32 IP addresses to monopolize all connections to and from a target bitcoin node with 85% probability



Eclipse attacks [HKZG15]

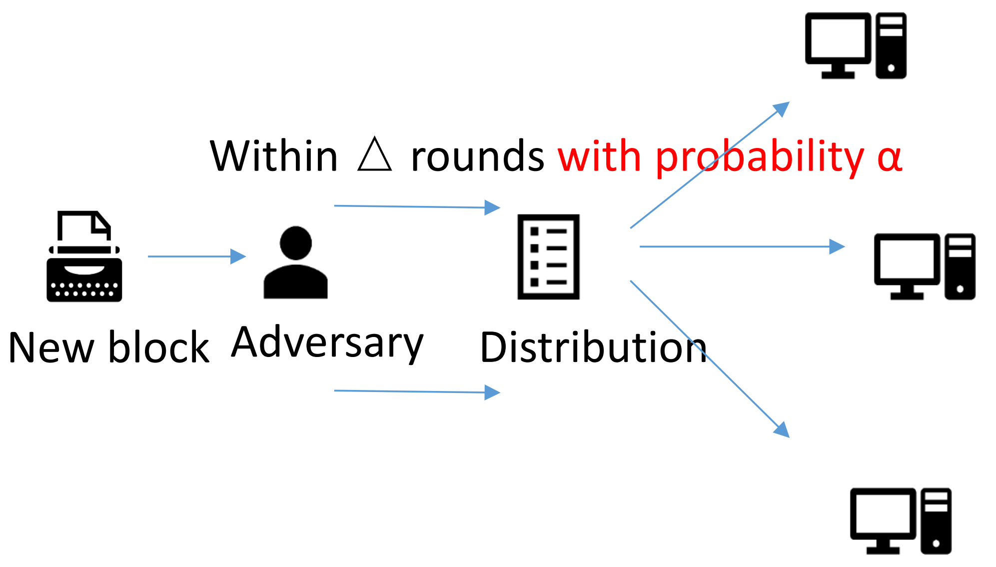
Blockchain protocol with delays

·

Is the blockchain protocol based on POW still secure in the asynchronous network, where long delay, say $\Delta \geq 1/np$, is allowed?

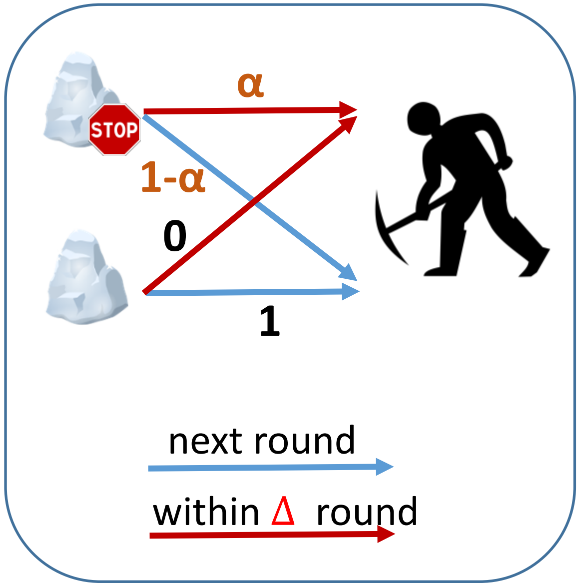
Our contribution

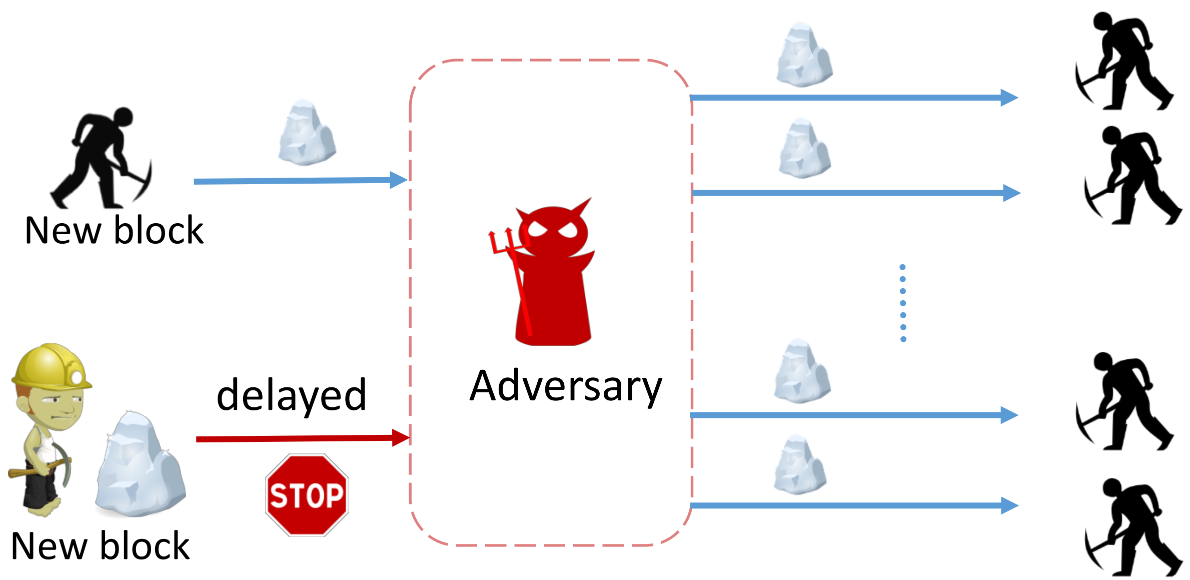
- Focus on the effect of long delay, especially $\Delta \geq 1/np$
  - Prove that the common prefix property and the chain growth property can still hold in our model when considering long delay
    - define chain growth and common prefix in a more subtle way
    - simplified proof method for POW based blockchain



Our blockchain model

- The adversary A
  - Deliver all messages sent by miners
  - Delay the target chains with probability α
    - Within Δ rounds
  - Do not have any hash power





Our blockchain model

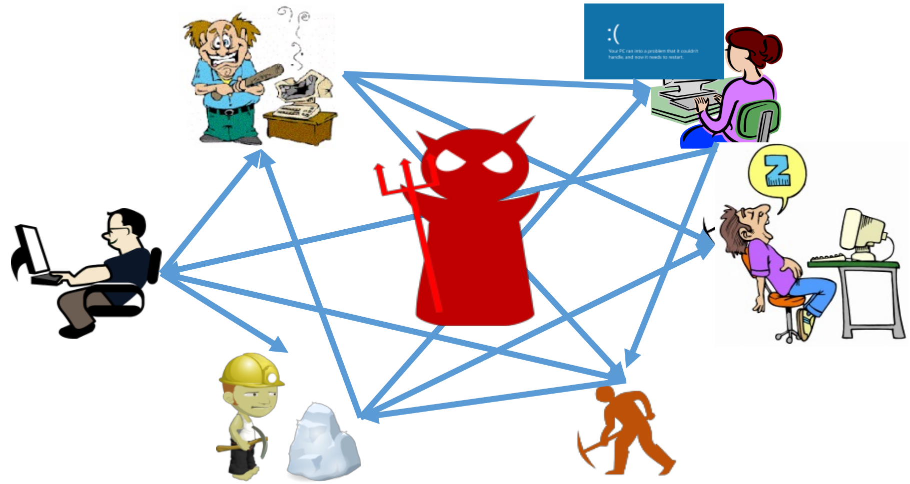
- Modification to blockchain protocol
  - Consecutive blocks cannot be mined by the same miner (not the same mining pool)
    - a single miner
      - an independent communication node of the network
      - has a unit computational power
  - May lead to possible forks
  - In practice It is unlikely that a miner can mine two consecutive blocks
    - large number of miners n
    - small difficulty parameter p

Our blockchain model

Too weak?

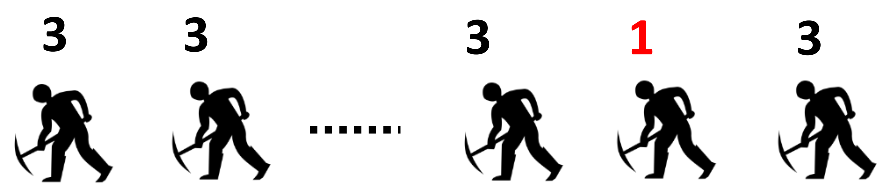
- Honest miners setting
  - The adversary does not corrupt any miners (No hash power)
  - Our model captures a class of practical attacks in the real world
- For the adversary in a large-scaled blockchain protocol
  - More difficult to control a sizable fraction of hashing power
  - Much easier to disrupt communications among miners
  - Present a concrete attack in which an adversary without any hash power may threaten the common prefix property

Security requirements



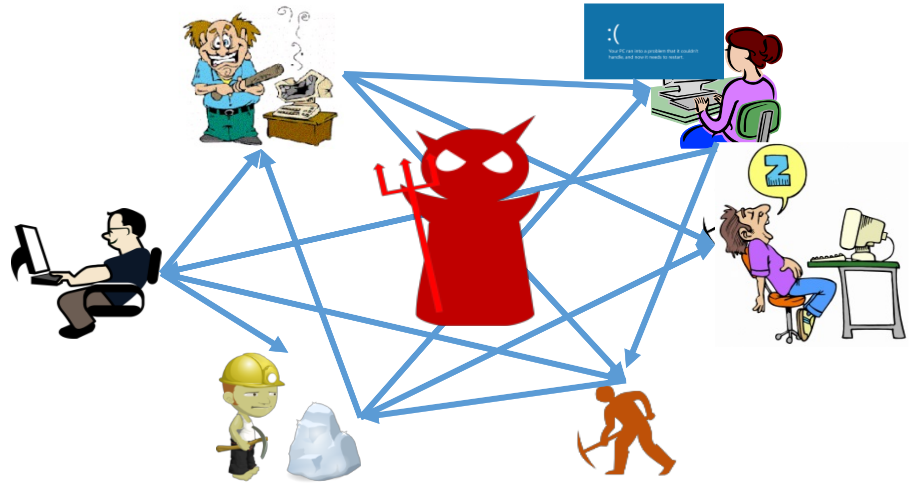
Chain Growth

- Previous work: the minimum length increase of **all honest miners**' chains during T rounds

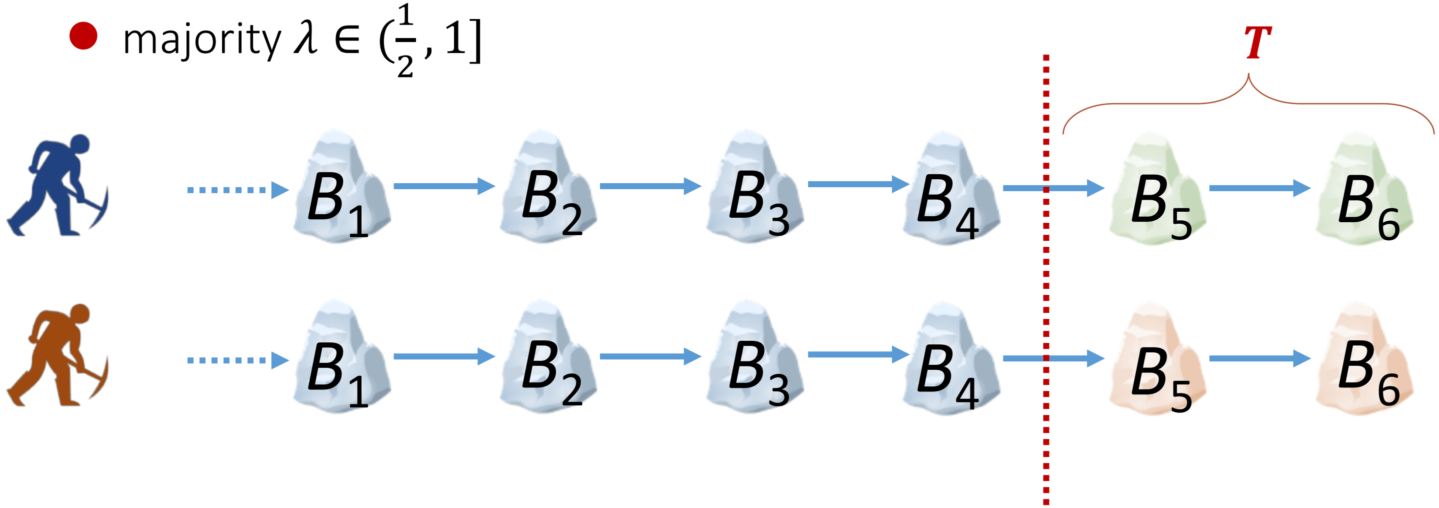


- Our work: the length increase of the **majority of honest miners**' chains
  - majority $\lambda \in (\frac{1}{2}, 1]$
  - Exclude the "bad" honest minority
  - Chain growth in [PSS17] is a special case of ours when $\lambda = 1$

Security requirements



- **Common Prefix**
  - Previous work: **All** the honest miners have the **same** history (prefix)
  - Our work: **The majority** of the honest miners have the **same** history
    - Allow **some** miners' chains to be **inconsistent** with the main chain
    - majority $\lambda \in (\frac{1}{2}, 1]$

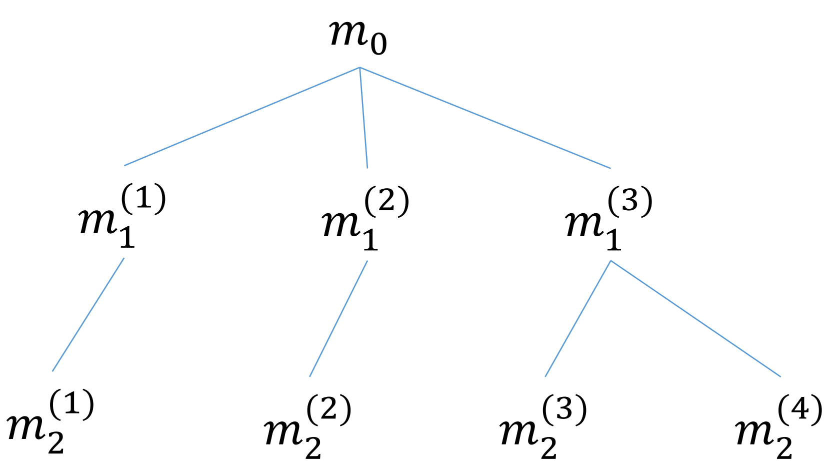


Security proof

- How to capture the evolution of the main chains?

State of the Main Chain

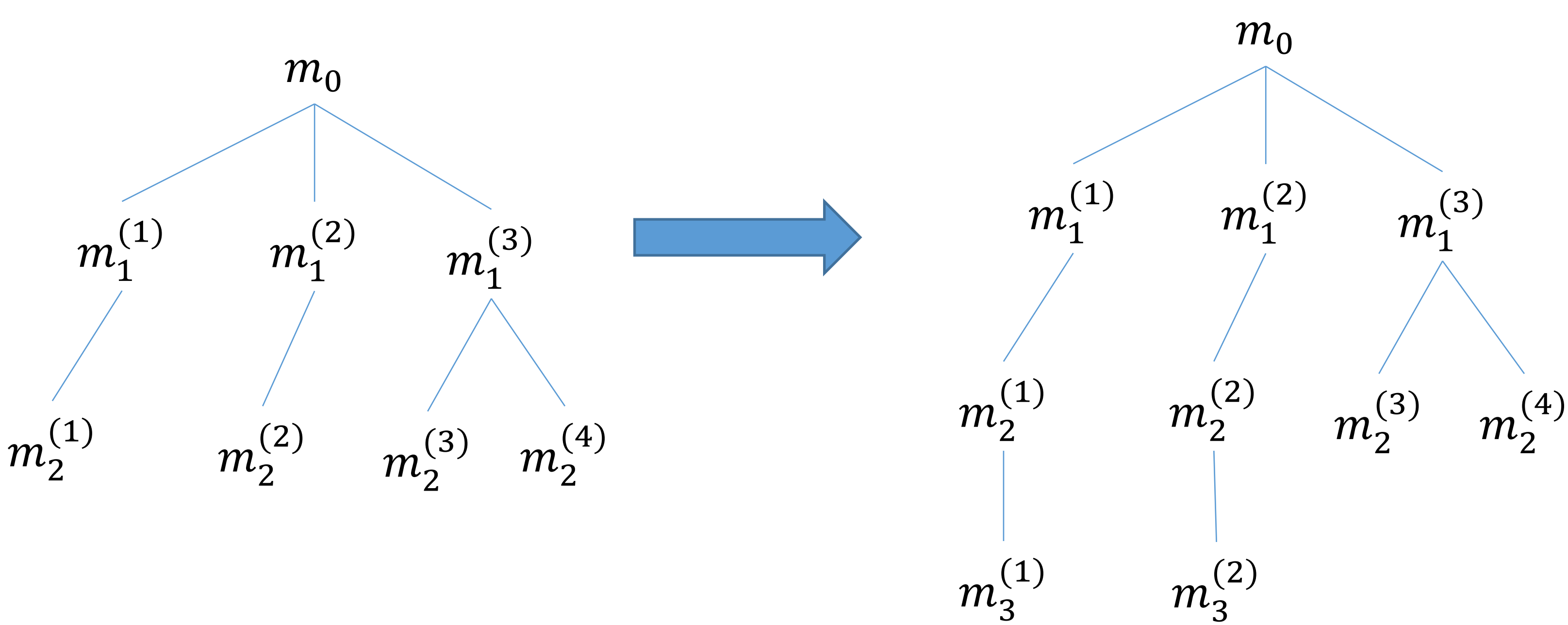
- Tree$_{MC}$ to capture the evolution of the main chains
  - Inspired by **F**$_{tree}$ model [PSS17], record all the branches (or forks)
  - Tree$_{MC}$ in our model
    - Only store the current state of the main chains
    - Delayed chains are not recorded in Tree$_{MC}$
    - Basic operations: AddBlock, DeleteBlock



$$C_1 = (m_0, m_1^{(1)}, m_1^{(1)})$$
$$C_2 = (m_0, m_1^{(2)}, m_2^{(2)})$$
$$C_3 = (m_0, m_1^{(3)}, m_2^{(3)})$$
$$C_4 = (m_0, m_1^{(3)}, m_2^{(4)})$$

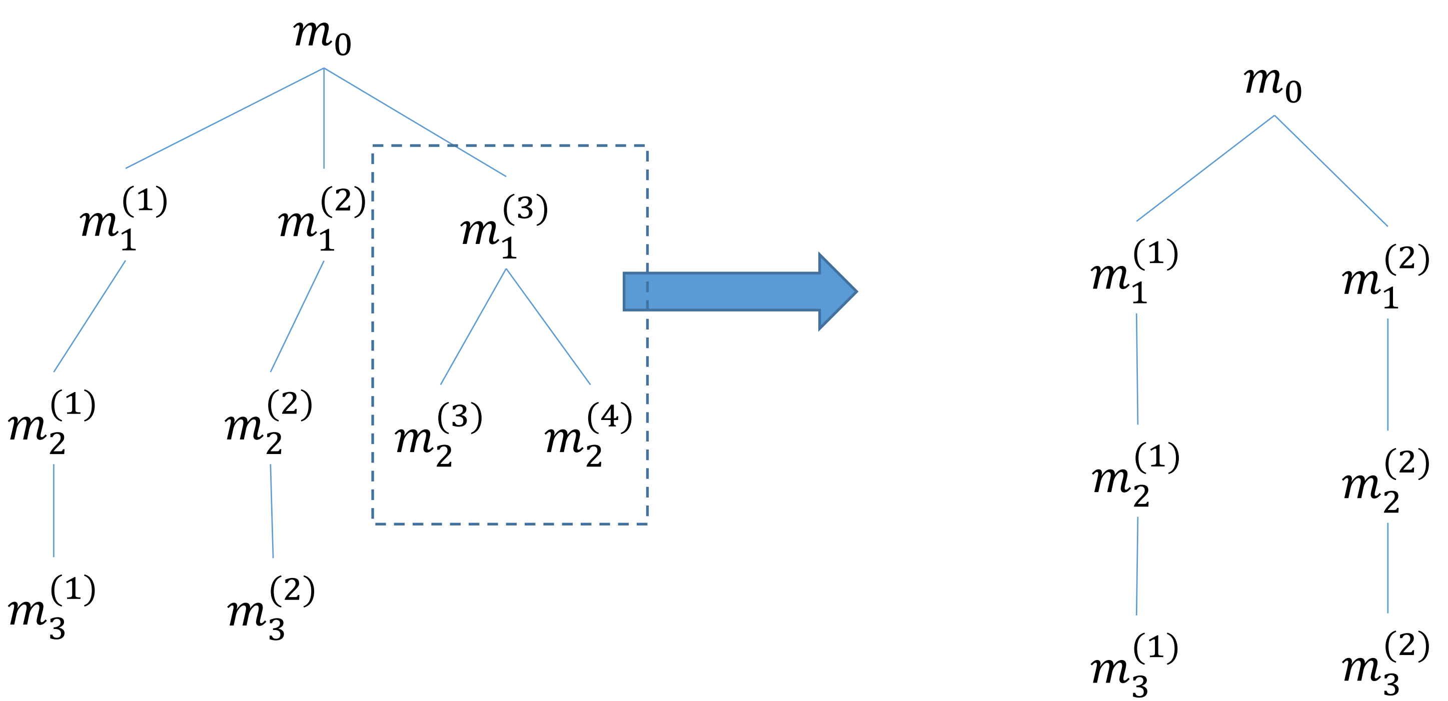
State of the Main Chain

- AddBlock:
  - When the adversary broadcasts $C_1 = (m_0, m_1^{(1)}, m_2^{(1)}, m_3^{(1)})$ and $C_2 = (m_0, m_1^{(2)}, m_2^{(2)}, m_3^{(2)})$



State of the Main Chain

- DeleteBlock:
  - Remove the useless nodes



Difference between Tree_{MC} and the miners' view

- Each miner has their own view of the main chain, which may be different with Tree_{MC}
- In terms of chain growth and common prefix , the difference is negligible
  - Reduced to the security of Tree_{MC}
  - Simple proof for Tree_{MC}
    - Useful properties on the depth of Tree_{MC}

Lemma 1. *Properties of Tree_{MC}.*

1. *If new blocks are successfully added to Tree_{MC} at the end of a round, then the depth of Tree_{MC} increases.*
2. *The depth of Tree_{MC} increases by at most 1 at each round.*
3. *If only one block is added to Tree_{MC} at the end of a round, then Tree_{MC} has only one branch and the depth increases by 1.*

Chain Growth

Theorem 1 *(Chain growth). Assume* $1/2 < \lambda \leq 1 - 8\alpha p\Delta$*. The blockchain protocol* $(\Pi, \mathcal{C})$ *has the chain growth rate* $g = \frac{(1-\delta)f}{1+fE[R^i_{delay}]}$ *with majority* λ*, where* $f = 1-(1-p)^n$, $E[R^i_{delay}] = \frac{\alpha - \alpha\omega^{\Delta-1}[\omega+\Delta(1-\omega^2)]}{1-\omega}$ *and* $\omega = 1-(1-\alpha)f$*.*

Main idea of proof

$$g > \frac{t}{R_{mine} + R_{delay}}$$

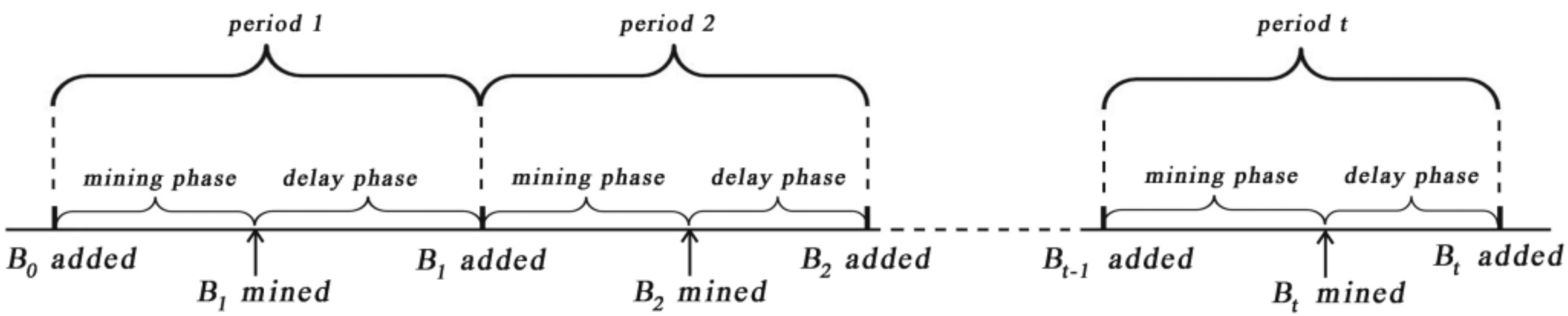


Fig. 1. The rounds during which t consecutive blocks are added to Tree$_{MC}$

Common Prefix

Theorem 2 *(Common prefix). Assume* $0 < \alpha < 1 - np$ *and* $1/2 < \lambda \leq 1 - 8\alpha p\Delta$*. The blockchain protocol* $(\Pi, \mathcal{C})$ *satisfies the common prefix property with parameter* λ*.*

Main idea of proof

The event **converge**

- Only one miner succeeds in mining at round $r*$.
- $C*$ is delayable while there is no new block mined in following Δ rounds
  OR The chain $C*$ is undelayable

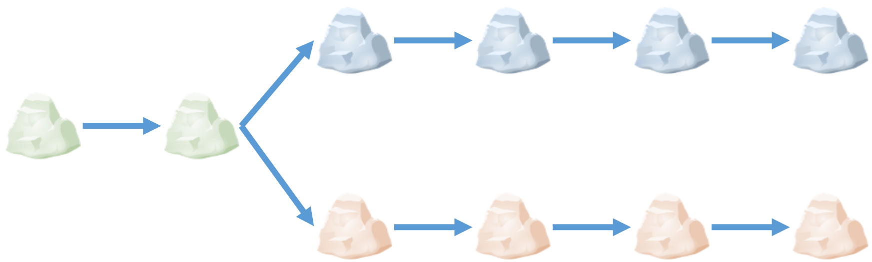
$$\Pr[\textbf{converge}] > 1 - np(1 + \alpha\Delta)$$

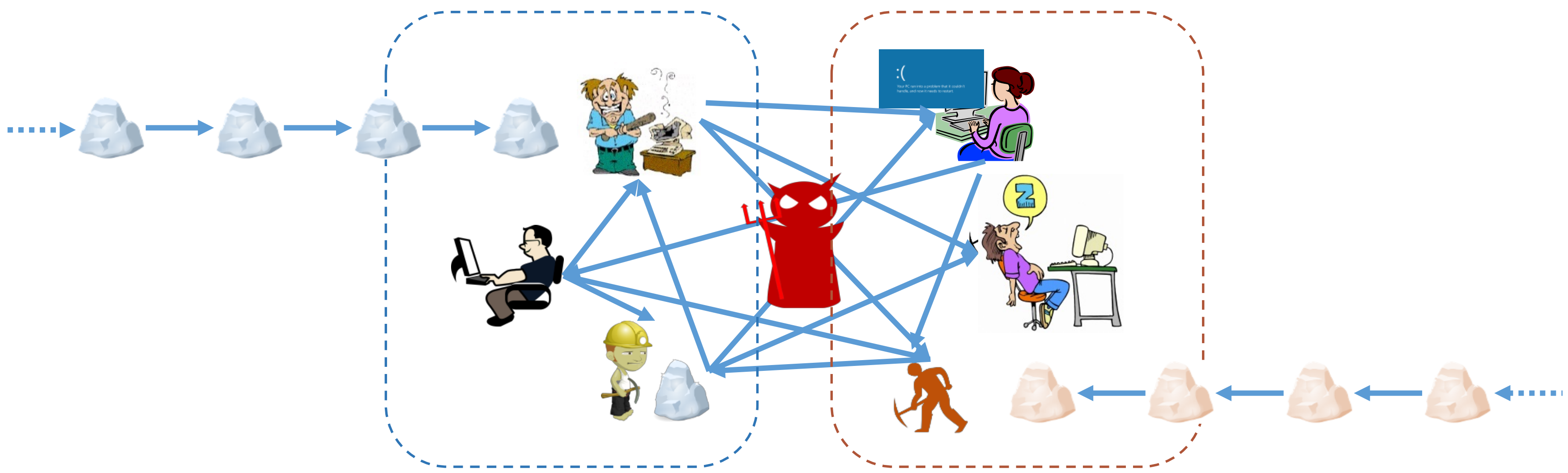
For Tree_{MC} with common prefix of depth d-T

$$1 - \left(np(1 + \alpha\Delta)\right)^T$$

Long Delay Attack on Common Prefix

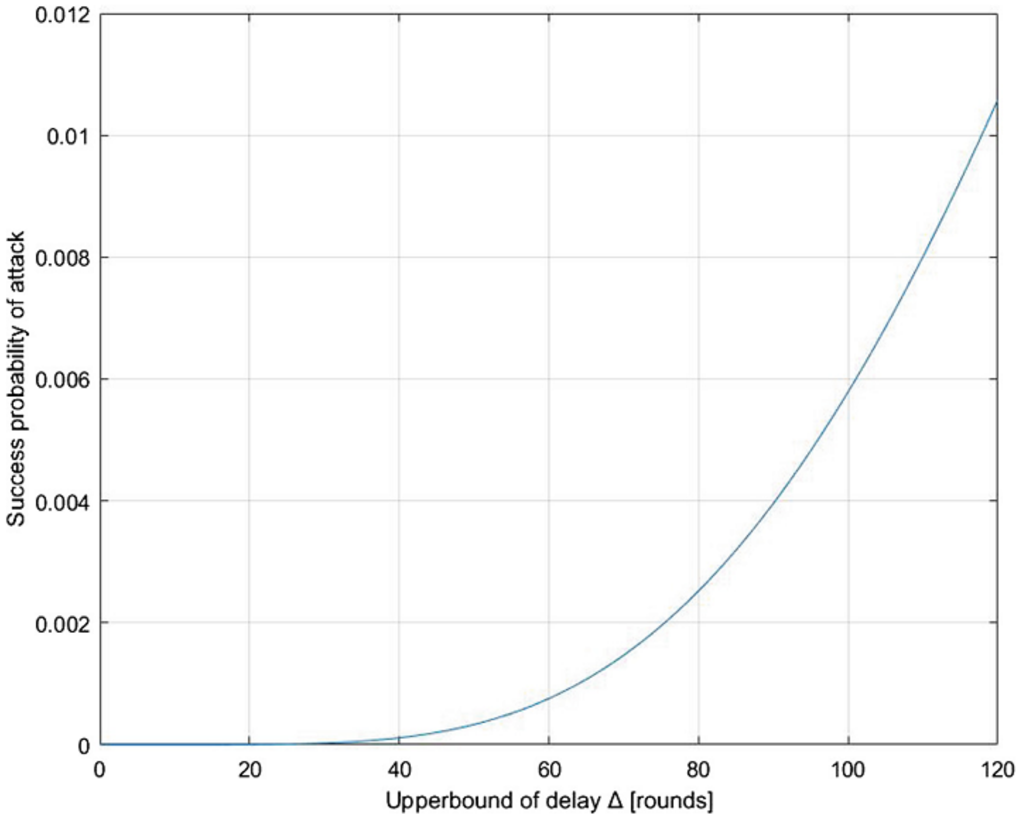
- Concrete attack on the common prefix of Tree_{MC}
  - when Δ and α are "too" large relative to a fixed T
  - Goal of attack: increase the length of the two branches by T





Long Delay Attack on Common Prefix

- With inappropriate parameters, adversaries without any hash power can threaten the common prefix property
  - For $\alpha = 0.8$ and $T = 6$, the success probability increases as Δ gets larger.



the success probability grows much faster when $\Delta > 60$ (10 min). When $\Delta > 120$ (20 min), the success probability can reach about 1%.

Future work

- Stronger security model
  - Convert honest miner setting to regular miner setting
- Robustness of blockchain for data storage
  - Provide reliable storage with provable robustness

School of Cyber Security

Shandong University, Qingdao

Welcome to visit! & Welcome to join us!

pwei@sdu.edu.cn

Thanks!
&
Questions?