

CS 598:  
AI Methods for Market Design

Lecture 11: Cryptoeconomics (Bitcoin)

Xintong Wang  
Spring 2024

# Bitcoin Transactions

- Enable digital payments between untrusted parties...  
*with no central authority* (no banks or governments)
- A Bitcoin transaction includes
  - Sender(s)
  - Receiver(s)
  - Amount to transfer (in BTC)
  - A proof of ownership (pointer to last transaction with these coins)
  - Transaction fee

# Bitcoin Transactions

- Enable digital payments between untrusted parties...  
*with no central authority* (no banks or governments)
- A Bitcoin transaction includes
  - Sender(s) *Cryptographically signed by sender*
  - Receiver(s)
  - Amount to transfer (in BTC)
  - A proof of ownership (pointer to last transaction with these coins)
  - Transaction fee *Transactions are authorized in **a ledger** and broadcasted (P2P network)*

# How Are Transactions Added to Ledger?

- Ledger: history of all transactions authorized that are grouped in “blocks”
- A block includes
  - Some transactions (~1000-2000)
  - A reference (hash) to the preceding block
  - A “nonce” (a bunch of bits)

- A blockchain 

# The Blockchain

*How / Who add new blocks to the blockchain? How to make sure that everyone agrees on the content?*

- Incentivize *miners* to add blocks by monetary rewards (how BTCs gets “*minted*”) 6.25BTC currently
- Make it hard by solving a computationally difficult puzzle (“*proof of work*”)



# Mining

- The process of finding new *valid* blocks
- The *intended* mining behavior includes:
  - Choose a subset of outstanding transactions (e.g., those with higher transaction fees)
  - Try to find a valid block by setting the bits in the nonce
- Append to the current last block of the blockchain

# Mining

- The process of finding new *valid* blocks
- The *intended* mining behavior includes:
  - Choose a subset of outstanding transactions (e.g., those with higher transaction fees)
  - Try to find a valid block by setting the bits in the nonce  
bits in nonce → SHA-256 → output (256 bits)

Solve the *cryptographic hash function* (a random function) s.t. the leading  $l$  bits are 0;  $l$  is set to control the rate

$l = 80$ : on average, succeed every  $2^{80}$  attempts

- Append to the current last block of the blockchain

# Forks

- Issue: Two different valid blocks are discovered at roughly the same time → a fork

b1 ← b2 ← b3 ← ... ← b6

↙ b4 (“orphaned” no rewards)



# Forks

- Issue: Two different valid blocks are discovered at roughly the same time → a fork

b1 ← b2 ← b3 ← ... ← b6

↙ b4 (“orphaned” no rewards)

- Specified behavior: Interpret authorized transactions as those in the longest chain (break ties in favor of the block you heard the first)

# Forks

- Issue: Two different valid blocks are discovered at roughly the same time → a fork

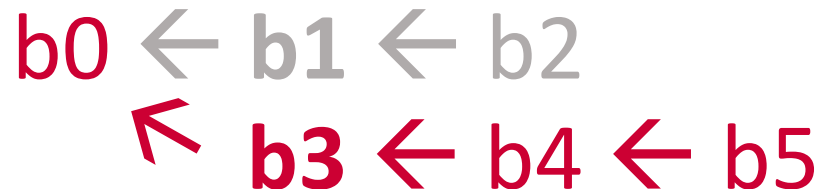
b1 ← b2 ← b3 ← ... ← b6

↙ b4 (“orphaned” no rewards)

- Specified behavior: Interpret authorized transactions as those in the longest chain (break ties in favor of the block you heard the first)
- Consequence: Consider a transfer of funds as complete only after transactions added to blockchain *and* extended by several more blocks

# Incentives: Forking Attacks

- Double-spend attack: deliberately create forks
  - Alice pays Bob in block b1
  - Block b2 is added after b1
  - Alice tries to orphan b1 and b2 by extending b0 with three blocks before anyone extends b2
- $\alpha$ : the fraction of computational power possessed
  - Alice's success probability:  $\alpha^3$  or  $\alpha^{k+2}$  if Bob waits for k blocks to be added



# Incentives: Forking Attacks

- 51% Attack: If  $\alpha > 0.5$ , the miner can act like a centralized authority (govern the longest chain)

# Incentives: Selfish Mining

- Selfish mining: the behavior of *block withholding* (don't tell other miners about your eligible block)
- Strategy:
  - Alice finds eligible block s1
  - Alice tries to privately extend s1 with another block s2
  - If b4 is announced first, Alice needs to restart
  - If s2 is found first, Alice mines secret chain until her "lead" drops to 1



# Incentives: Selfish Mining

- Selfish mining: the behavior of *block withholding* (don't tell other miners about your eligible block)
- Strategy:  $\alpha > \frac{1}{3}$ : selfish mining better than honest mining (Eyal & Sirer, 2014)
  - Alice finds eligible block s1
  - Alice tries to privately extend s1 with another block s2
  - If b4 is announced first, Alice needs to restart
  - If s2 is found first, Alice mines secret chain until her "lead" drops to 1

