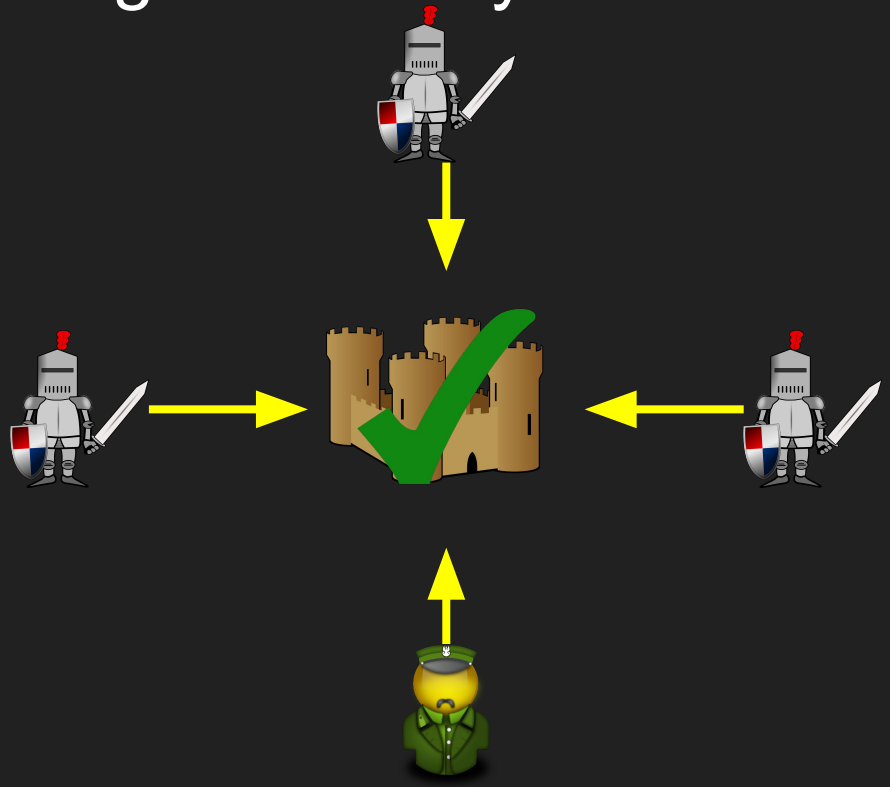
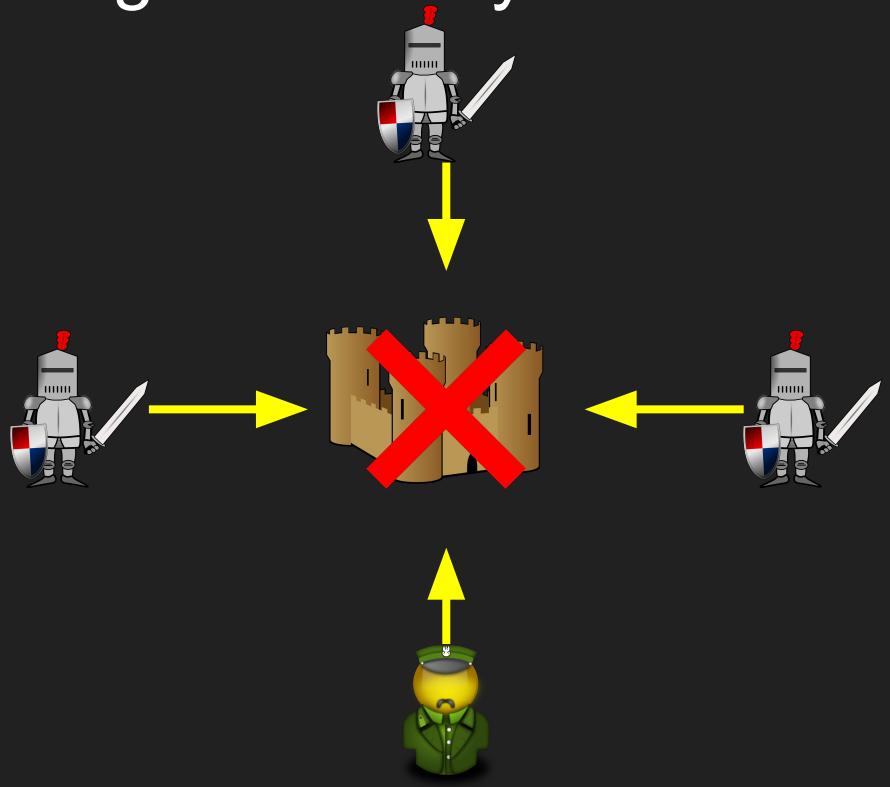
Consensus Algorithms

Consensus Algorithms: Byzantine Generals



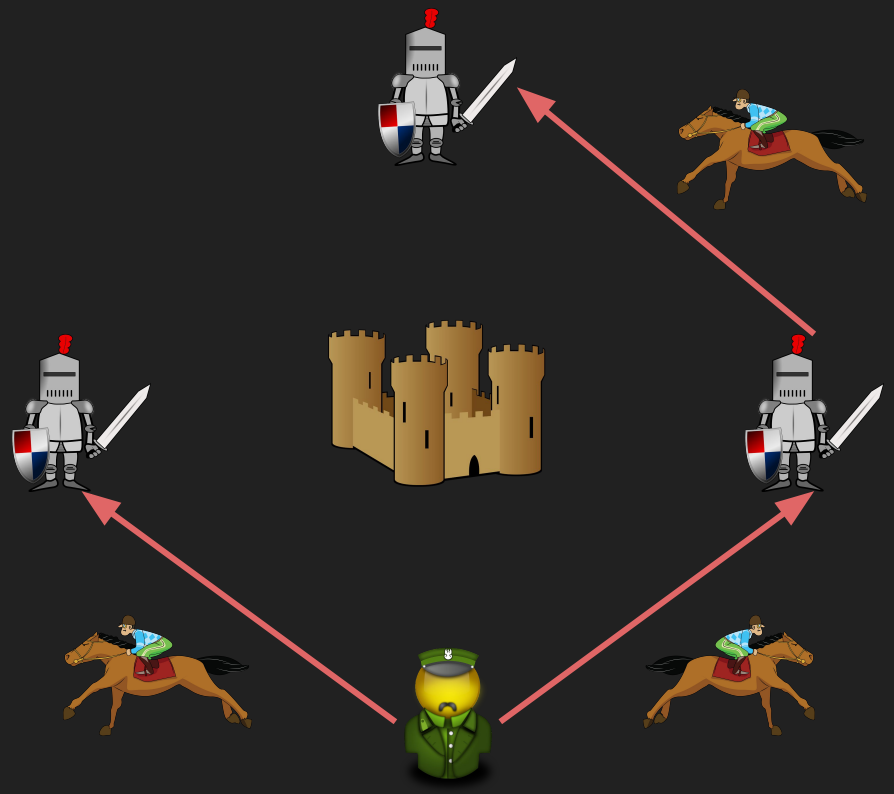
Consensus Algorithms: Byzantine Generals



Consensus Algorithms: Byzantine Generals

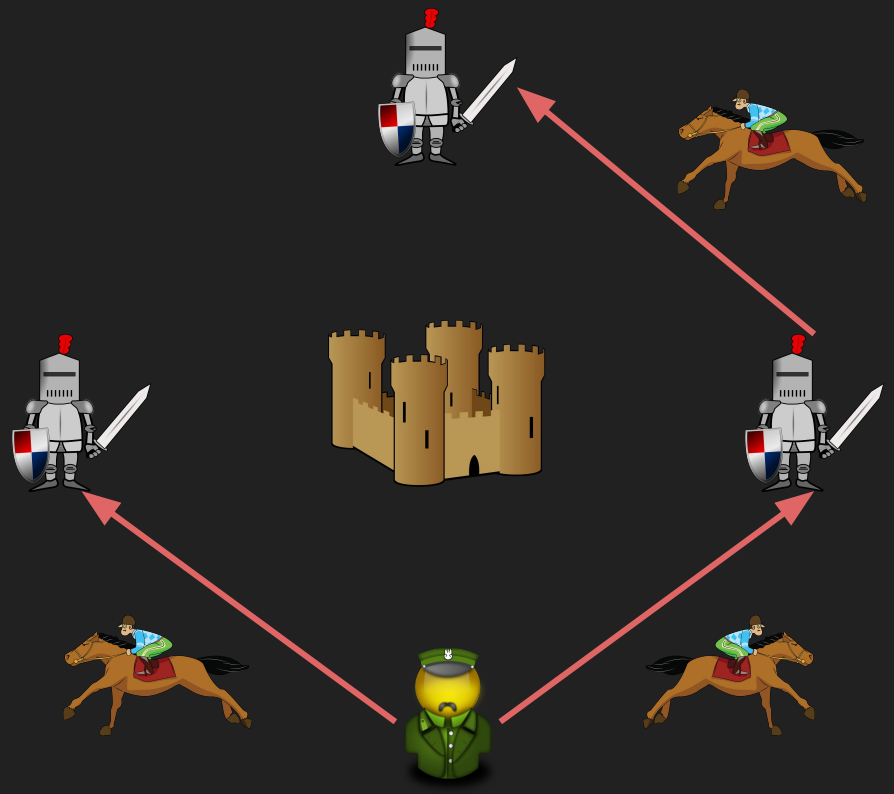
- Key Question: How do we coordinate with all the other generals at once?

- Assume we can’t send signals the enemy can see (like torches)
- We’re going to have to send messengers



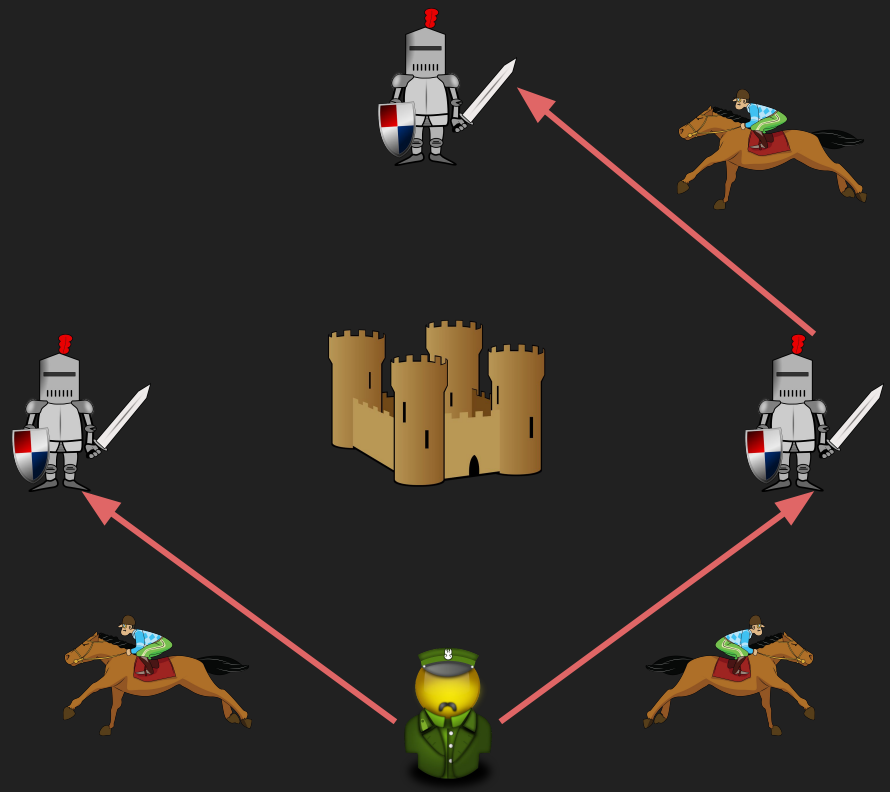
Consensus Algorithms: Byzantine Generals

- What issues might we have?
  - How do we know the messengers made it?
  - How do we know that the message wasn't intercepted and replaced?
    - Same for the response
  - How do we know that the generals will even go along with the plan?



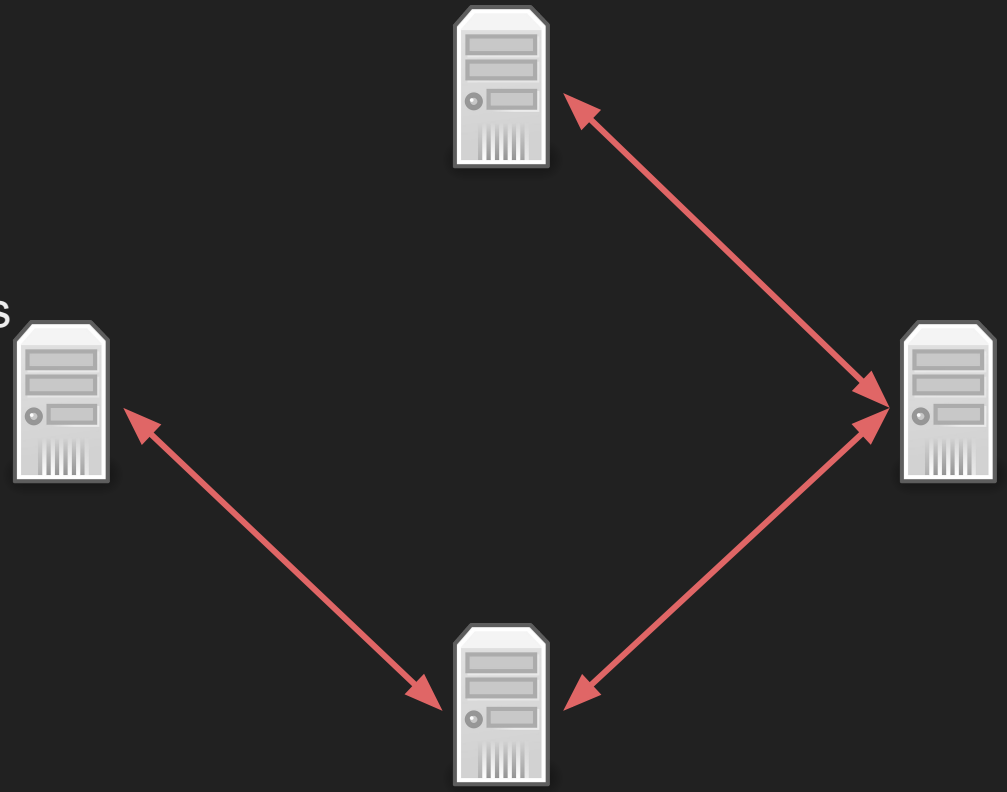
Consensus Algorithms: Byzantine Generals

- How does this relate to Computer Science?
  - Replace generals with computers and messengers with network packets



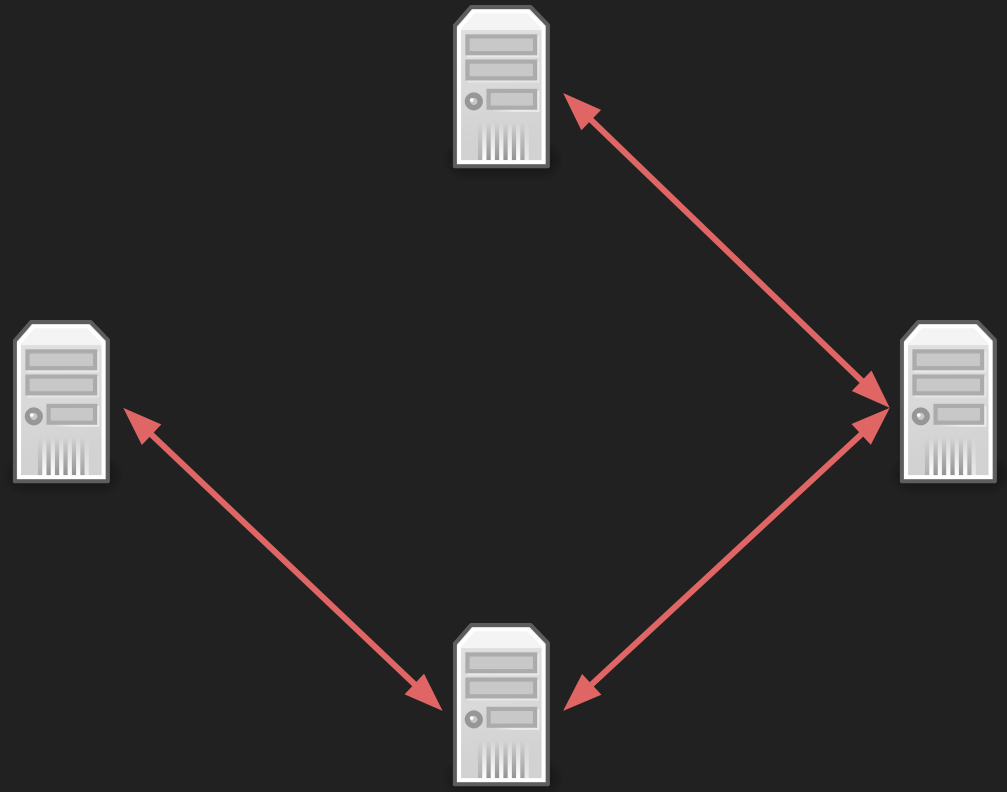
Consensus Algorithms: Byzantine Generals

- How does this relate to Computer Science?
  - Replace generals with computers and messengers with network packets



Consensus Algorithms: Byzantine Generals

- What issues might we have?
  - How do we know the packets made it?
    - Networking (TCP)
  - How do we know that the packet wasn't intercepted and replaced?
    - Encryption, Digital Signatures, etc.
  - How do we know that the other computers aren't working against us?



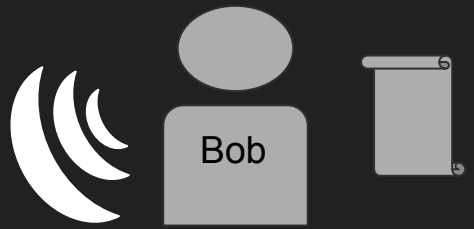
Proof of Work

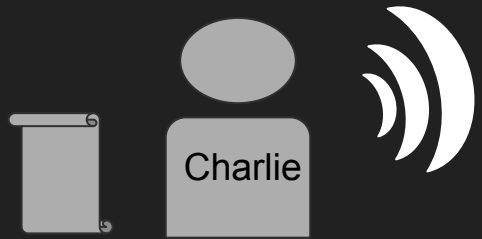
Proof of Work: Cryptographic Hashing

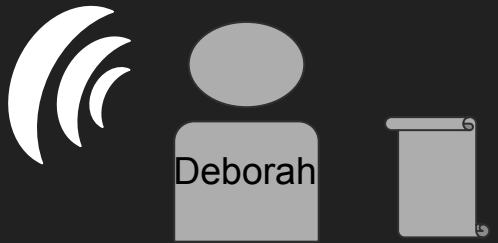
Requirements for a good cryptographic hash:

- Deterministic (i.e. a given input will always result in the same output)
- Computationally infeasible to generate the input from the output
- Small changes to input should result in big (random-looking) changes to output (the outputs should not appear correlated in any way)
- Computationally infeasible to create two inputs with the same output
- Ideally, fast to compute

Proof of Work





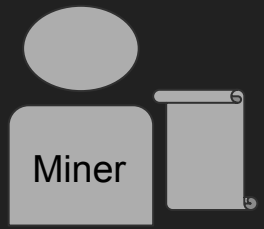


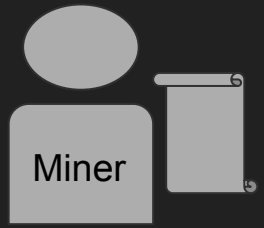
Proof of Work

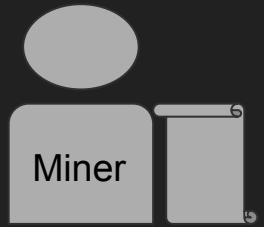
- BIG IDEA: trust the ledger that has had the most “work” into it.

- What do we mean by work? Hashing!

Proof of Work

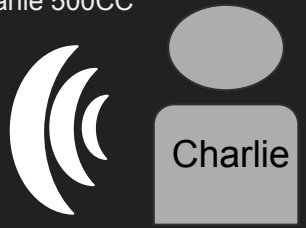




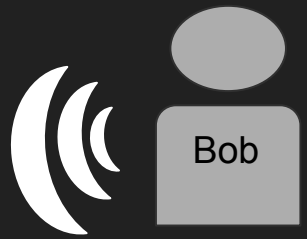


Alice owes Bob 100CC

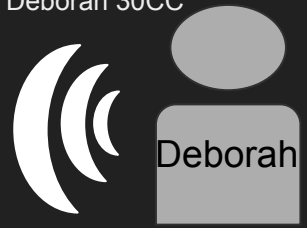
Alice owes Charlie 500CC



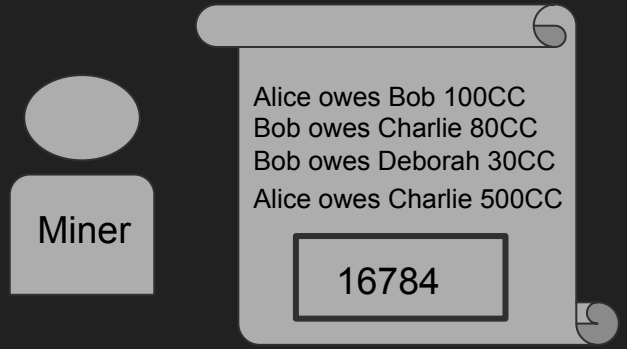
Bob owes Charlie 80CC



Bob owes Deborah 30CC



Proof of Work

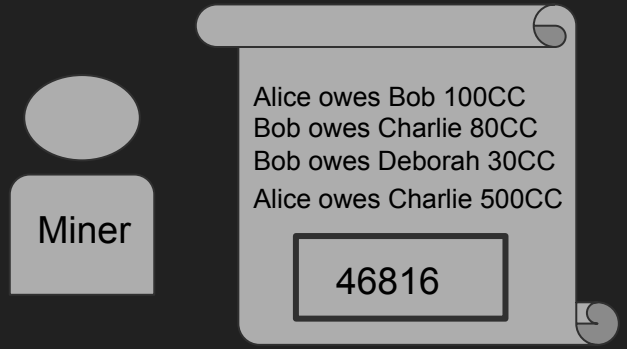


SHA-256

N = 10

00011101010011100010100001010010010010000101001010000111111111100110….

Proof of Work

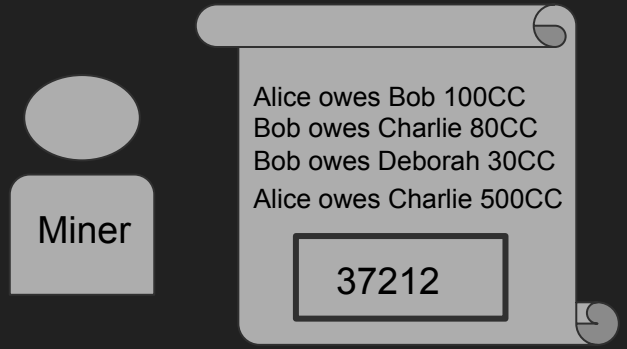


SHA-256

N = 10

01110100111111111111111
00101000000101011010100
0000101001010….

Proof of Work

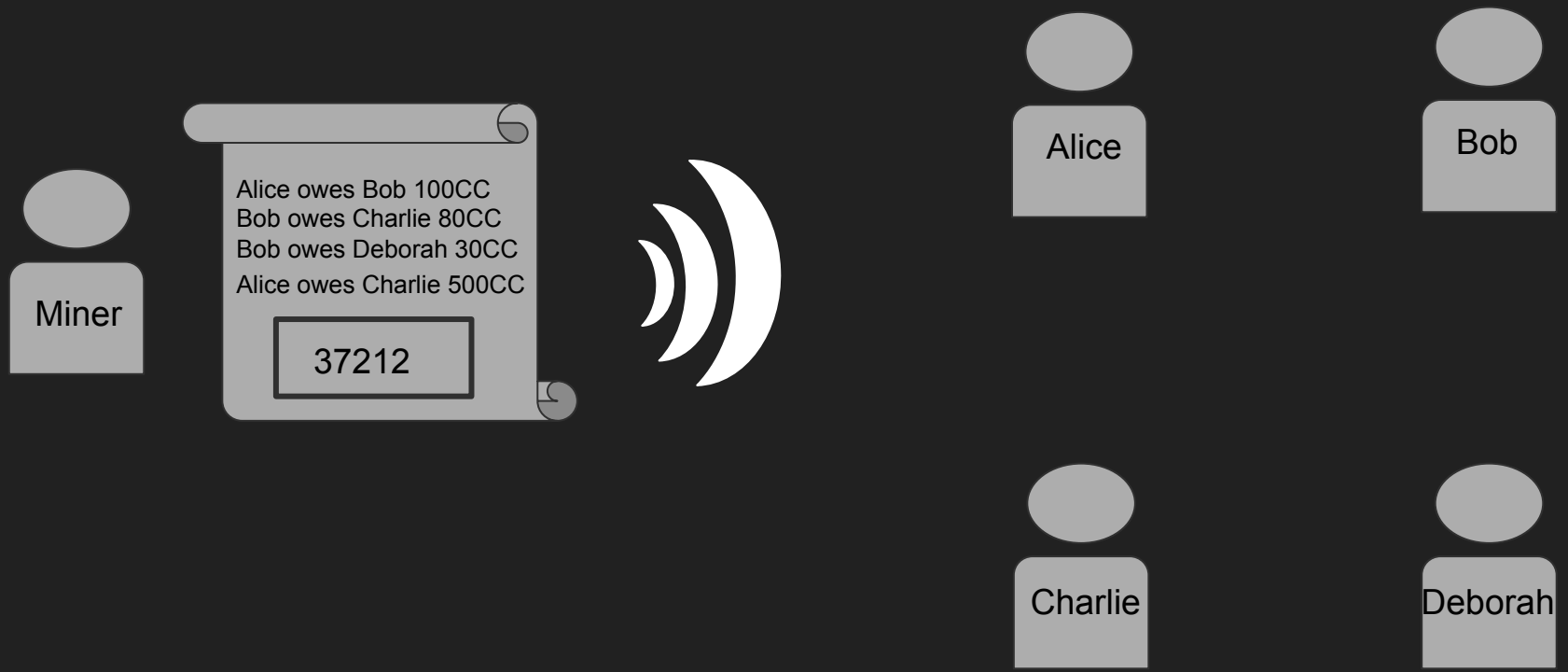


SHA-256

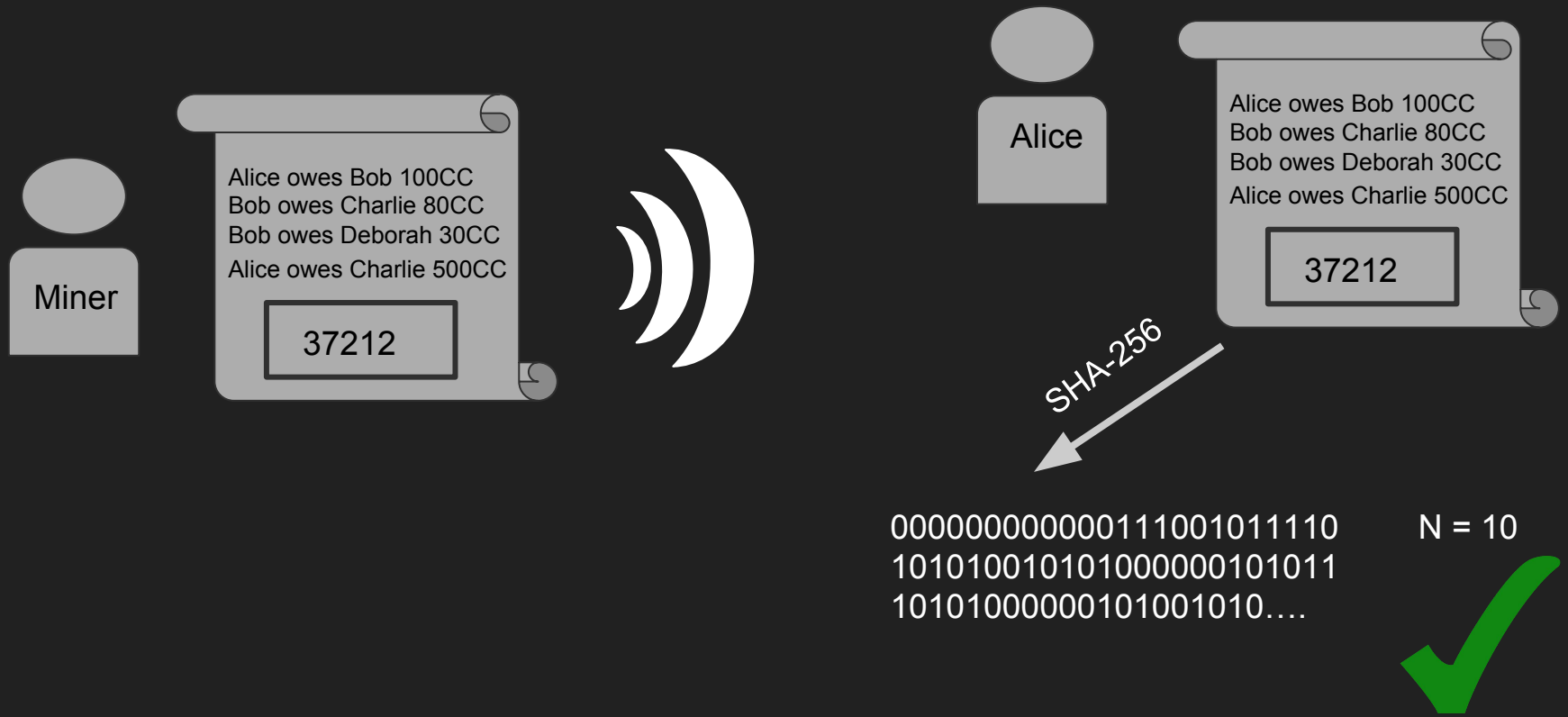
N = 10

00000000000011100101111010
10101001010100000010101110
10101000001010010 10….

Proof of Work

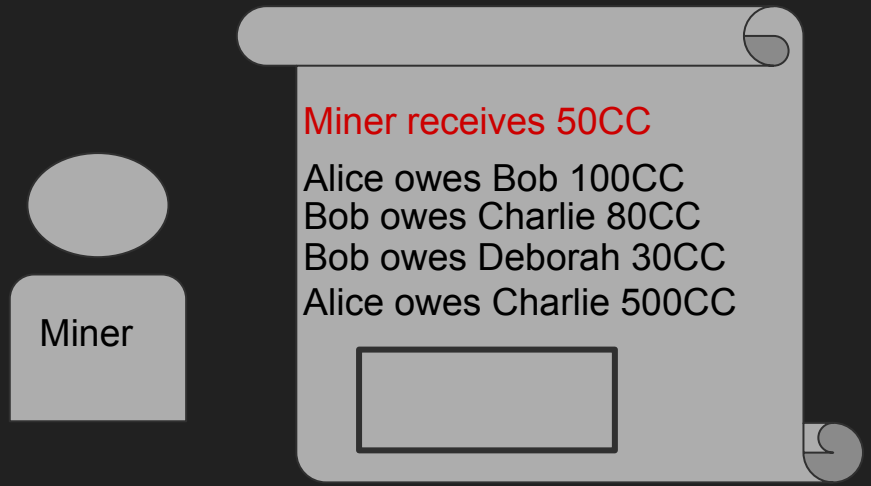


Proof of Work



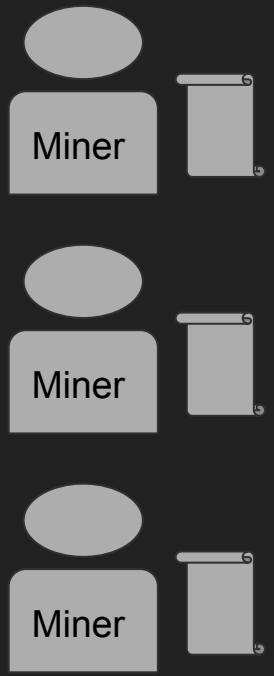
Proof of Work

- What's in it for the Miner?



Proof of Work

- How can Alice make sure her transaction makes it in?

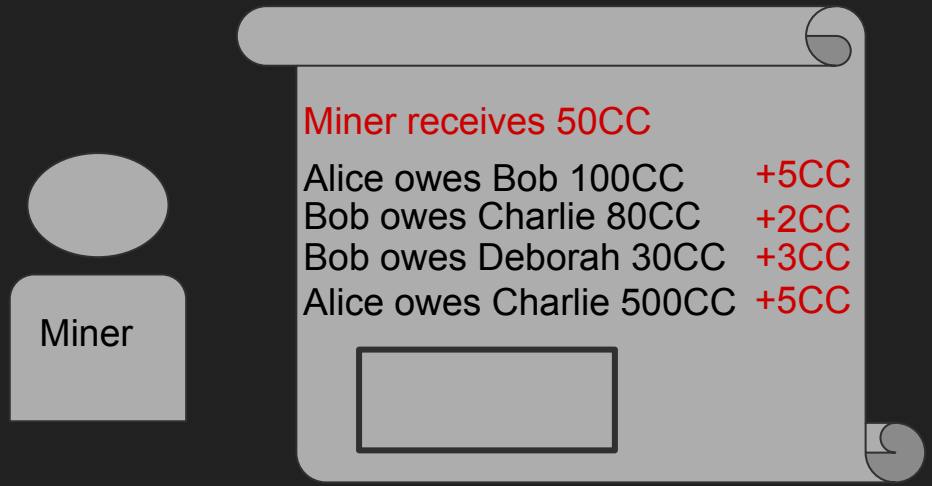


Alice owes Bob 100CC

Alice tips Miner 5CC

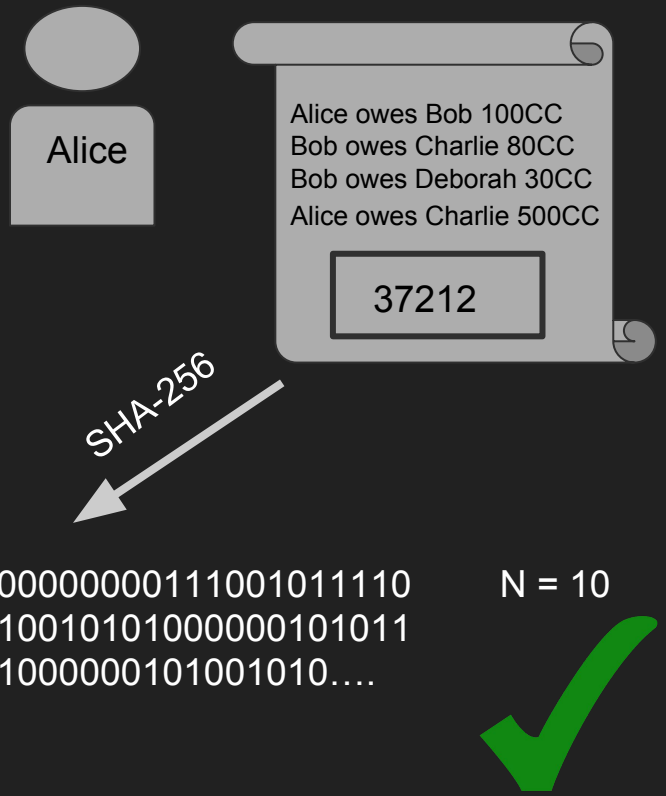
Proof of Work

- How can Alice make sure her transaction makes it in?

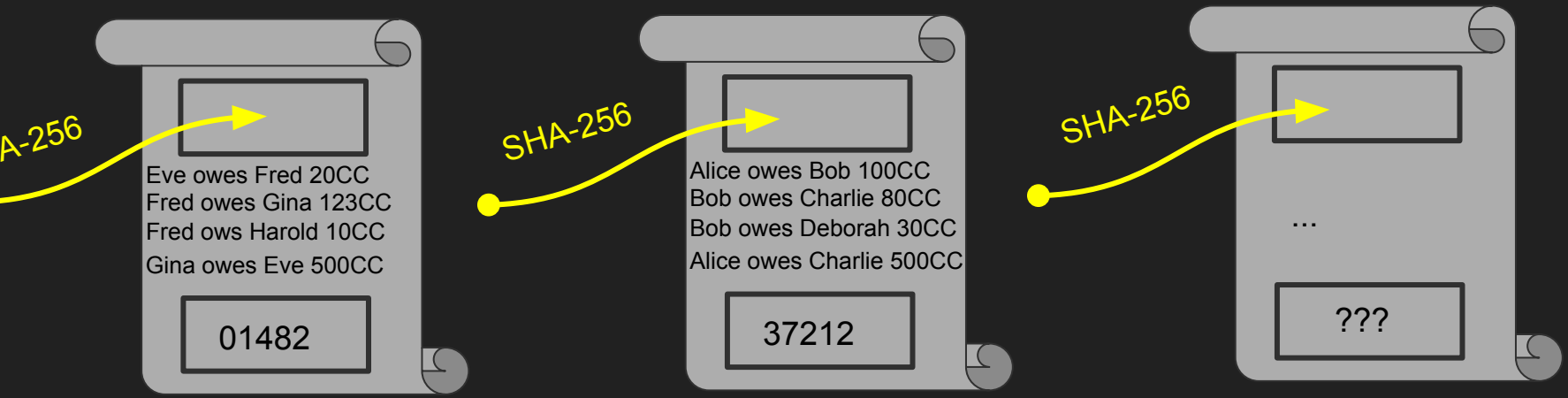


Proof of Work

- How does Alice know she hasn't missed anything?

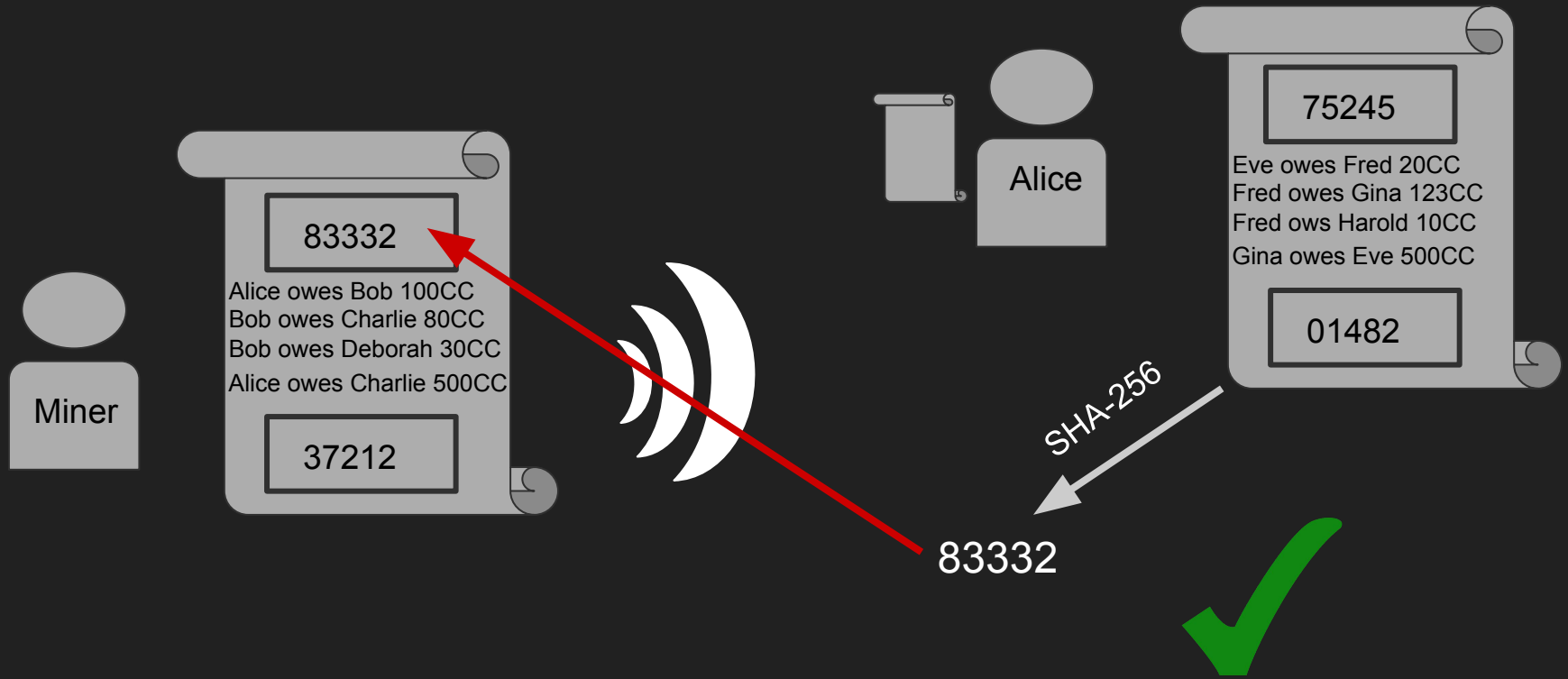


Proof of Work

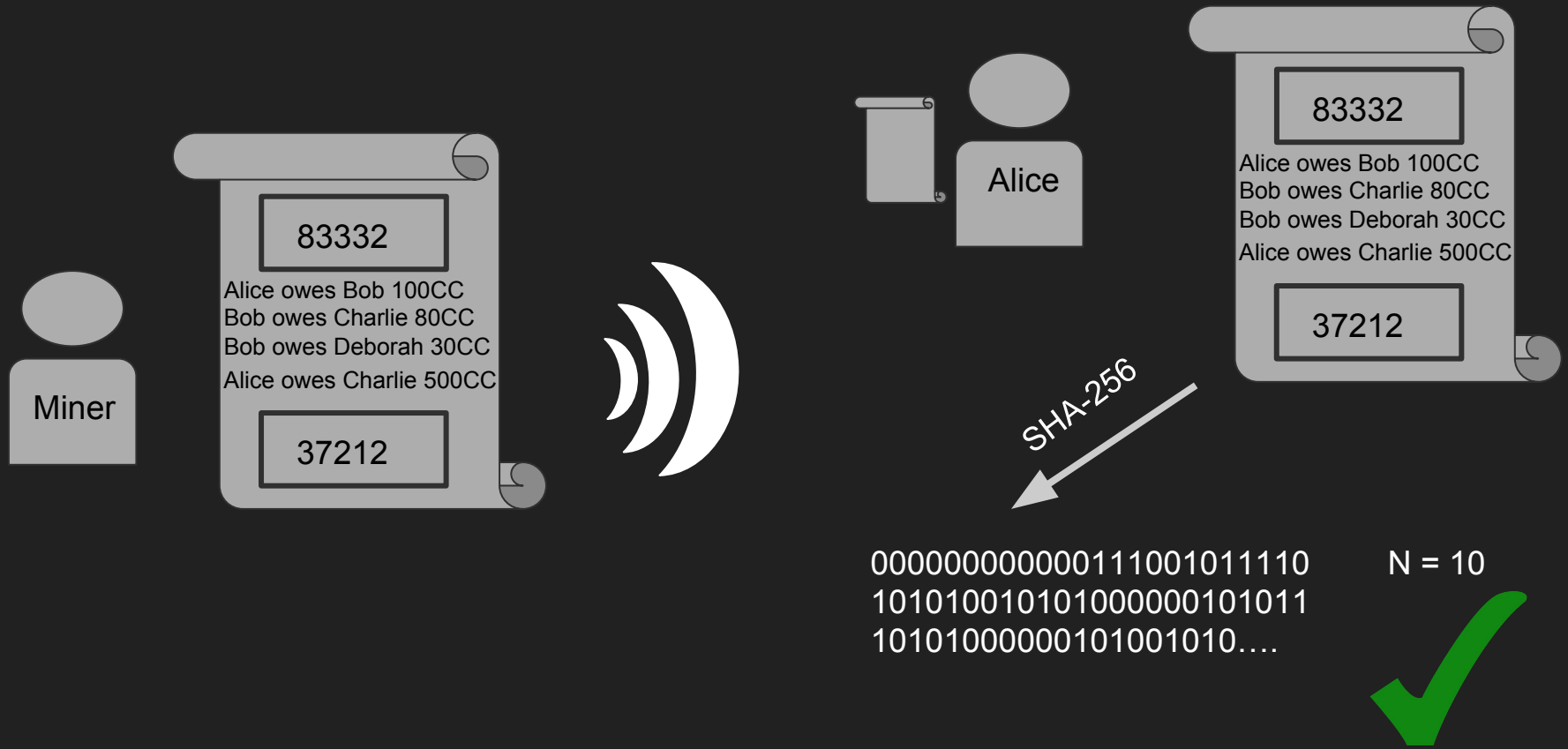


BLOCKCHAIN

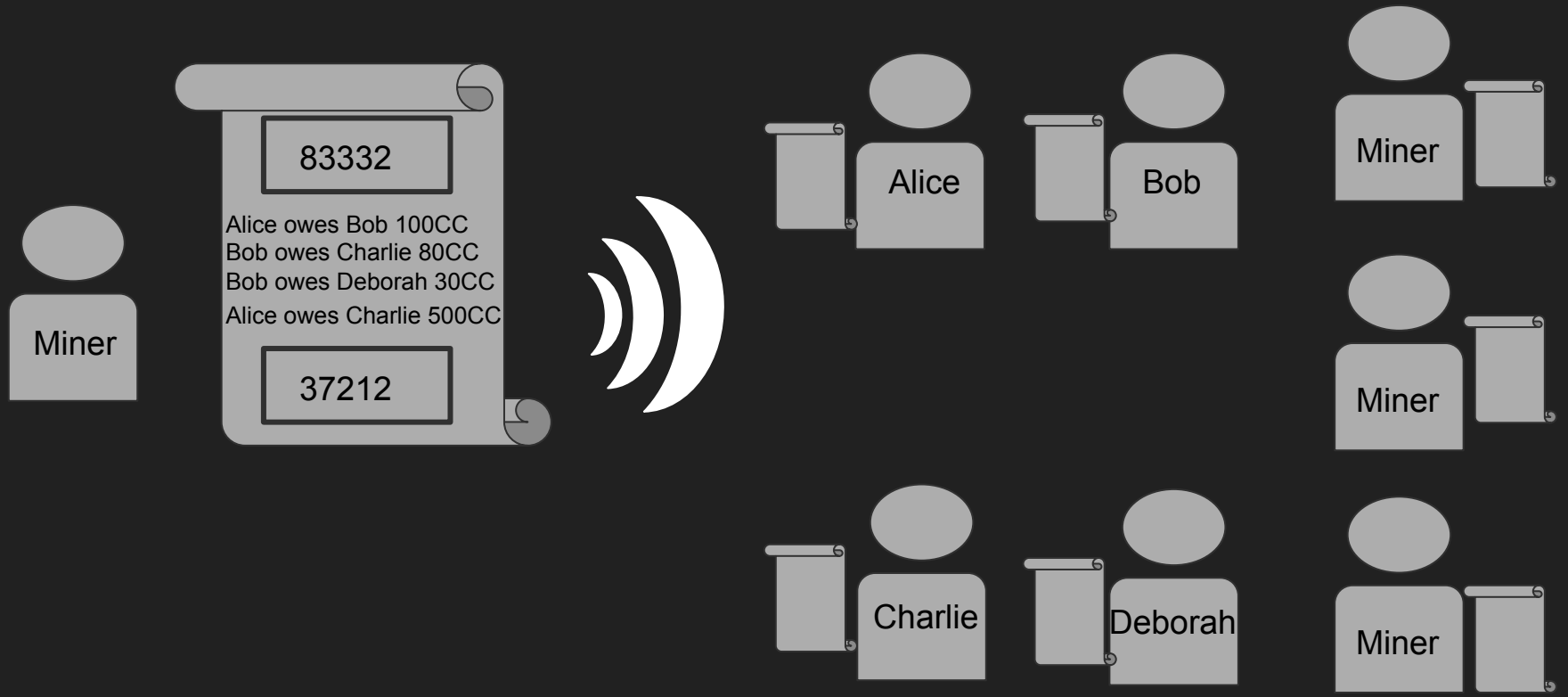
Proof of Work



Proof of Work

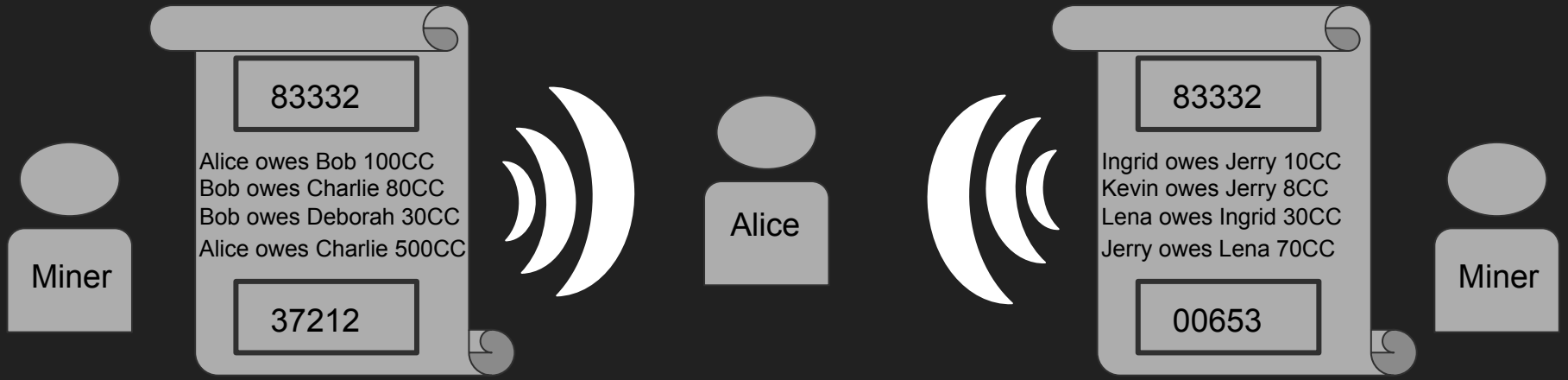


Proof of Work



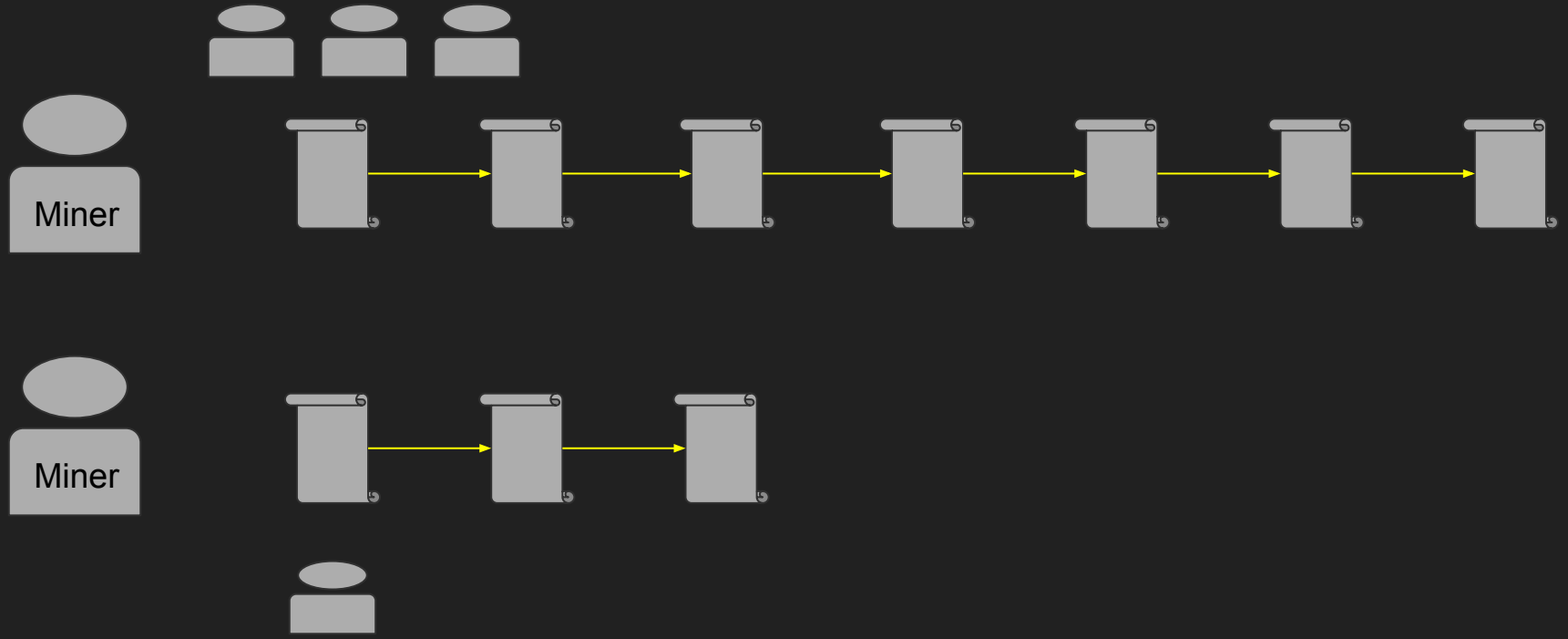
Proof of Work

- What if Alice hears two different Miners at the same time?



Proof of Work

- Answer: choose the longest chain



Proof of Work

- Also protection against fraud!
  - (So long as you don’t have 51% of computation power)