Visualizing Proof-of-Work Algorithms using MicroProcessors

A Little About Me...

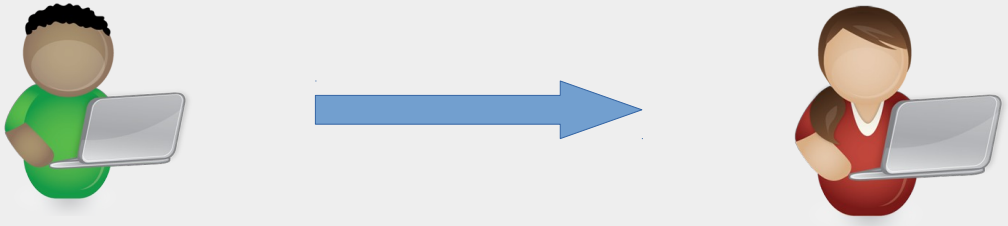
- Software Engineer @ Microsoft in Pittsburgh
- Run https://chaintuts.com creating Bitcoin & blockchain related tutorials
  - On YouTube, Twitter (@chaintuts), Github (chaintuts)
  - Articles, Videos, and Code
  - Check it out and share!
- Interested in how the tech works and its societal & financial impacts

Why Proof-of-Work?

- Traditional financial systems required *trust* in a central authority
- Ex: Bob pays Alice via PayPal – PayPal verifies Bob has funds, deposits in Alice's account

Why Proof-of-Work?

- Bitcoin is *decentralized* and *peer-to-peer*
- But this comes with problems!
  - How do we prevent fraudulent transactions like double-spends, etc.?

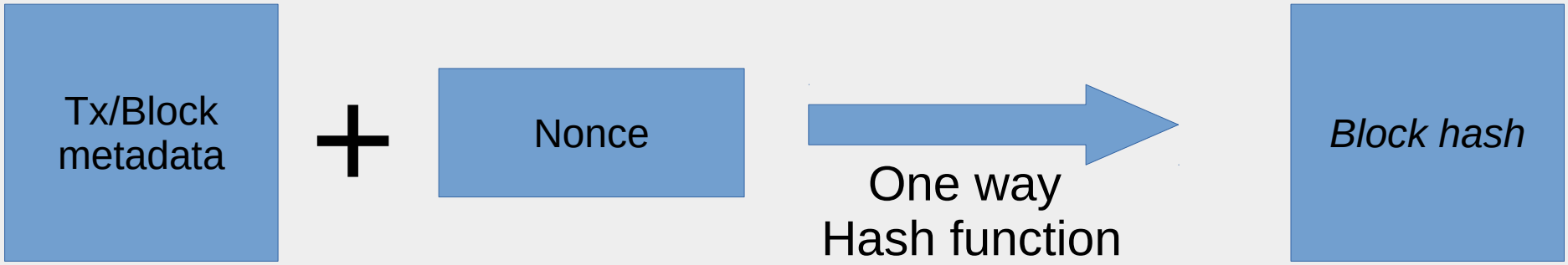


Blockchain & POW

- Everyone running Bitcoin software gets a copy of a distributed ledger called the blockchain
  - Like a spreadsheet that shows all transfers – we can infer address balances from this
- Every 10 minutes, pending txs batch processed into a new “block”
- This batch processing contains a security “challenge” called proof-of-work

Blockchain & POW

- Each “block” contains tx data + proof of work *nonce* (a random number)
- This gets run through a one way function called a *hash*
- *Block hash* has to meet security challenge requirements

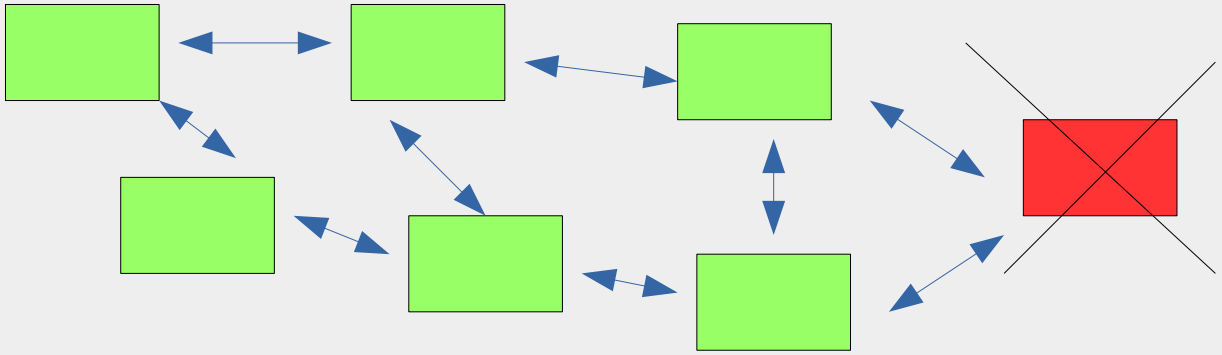


Blockchain & POW

- Only way to find *nonce* that meets guidelines is to guess a bunch of times – no formula or predictable outcome
- Once the answer is found, anyone can verify in one step!
- Bitcoin nodes share and *verify* the shared blockchain ledger so that the rules are followed

Blockchain & POW

- Network ignores malicious blocks that don’t fit the rules
- **Attackers would have to out-solve the rest of the Bitcoin network – nearly impossible at scale!**



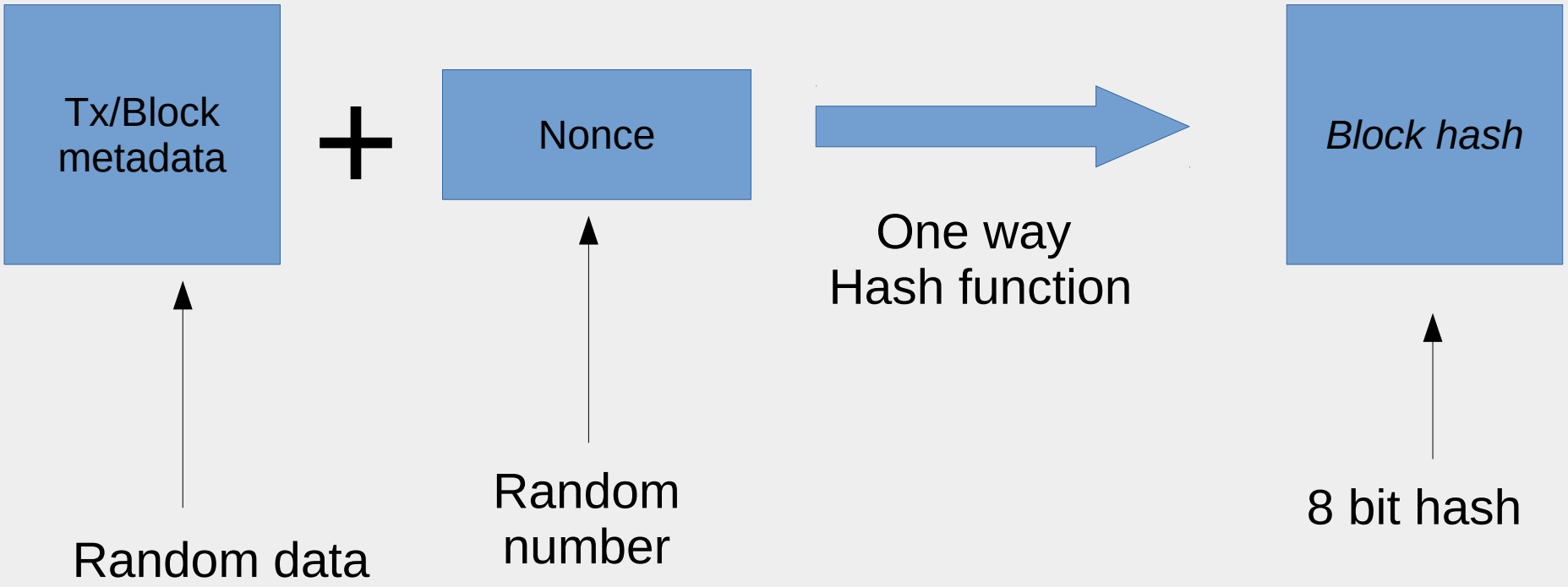
BTC: ~75 quintillion attempts / sec
BCH: ~2 quintillion attempts / sec
ETH: ~170 trillion attempts / sec
LTC: ~400 trillion attempts /sec

Hypothetically attacker: 100 billion attempts / sec?

What is the POW - “Security Challenge”?

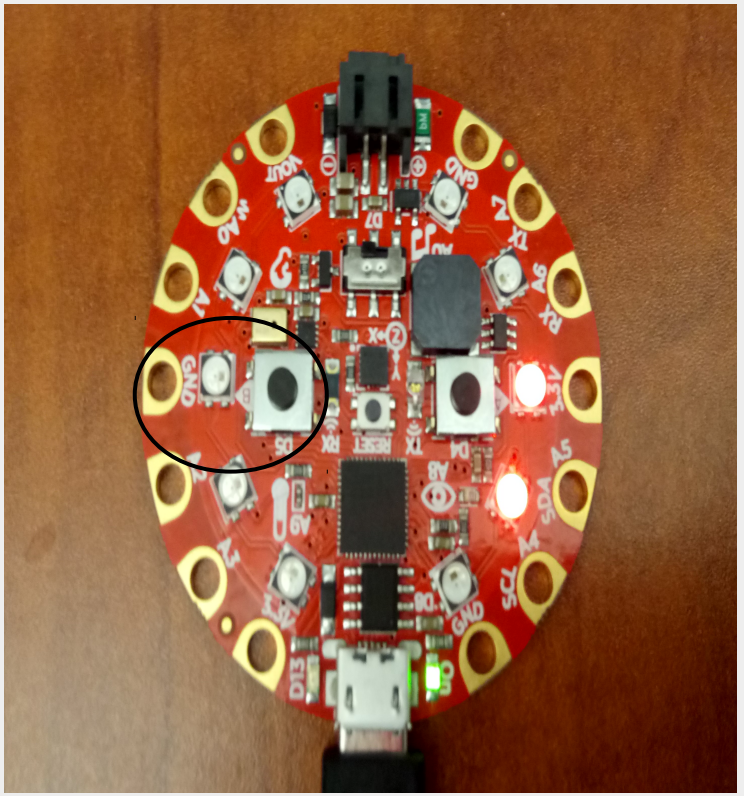
- Hashes (result of the one way function) are just numbers represented in binary
- The challenge is to find a *nonce* such that the resulting hash is **less than** the *difficulty target*
- The lower the *difficulty*, the longer it takes to find an answer!!
  - Bitcoin adjusts target so a solution is found and a new block is processed about every 10 minutes

Our POW Simulation



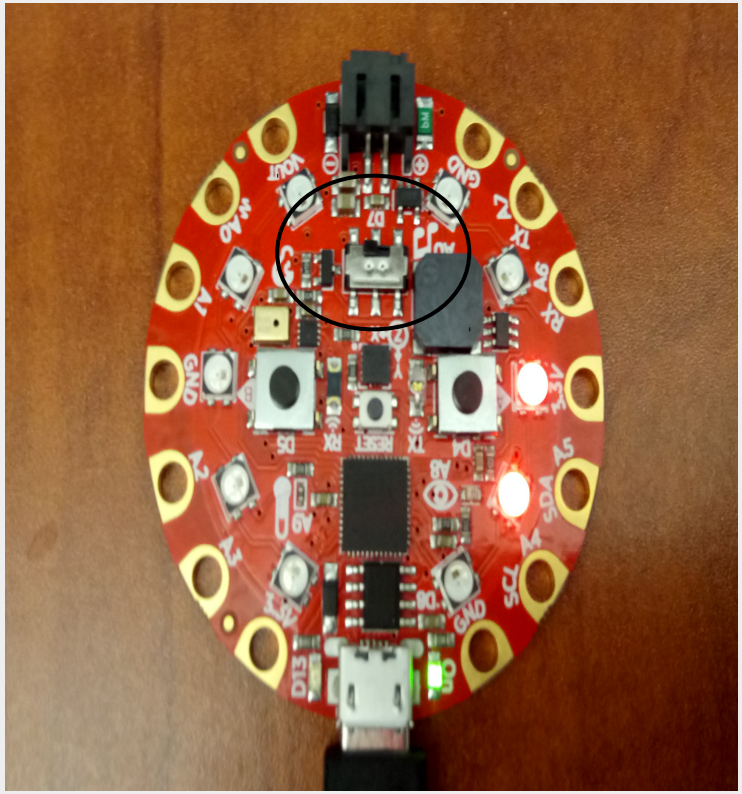
Our POW Simulation

- The difficulty target menu – press B to set difficulty from 1 – 7
  - Let’s do a sample run at 2
- **Accessibility:** Shake the board to turn on “sound mode”



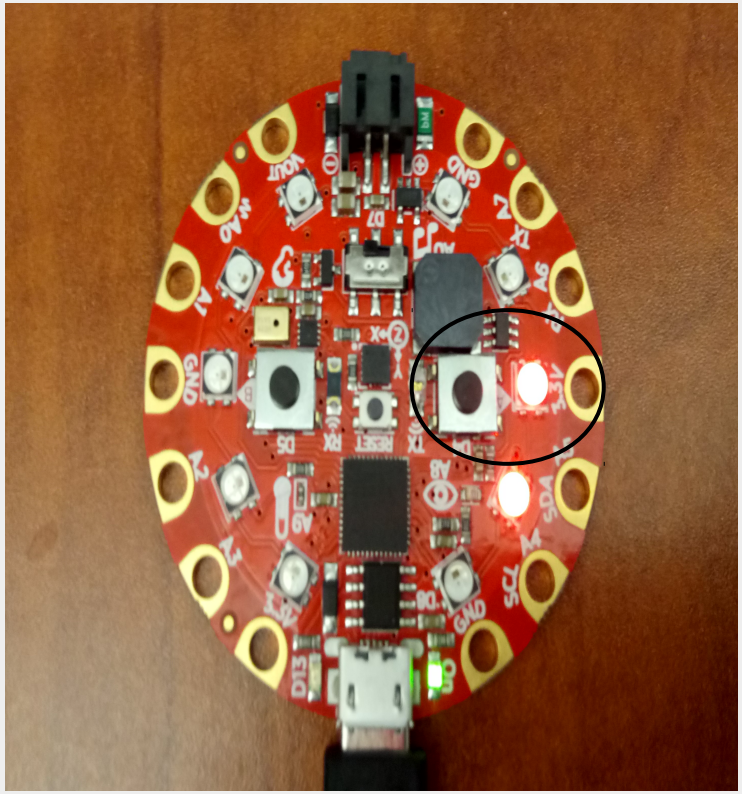
Our POW Simulation

- Optional – ensure toggle switch is set toward the “Ear”, Vout, A0
  - Enables logging so you can see attempts across runs



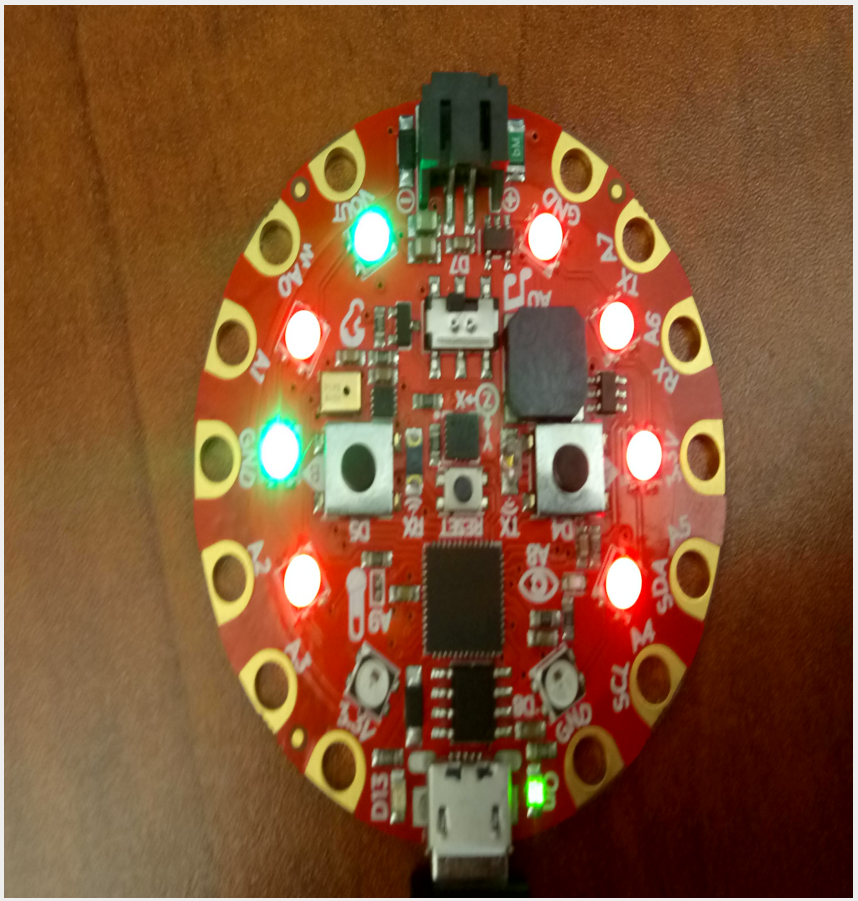
Our POW Simulation

- Start simulation by pressing A
  - LEDs will flash attempts represented in binary!!



Our POW Simulation

- When a solution is found, LEDs will stay lit!



Final “block hash”:

0 0 0 0 1 0 1 0

Our POW Simulation

- 8 bit numbers:

0 0 1 0 0 0 0 0

128 64 32 16 8 4 2 1

1 0 0 1 1 1 1 1

...

0 0 0 0 1 0 1 0

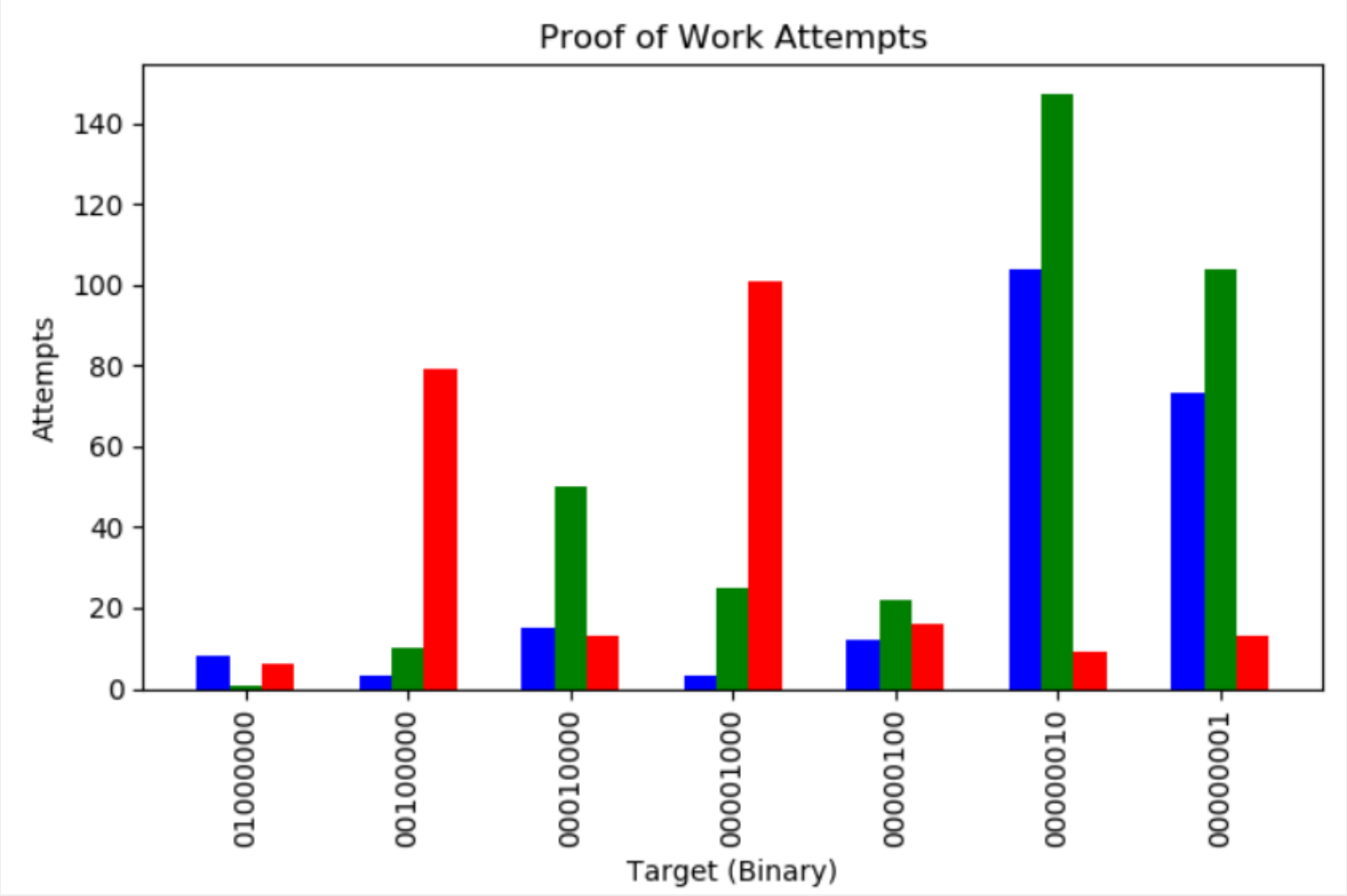
Target hash is 32 or less

Guess 1 – Hash is 159

Guess n – Hash is 10
Solution is found!!

Our POW Simulation

- Lower target = lower probability of a guess being a solution (more difficult)



Our POW Simulation

- If you'd like – do this visualization yourself after a few runs!
  - https://chaintuts.com/graph_pow.py
  - Plug in Circuit Playground
  - Copy `pow_log.csv` to your machine and run Python script in the same directory

Simulation vs. Real Bitcoin

- Size of numbers
  - Our simulation is 8 bit (0 – 255)
  - Bitcoin uses 256 bit numbers (0 – HUGE)

Simulation vs. Real Bitcoin

- Variables – Computing Power, Time, Difficulty
  - Our simulation is one dimensional – takes longer as problem gets harder
  - In Bitcoin, the more power that runs in parallel (all the mining nodes), the harder the problem is made to keep block time at 10 minutes

The more power on the network, the harder fraud is!

Questions?

Visualizing Proof-of-Work Algorithms using MicroProcessors

A Little About Me...

- Software Engineer @ Microsoft in Pittsburgh
- Run https://chaintuts.com creating Bitcoin & blockchain related tutorials
  - On YouTube, Twitter (@chaintuts), Github (chaintuts)
  - Articles, Videos, and Code
  - Check it out and share!
- Interested in how the tech works and its societal & financial impacts

Why Proof-of-Work?

- Traditional financial systems required *trust* in a central authority
- Ex: Bob pays Alice via PayPal – PayPal verifies Bob has funds, deposits in Alice's account

Why Proof-of-Work?

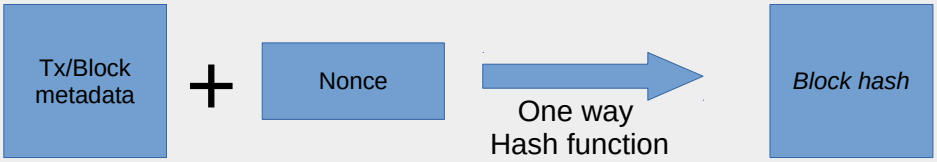
- Bitcoin is *decentralized* and *peer-to-peer*
- But this comes with problems!
  - How do we prevent fraudulent transactions like double-spends, etc.?

Blockchain & POW

- Everyone running Bitcoin software gets a copy of a distributed ledger called the blockchain
  - Like a spreadsheet that shows all transfers – we can infer address balances from this
- Every 10 minutes, pending txs batch processed into a new “block”
- This batch processing contains a security “challenge” called proof-of-work

Blockchain & POW

- Each “block” contains tx data + proof of work *nonce* (a random number)
- This gets run through a one way function called a *hash*
- *Block hash* has to meet security challenge requirements

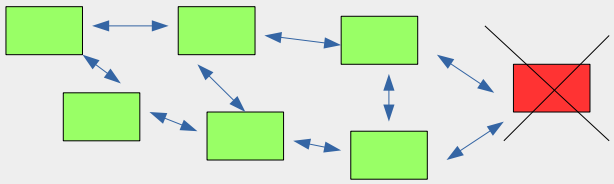


Blockchain & POW

- Only way to find *nonce* that meets guidelines is to guess a bunch of times – no formula or predictable outcome
- Once the answer is found, anyone can verify in one step!
- Bitcoin nodes share and *verify* the shared blockchain ledger so that the rules are followed

Blockchain & POW

- Network ignores malicious blocks that don't fit the rules
- **Attackers would have to out-solve the rest of the Bitcoin network – nearly impossible at scale!**



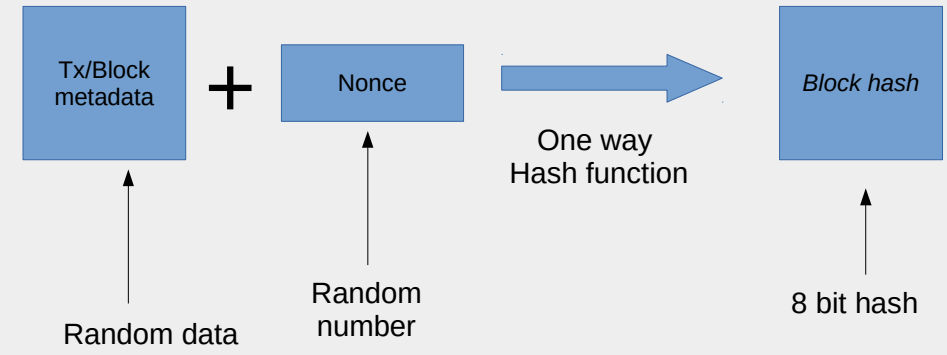
BTC: ~75 quintillion attempts / sec
BCH: ~2 quintillion attempts / sec
ETH: ~170 trillion attempts / sec
LTC: ~400 trillion attempts /sec

Hypothetically attacker: 100 billion attempts / sec?

What is the POW - “Security Challenge”?

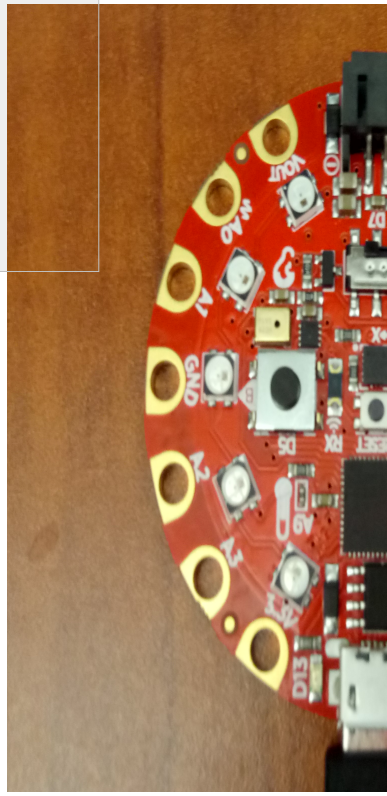
- Hashes (result of the one way function) are just numbers represented in binary
- The challenge is to find a *nonce* such that the resulting hash is **less than** the *difficulty target*
- The lower the *difficulty*, the longer it takes to find an answer!!
  - Bitcoin adjusts target so a solution is found and a new block is processed about every 10 minutes

Our POW Simulation



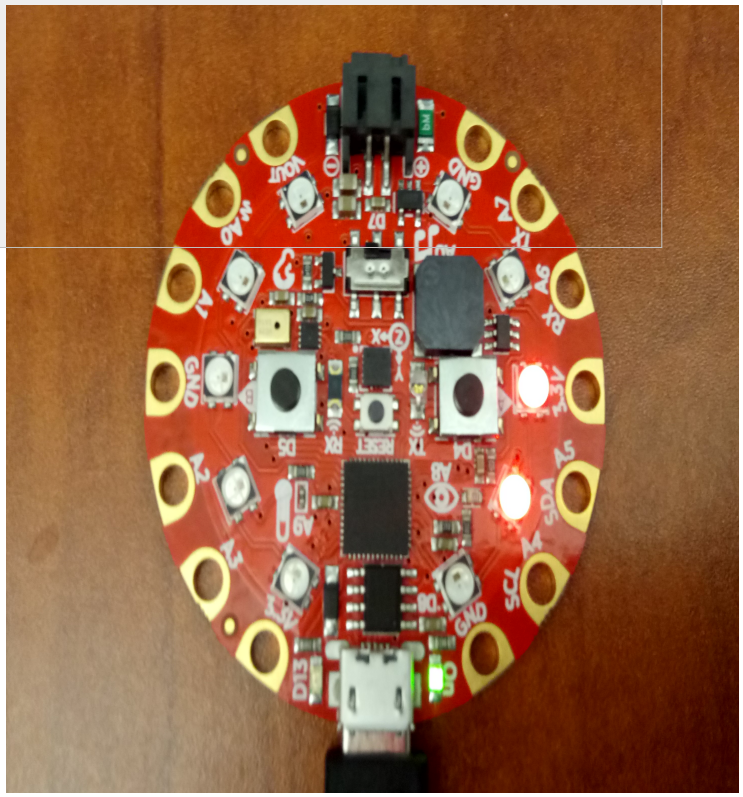
Our POW Simulation

- The difficulty target menu – press B to set difficulty from 1 – 7
  - Let’s do a sample run at 2
- **Accessibility:** Shake the board to turn on “sound mode”



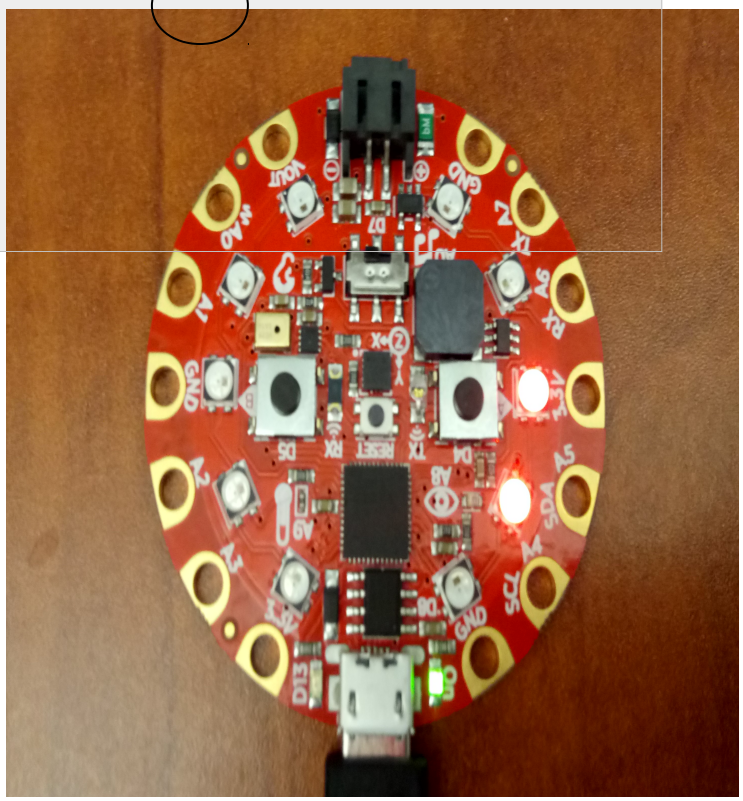
Our POW Simulation

- Optional – ensure toggle switch is set toward the “Ear”, Vout, A0
  - Enables logging so you can see attempts across runs



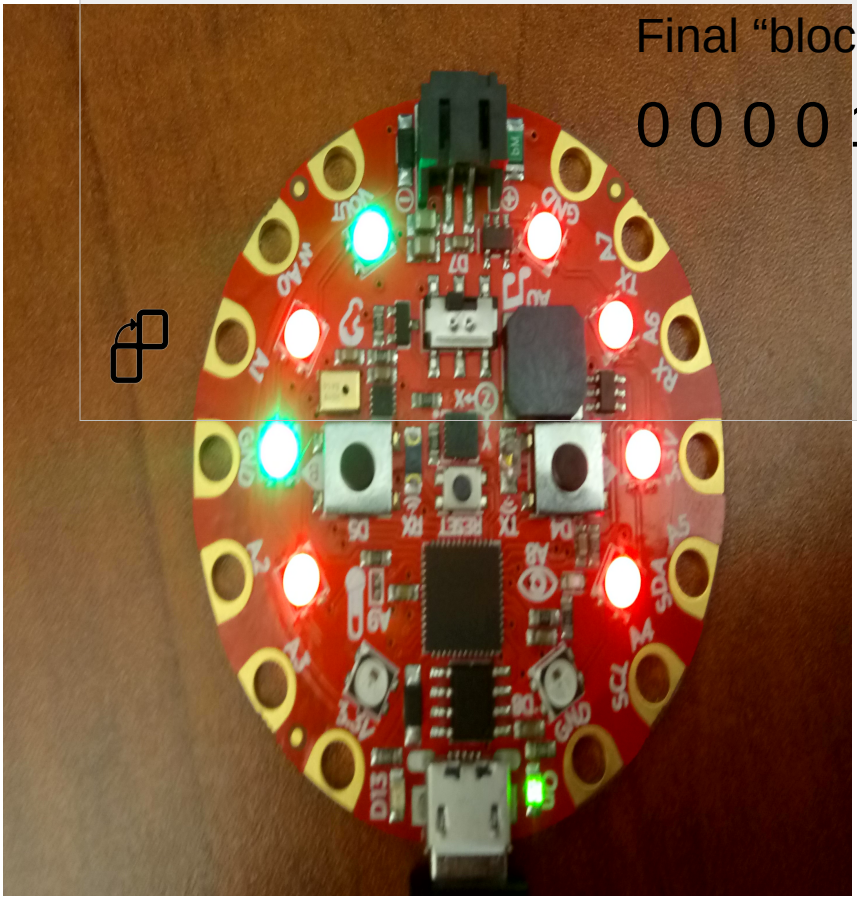
Our POW Simulation

- Start simulation by pressing A
  - LEDs will flash attempts represented in binary!!



Our POW Simulation

- When a solution is found, LEDs will stay lit!



Final “block hash”:

0 0 0 0 1 0 1 0

Our POW Simulation

- 8 bit numbers:

0 0 1 0 0 0 0 0

128 64 32 16 8 4 2 1

Target hash is 32 or less

1 0 0 1 1 1 1 1

Guess 1 – Hash is 159

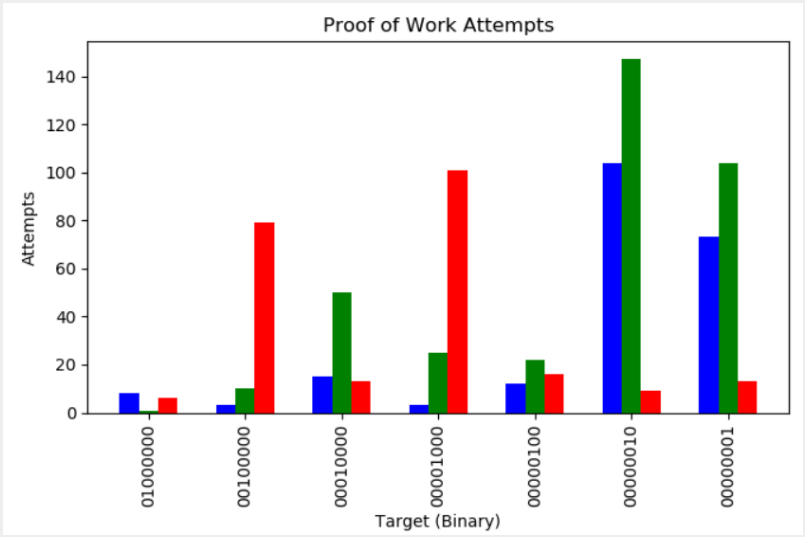
…

0 0 0 0 1 0 1 0

Guess n – Hash is 10
Solution is found!!

Our POW Simulation

- Lower target = lower probability of a guess being a solution (more difficult)



Our POW Simulation

- If you'd like – do this visualization yourself after a few runs!
  - https://chaintuts.com/graph_pow.py
  - Plug in Circuit Playground
  - Copy `pow_log.csv` to your machine and run Python script in the same directory

Simulation vs. Real Bitcoin

- Size of numbers
  - Our simulation is 8 bit (0 – 255)
  - Bitcoin uses 256 bit numbers (0 – HUGE)

Simulation vs. Real Bitcoin

- Variables – Computing Power, Time, Difficulty
  - Our simulation is one dimensional – takes longer as problem gets harder
  - In Bitcoin, the more power that runs in parallel (all the mining nodes), the harder the problem is made to keep block time at 10 minutes

  The more power on the network, the harder fraud is!

Questions?