Consensus in the Internet Setting

Ertem Nusret Tas
Recap of the Last Lecture

- Byzantine Generals Problem
- Definition of Byzantine adversary
- Synchronous and asynchronous networks
- Byzantine Broadcast
- Dolev-Strong (1983)
- State Machine Replication (SMR)
- Security properties for SMR protocols: Safety and Liveness
Sybil Attack

How to select the nodes that participate in consensus?

Two variants:

- *Permissioned*: There is a fixed set of nodes (previous lecture).
- *Permissionless*: Anyone satisfying certain criteria can participate.

Can we accept any node that has a signing key to participate in consensus?

Sybil Attack!
Consensus protocols with Sybil resistance are typically based on a bounded (scarce) resource:

<table>
<thead>
<tr>
<th>Resource dedicated to the protocol</th>
<th>Some Example Blockchains</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proof-of-Work</strong></td>
<td>Total computational power</td>
</tr>
<tr>
<td></td>
<td>Bitcoin, PoW Ethereum...</td>
</tr>
<tr>
<td><strong>Proof-of-Stake</strong></td>
<td>Total number of coins</td>
</tr>
<tr>
<td></td>
<td>Algorand, Cardano, Cosmos, PoS Ethereum...</td>
</tr>
<tr>
<td><strong>Proof-of-Space/Time</strong></td>
<td>Total storage across time</td>
</tr>
<tr>
<td></td>
<td>Chia, Filecoin...</td>
</tr>
</tbody>
</table>

How does Proof-of-Work prevent Sybil attacks?

We assume that the adversary controls a small fraction of the scarce resource!
To mine a new block, a miner must find *nonce* such that

\[ H(h_{prev}, txn \ root, nonce) < Target = \frac{2^{256}}{D} \]

Each miner tries different nonces until one of them finds a nonce that satisfies the above equation.

**Difficulty:** How many nonces on average miners try until finding a block?

*New block:* random process but app. once in every 10 minutes
Genesis block

Bitcoin: Block Headers

BH₁

version (4 bytes)
prev (32 bytes)
time (4 bytes)
bits (4 bytes)
nonce (4 bytes)
Tx root (32 bytes)

BH₂

prev
Tx root

80 bytes

Target (T): \( \frac{2^{256}}{D} \)
Bitcoin: Difficulty Adjustment

New target: $T_2 = T_1 \frac{t_1}{2016 \times 10 \text{ mins}}$

New target: $T_3 = T_2 \frac{t_2}{2016 \times 10 \text{ mins}}$

2016 blocks
Time it took to mine: $t_1 (\text{min})$
Target: $T_1$

2016 blocks
Time it took to mine: $t_2 (\text{min})$
Target: $T_2$

2016 blocks
Time it took to mine: $t_3 (\text{min})$
Target: $T_3$

New target is not allowed to be more than $4x$ old target.
New target is not allowed to be less than $\frac{1}{4} x$ old target.
Nakamoto Consensus

Bitcoin uses **Nakamoto consensus**:

- **Fork-choice / proposal rule**: At any given time, each honest miner attempts to extend (i.e., mines on the tip of) the heaviest (longest for us) chain in its view (Ties broken adversarially).

- **Confirmation rule**: Each miner confirms the block (along with its prefix) that is $k$-deep within the longest chain in its view.
  - In practice, $k = 6$.
  - Miners and clients accept the transactions in the latest confirmed block and its prefix as their log.
  - Note that *confirmation* is different from *finalization*.

- **Leader selection rule**: Proof-of-Work.
Nakamoto Consensus

Available under dynamic participation

k=2

Confirmed

txs₁

txs₃

txs₄
Characterized by *open participation*:

- Adversary can create many Sybil nodes to take over the protocol.
- Honest participants can come and go at will.

**Goals:**

- Limit adversary’s participation.
  - **Sybil resistance (e.g., Proof-of-Work)!**
  - Maintain availability (liveness) of the protocol against changing participation by the honest nodes.
  - **Dynamic availability!**
Can we show that Bitcoin is secure under synchrony against a Byzantine adversary?

What would be the best possible resilience?

\[ \beta < \frac{1}{2} \]

Fraction of the mining power controlled by the adversary.
Nakamoto’s Private Attack: $\beta \geq 1/2$

Private attack (mostly) fails if $\lambda_a < \lambda_h$, i.e., if $\beta < 1 - \beta$, i.e., if $\beta < \frac{1}{2}$.

Private attack (mostly) succeeds if $\lambda_a \geq \lambda_h$, i.e., if $\beta \geq 1 - \beta$, i.e., if $\beta \geq \frac{1}{2}$.

Can another attack succeed?

A Peer-to-Peer Electronic Cash System (2008)
Forking

Multiple honest blocks at the same height due to network delay.
Adversary’s chain grows at rate proportional to (shown by $\propto$) $\beta$!
Honest miners’ chain grows at rate less than $1 - \beta$ because of forking!

Now, adversary succeeds if $\beta \geq \frac{(1-\beta)}{2}$, which implies $\beta \geq \frac{1}{3}$!!
**Theorem:** If $\beta < 1/2$, there exists a small enough mining rate $\lambda(\Delta, \beta) = \lambda_a + \lambda_h$ (by changing difficulty) such that Bitcoin satisfies security (safety and liveness) except with error probability $e^{-\Omega(k)}$ under synchronous network.

- This is the error probability for confirmation.
- We say ‘confirmation’ instead of finalization because when you confirm a block or transaction, you confirm it with an error probability...
- ...unlike finalizing a block where there is no error probability*

Now, we see why Bitcoin has 1 block every 10 minutes, instead of 1 block every second...

---

*This is the error probability for confirmation.

We say ‘confirmation’ instead of finalization because when you confirm a block or transaction, you confirm it with an error probability...

...unlike finalizing a block where there is no error probability.
Is $\beta < 1/2$ really?

Amounts paid to the miner:
- $tx_1$: 4 BTC
- $tx_2$: 3 BTC
- $tx_3$: 2 BTC
- $tx_4$: 1 BTC

Miners have incentive to violate the protocol!

Need to think about incentives!!
No Attacks on Bitcoin?

Ghash.IO had >50% in 2014
- Gave up mining power

Why are visible attacks not more frequent?

Miners care about the Bitcoin price?
- Can short the chain!

Might not always be rational to attack.

No guarantees for the future!
Is Bitcoin the Endgame?

Bitcoin provides Sybil resistance and dynamic availability. Is it the Endgame for consensus?

No!

Bitcoin is secure only under *synchrony* and loses security during periods of *asynchrony*. It *confirms* blocks with an error probability depending on $k$, i.e., blocks are not *finalized*.

Energy consumption?
Next lecture: Incentives and Accountability in Consensus
Slides going forward is optional material and present a simplified security proof for Nakamoto consensus.
Let’s recall the definition of security for SMR protocols. Let $ch_t^i$ denote the confirmed (i.e., $k$-deep) chain accepted by a client $i$ at time $t$.

**Safety (Consistency):**
- For any two clients $i$ and $j$, and times $t$ and $s$: $ch_t^i \leq ch_s^j$ (prefix of) or vice versa, i.e., chains are consistent.

**Liveness:**
- If a transaction $tx$ is input to an honest replica at some time $t$, then for all clients $i$, and times $s \geq t + T_{conf}$: $tx \in ch_s^i$. 

No double spend
No censorship
Modelling Bitcoin (Optional)

Many different miners, each with infinitesimal power.
Total mining rate: \( \lambda \) (1/minutes).
In Bitcoin, \( \lambda = 1/10 \).

Adversary is Byzantine and controls \( \beta < \frac{1}{2} \) fraction of the mining power.
- Adversarial mining rate: \( \lambda_a = \beta \lambda \)
- Honest mining rate: \( \lambda_h = (1 - \beta)\lambda \)

Each mined block is adversarial with probability \( \beta \) independent of other blocks.

Network is synchronous with a known upper bound \( \Delta \) on delay.
Suppose there is at most one honest block at every height.

This is the case if the network delay $\Delta = 0$.

GR (2022): If any attack succeeds in violating a target transaction tx’s safety, then the private attack with premining also succeeds in violating the target transaction’s safety.
We will show that if any attack succeeds in violating safety of a target transaction tx within the first honestly mined block, then the private attack also succeeds in violating the target transaction’s safety.

For the full proof, see “Bitcoin’s Latency Security Analysis Made Simple”.
Suppose a transaction $tx$ is confirmed within the first block $b$ mined by the honest miners in an honest view.

Let’s observe a ‘reorg’ of block $b$ by some attack.

We will show that the private attack will also succeed in ‘reorging’ $b$!
Block $b$ contains the transaction $tx$ that is ‘reorged’.
Consider the first time that $t$ block $b$ is reorged.
• Right before $t$, block $c$ is seen at the tip of the longest chain by an honest node.
• Right after $t$, block $d$ is seen at the tip of the longest chain by another (potentially the same) honest node.
Fact 1: At each height until $h_c$, there is at least one adversary block.

- Why?
- Because there can be at most one honest block at any height.
• **Fact 2:** Every block after $h_c$ thru $h_d$ are adversarial (one block per height).
  • Why?
  • Otherwise, we contradict with the definition of blocks $c$ and $d$. 

$$A \geq h_d$$
• **Fact 3:** There are at most $h_c$ honest blocks.

$$H \leq h_c$$
Security Proof: Safety (Optional)

- Combining everything...
  - \( H \leq h_c \)
  - \( A \geq h_d \)
  - \( h_d \geq h_c \geq k \)
  - This implies \( A \geq H \) and \( A \geq k \).

Private attack also succeeds!
Why?
Security Proof: Safety (Optional)

\[ A \geq H \text{ and } A \geq k: \]

\[ H \leq A \]

Private attack also succeeds!
Security Proof: Safety (Optional)

If every honest block is at a separate height...
Best attack to reorg a transaction is the **private attack with premining**!
Probability that a private attack with premining succeeds $\leq e^{-\Omega(k)}$; if $\lambda_a < \lambda_h$, i.e., $\beta < 1/2$!
Safety!
Security Proof: Safety (Optional)

\[ t_0 = 0 \]
\[ t_1 \]
\[ t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7 \quad t_8 \]

Multiple honest blocks at the same height due to network delay.

Forking!

Probability that a block is an honest block at a unique height: \( e^{-\lambda \Delta (1 - \beta)} \)
**Trick:** We give honest blocks that fall into the same height as previous honest blocks to the adversary.

Mining rate of ‘honest’ blocks with new definition = $e^{-\lambda \Delta (1 - \beta)}$
Every honest block is *again* at a separate height!

Best attack to reorg a transaction is the private attack with premining.

Probability that a private attack with premining succeeds \( \leq e^{-\Omega(k)} \); if \( \frac{1}{2} < e^{-\lambda\Delta(1 - \beta)} \).

Safety!
Security Proof: Liveness (Optional)

Growth rate of the blockchain $\geq e^{-\lambda \Delta}(1 - \beta) \lambda$.

Arrival rate of adversary blocks: $\beta \lambda$

If $\frac{1}{2} < e^{-\lambda \Delta}(1 - \beta)$, then $e^{-\lambda \Delta}(1 - \beta) \lambda > \beta \lambda$.

Thus, over a sufficiently large time interval (call this $u$), the $k$-deep prefix of the longest chain in the view of each honest node must contain new honest blocks except with probability $e^{-\Omega(u)}$.

Liveness!