Nakamoto Consensus

Benedikt Bünz
Consensus

- Security Properties:
  - Consistency: Honest nodes do not contradict
  - Liveness: Progress is made

- Network Models
  - Synchronous: Messages get delivered immediately
  - Partially Synchronous: Messages are out of order
Consensus Committee Leader Accepts/Rejects TXs
Problems with approach

- Known committee
  - (must communicate)
- Large committee
  - Large communication
- Honest majority (incentives)
- Predictable Leader
  - Bribing 💯
Recap

genesis block

BH₁

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

80 bytes

BH₂

prev

Tx root

BH₃

prev

Tx root

...
Nakamoto Consensus

PoW: Find nonce s.t. $H(\text{Block}) < \text{Target}$
Target s.t. blocks found every 10 min
Nakamoto Consensus

BH\_1
Prev
Time
Nonce
Root

BH\_2
Prev
Time
Nonce
Root

BH\_3

H(BH\_2)
Time
Root

H(BH\_2)
Time
83512
Root

coinbase Tx

Genasis
Nakamoto Consensus

- **Genesis**: Start point
- **BH₁**: Previous block hash
- **BH₂**: Previous block hash
- **BH₃**: Previous block hash

- **Time**: Time stamp
- **Nonce**: Random number
- **Root**: Hash of the block

- **H(BH₂³)**: Hash of the block hash
- **H(BH₃)**: Hash of the block hash

- **coinbase Tx**: Initial transaction
Nakamoto Consensus

- Miners “race” to add blocks
  - Prepare Block Template
  - Find nonce (PoW solution)
  - One winner every ~10 min
  - Target adjusted every 2 weeks
  - Probability winning ~ Computation power

PoW:
Find nonce s.t. $H(\text{Block}) < \text{Target}$
Nakamoto Consensus

- Miners “race” to add blocks
  - Need to find PoW solution
  - Probability winning ~ Computation power
  - One winner every ~10 min
  - Target adjusted every 2016 Blocks
  - On average 2016 blocks = 2 weeks
- (Honest) miners extend longest chain
- Timestamps must be roughly accurate
- *All transactions must be valid*
- Blocks/Transactions become final after $k$ blocks
- Leader election/race combined with tx adding

PoW:
Find nonce s.t.
\[ H(\text{Block}) < \text{Target} \]
Forks and Orphans

Working on B

Working on A
Forks and Orphans

Orphaned block

Working on B C

Working on A C
Preventing double spends

I’ll wait k blocks

Here are the keys

I’ll just produce a different chain

We’ll be working on the longest chain

No Car TX

3 BTC For Car
51% Attack

Cloud

I’ll just produce a different chain

New longest chain

No Car TX

3 BTC For Car
Nakamoto properties

1. **Consistency.** Honest nodes agree on all but last \( k \) blocks (except with prob. \( O(2^{-k}) \))

2. **Chain quality.** Any consecutive \( k \) blocks contain “sufficiently many” honest blocks (except with prob. \( O(2^{-k}) \)). Miners controlling \( p \) fraction of power should roughly mine \( p \) fraction of blocks.

3. **Chain growth.** Chain grows at a steady rate.

   \[ g \text{-chain growth: Growth by } k \text{ blocks every } k/g \text{ “rounds”} \]
Nakamoto properties => Blockchain

• Consistency implies Blockchain consistency

• Chain growth + chain quality implies Blockchain liveness
  - The chain grows by k blocks every k/g periods
  - By chain quality, a high fraction of blocks are contributed by honest miners, and therefore include all transactions they heard so far
Nakamoto consensus

**Consistency intuition:** Suppose adversary has 49% power

- Adversary can fork chain by 1 block faster than honest miners extend current chain w/ prob. close to $\frac{1}{2}$, or by 2 with prob. $\frac{1}{4}$
  - No problem! If adversary broadcasts fork, everyone switches, this is now the longest chain

- What if miner forks chain 6 blocks deep and doesn’t broadcast until it has a longer chain than honest?
  - Probability $\frac{1}{64}$ it mines 6 blocks before honest mines 1
  - Probability $< 8 \times 2^{-7}$ it mines 7 blocks before honest mines 2
  - What is probability adversary ever catches up?
**Consistency intuition:** (continued...)

Suppose adversary has $p < 1/2$ fraction of power. What is the probability adversary catches up from 6 blocks behind?

- *Simplified model:* repeated rounds, in every round adversary catches up by 1 block with probability $p$, and falls behind by 1 block with probability $1 - p$.

- Biased random walk on number line starting at 0, +1 with probability $p$ and -1 with probability $1 - p$. Probability walk ever reaches 6?

- Probability $P_z$ that walk ever reaches $+z$ is \( \left( \frac{p}{1-p} \right)^z \) (e.g. $p = 1/3$, then $P_6 < 0.0062$)
What goes wrong if adversary has $p > 1/2$ power?

- Adversary’s private fork grows at faster rate than honest chain
- For any $k$, adversary starts $k$ blocks behind, will eventually catch up to length of honest chain
45% Attack

Cloud

I’ll just produce a different chain

No Car TX

3 BTC For Car

Incur network delays and orphans
Nakamoto consensus

Network delay & work difficulty

• What happens if miners can solve puzzles faster than they can propagate solutions through network?

• Adversary might receive the next valid block $\Delta$ steps ahead of the other honest nodes ($\Delta = \text{delay}$)

$$\Rightarrow \text{Adversary starts working on next puzzle with a } \Delta \text{ time head start over other honest nodes}$$

$O(\Delta)$ “free” hash trials
Nakamoto consensus

Adjusting difficulty for $\Delta$

Honest mining fraction (say 60%)

\[
\frac{\alpha}{1 + \alpha \lambda \Delta} > \beta
\]

Network mining rate (1 Block/10min)

Adversary power (40%)

$\lambda \Delta$ is the mining rate * the delay. That is #blocks/delay (say 0.1)

**Intuition:**
On average, honest nodes waste a $\Delta$ steps of work every block they find, while adversary never wastes work. So “effective” reduced honest rate is \[
\frac{\alpha}{1 + \alpha \lambda \Delta}
\]
Theorem: There exists a k such that Nakamoto Consensus has consistency and liveness if and only if:

\[
\frac{\alpha}{1 + \alpha \lambda \Delta} > \beta
\]

Private chain attack = Actual security (was an open question)

Interpretation:
The less Δ relative to block time, the closer this gets to \( \alpha > \beta \). For large Δ the adversary needs much less than 50% of the mining power to attack.
Blue line = max value of $\beta$ s.t. $\frac{\alpha}{1+\alpha \lambda \Delta} > \beta$

Nakamoto magically chose $\frac{1}{\lambda \Delta} = 60$
(10 min blocktime assuming 10s network delay)
Short Forks and Liveness

Long forks are impossible but short forks may not be.
This is a liveness issue.
Need to ensure that some “honest” blocks are in the longest chain.
Could be used to censor transactions.
Nakamoto chain quality

• Chain Quality is percentage of honestly mined blocks
  • Honest mined blocks include all transactions!
  • Prevents censorship

• Say the adversary controls a $p$ fraction of the mining power $p < \frac{1}{2}$

• Ideally honest parties mine a $1 - p$ fraction

• Can prove they mine at least $1 - \frac{p}{1-p}$

$$p = \frac{1}{3} \rightarrow Q = \frac{1}{2}$$

If $p > \frac{1}{2}$ then adversary could mine every block in worst case
$\Rightarrow$ chain quality is 0
Chain Quality Theorem

• For every $p < \frac{1}{2}$, if mining difficulty is appropriately set as function of network delay $\Delta$ then Nakamoto consensus guarantees:

1. Consistency (for $\alpha, \beta, \Delta$ satisfying formula)
2. Chain quality: $1 - \frac{p}{1-p}$ fraction blocks honest
3. $O(1/\Delta)$-Chain growth
Nakamoto Consensus and Partial Synchrony

- Nakamoto Consensus can be secure up to \( \frac{1}{2} \) corruptions
- Can tolerate network delays
- Contradicts partial synchrony lower bound?
  - No
- Protocol needs a bound on delays (c)
- Consistency broken even with honest nodes
Nakamoto Properties

- Anonymous participation
- Nodes can join/leave
  - Very scalable
  - Sleeping Beauty property
- Leader not known beforehand
  - Makes bribing harder
- Up to ½ corruptions

- Slow
  - Even when everyone is honest
- Resource intensive
  - PoS based possible
- No finality
- No guarantees under long delays
Incentives

- Mining (solving PoW puzzles) is very expensive
- *Honest* majority does not seem realistic
- Satoshi’s genius idea: Combine issuance and rewards
- Block reward only paid if block part of longest chain
- High Variance -> Mining Pools
Incentives

Large opportunity cost for unsuccessful attacks
Selfish mining attack

Attacker has 1/3 of mining power. Miner is rational (maximize rewards)

Keeps block private

Once attacker has a two block lead he can mine until honest chains catch up
Selfish mining attack

Attacker has 1/3 of mining power. Miner is rational (maximize rewards)

Keeps block private

Once attacker has a two block lead he can mine until honest chains catch up

Attacker publishes chain and invalidates honest blocks
Selfish mining attack

Keeps block private

Attacker has 1/3 of mining power. Miner is rational (maximize rewards)

If honest miners finds block: Publish and it’s a block race (Attacker has at least 1/3 p of winning)
Selfish mining analysis

Honest reward = $\frac{1}{3}$

P Block Race:
2/3

Win: $\frac{1}{3}$ chance
2 of 3 blocks
Reward $\frac{2}{3}$
Loose: $\frac{2}{3}$ chance
Reward 0

$\frac{2}{3} \times \frac{1}{3} \times \frac{2}{3} + \frac{1}{3} \times \frac{2}{3} = \frac{10}{27} > \frac{1}{3}$

P Run away: $\frac{1}{3}$

Reward $> \frac{2}{3}$
Selfish Mining

Explains why chain quality < 1 - p

Optimal Selfish mining
No Attacks in Practice?

- Attacks possible but not seen
- Ghash.IO had >50%
  - Gave up mining power
  - No Selfish mining attacks
- Why?
  - Miners care about Bitcoin price
  - Not rational in $ terms to attack
  - Not guaranteed in the future
Next lecture:
Randomness beacons, VDFs, large scale PoS