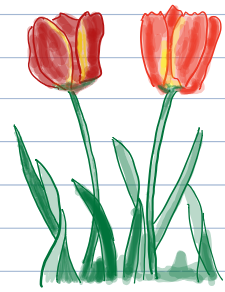
Secure Distributed Systems

CompSci 661 / 461



This video

- Selfish Mining
- why it is a denial of Service attack
- why it enhances effective mining rate (of the attacker) after a time
- How it works.

Selfish Mining

An attacker strategy that lowers the absolute number of blocks mined by the honest miners is lower than expected.

The absolute number of blocks mined by the attacker is the same.

I.e., the percentage of blocks mined by the attacker is higher.

Attacker has fraction q of mining power.
Honest has $p = 1 - q$, and assume $p > q$.

A set of miners with fraction q of mining power acting honestly should expect fraction q of all blocks mined by the entire network of miners.

But with selfish mining, this set will get something closer to

$$\hat{q} = \frac{q}{1-q}$$ of all blocks mined.

↑ the new proportion of blocks mined by the attacker

Shhh!! $\frac{q}{1-q}$ is not the exact answer! But it's very close... just go with it!

Example. $q = 0.4$

$$\hat{q} = \frac{0.4}{0.6} = 2/3$$ of all blocks mined.

instead of $2/5$.

In this example, we are assuming the best case for the attacker ($\gamma = 1$ ... which wont make sense to you for a few pages)

Important!

This is a proportion! not an increase in blocks mined by a selfish miner.

Example. Say that $q = 0.4$.

Normally, if 10 blocks were mined by the network, attacker would get 4. (on average)

With selfish mining, they still get 4.

We also know they got $2/3$ of all blocks.

That means the total is

$$2/3 \, x = 4$$
$$x = 6$$

I.e., the honest mined 2 blocks with $p = 0.6$ out of 10 total. $2/6 = 1/3 = 1 - q$

It's not selfish — it's a denial of service attack.

Another subtle point:

Let's say that instead of 6 blocks, the attacker does this until 2016 blocks are mined.

– Well!! Now the difficulty will change!

How is difficulty affected by selfish mining?

Normally, difficulty is adjusted as follows (in Bitcoin)

Let D be the current difficulty.
Let D' be the adjusted difficulty.

After 2016 blocks, let's say it took t minutes.

$$D' = D \cdot \frac{2016 \cdot 10 \text{ minutes}}{t \text{ minutes}}$$

Example

If it took 10% longer than expected?

$$D' = D \cdot \frac{20160}{22176} = \frac{1}{1.1} D = 0.91 D$$

In fact, under selfish mining instead of 1 block per 10 minute interval, there will be $(1-q)$ blocks.

Why? Because:

We know the mining rate of the attacker doesn't change.

And we know that they are getting $\frac{q}{1-q}$ of the X blocks produced. For the same mining power they normally get fraction q of T blocks produced.

So

$$\frac{q}{1-q} X = qT$$

$$X = (1-q)T$$

I.e., the network will produce $(1-q)T$ blocks only during an interval where T are produced normally.

That is, 2016 blocks will take $\frac{2016 \cdot 10}{(1-q)}$ minutes instead.

If selfish mining goes on for 2016 blocks:

$$D' = D \cdot \frac{20160}{\left(\frac{2016 \cdot 10}{1-q}\right)} = D \cdot (1-q)$$

Example: An attacker with $q = 0.4$ selfish mines $D' = 0.6D$.

Now attacker will produce more blocks!

That's because mining is easier by a ratio $D/D' = \frac{D}{D(1-q)} = \frac{1}{(1-q)}$

the selfish miner has an effective hash rate.

$$q \cdot \frac{1}{(1-q)} = \frac{q}{1-q}.$$

Instead of producing qT blocks, they will produce $q/1-q\, T$ blocks.

Example: $q = 0.4$. $T = 2016$ blocks

I. Before selfish mining:

attacker: $qT = 0.4 \cdot 2016 = 806$ blocks in 14 days — $\frac{25 \text{ min}}{\text{block}}$

honest: $(1-q)T = 0.6 \cdot 2016 = 1210$ blocks in 14 days — $\frac{16 \text{ min}}{\text{block}}$

II During selfish mining:

Two ways to look at this.

① How many blocks each during T·10 minutes?

attacker: $qT = 0.4 \cdot 2016 = 806$ blocks in 14 days

25 min/block

honest: produces many fewer during this time.

Let x be total blocks by attacker and honest

$$806 = \frac{q}{1-q} x$$

$$806 = 2/3\, x$$

$$x = 1210$$

therefore honest produce only

$1210 - 806 = 403$ in 14 days.

$\frac{50 \text{ min}}{\text{block}}$

② Eventually, the network will produce 2016 blocks.

This will take $\frac{T \cdot 10}{(1-q)}$ minutes $= \frac{20160}{0.6} = 33{,}600$ minutes ≈ 23 days.

The attacker will have discovered.

$\frac{q}{1-q} \cdot 2016 = \frac{2}{3} \cdot 2016 = 1{,}344$ blocks.
in 23 days

$\frac{25 \text{ min}}{\text{block}}$

the honest miners will have discovered

$\frac{1-q}{q} \cdot 2016 = \frac{1}{3} \cdot 2016 = 672$ blocks.
in 23 days

$\frac{50 \text{ min}}{\text{block}}$

III After 2016 blocks of selfish mining:

attacker will mine

$$\left(\frac{\hat{q}}{1-\hat{q}}\right)\cdot 2016 = 1{,}344 \text{ blocks}$$

in 14 days

$\frac{15 \text{ min}}{\text{block}}$

honest will have

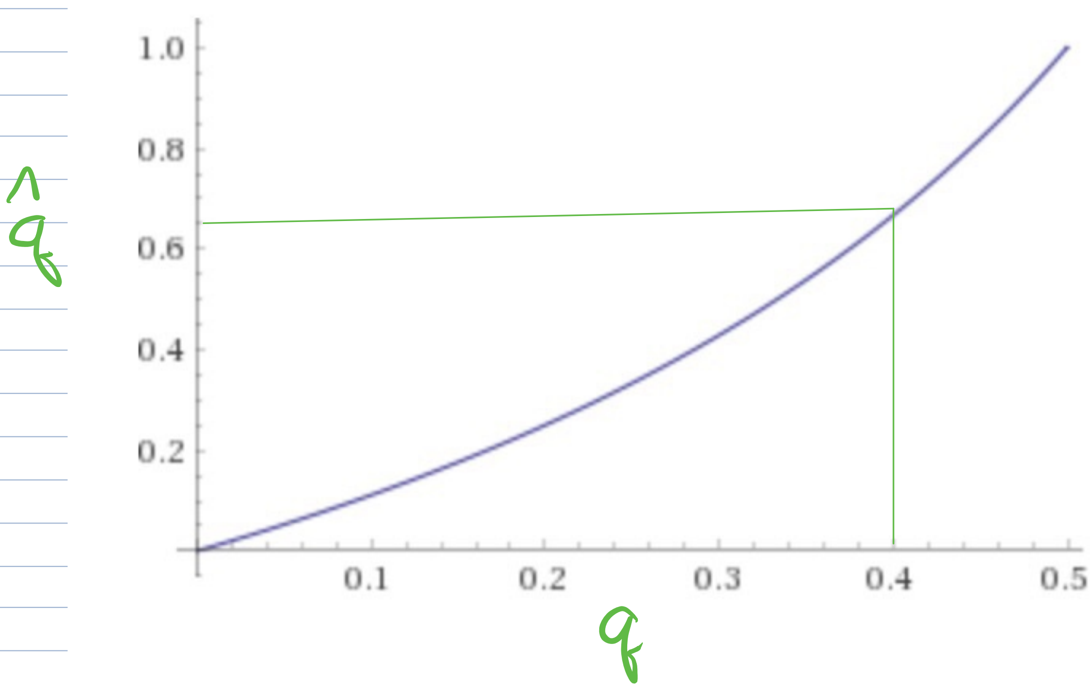
$$\left(\frac{1-\hat{q}}{\hat{q}}\right)\cdot 2016 = 672 \text{ blocks}$$

in 14 days

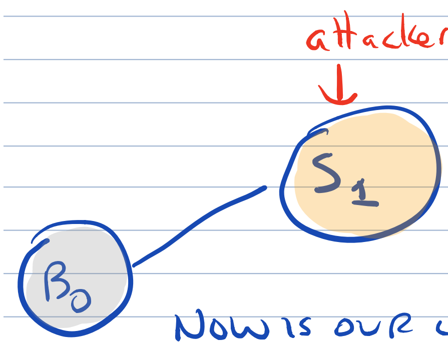
$\frac{30 \text{ min}}{\text{block}}$.

plot	$\frac{q}{1-q}$	$q = 0$ to 0.5

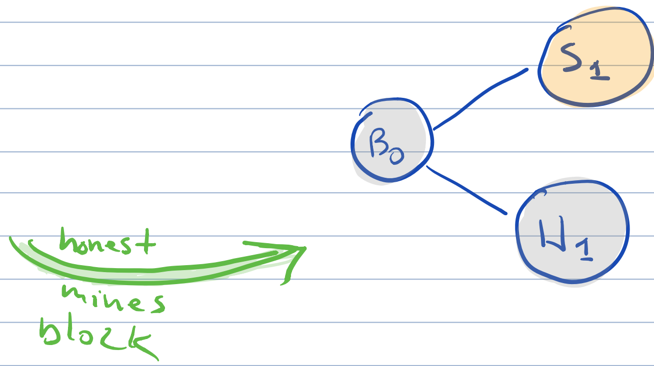
Plot:



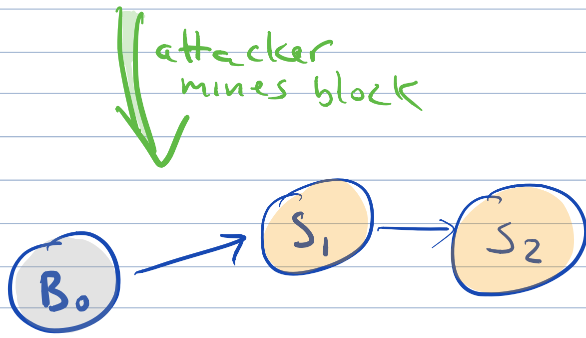
Here's the selfish mining strategy.

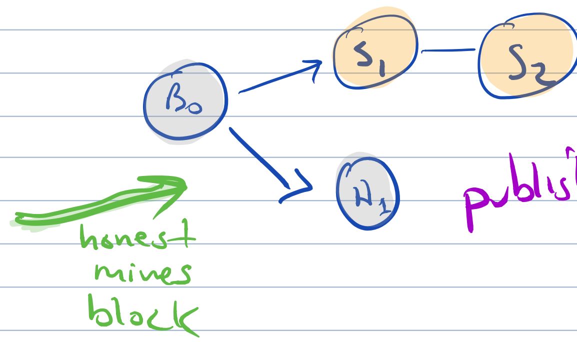


Now is our chance!
We are one block ahead – dont tell any one!



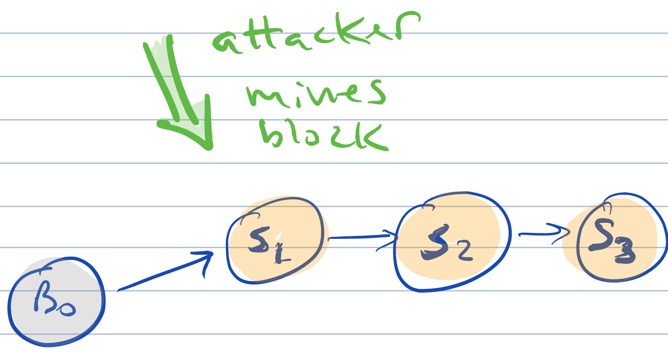
there's a chance the H1 was heard first by some γ fraction of miners

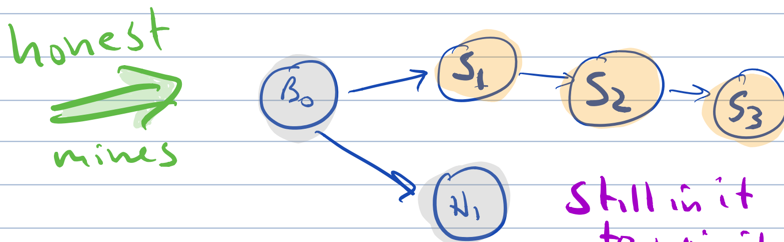




publish!

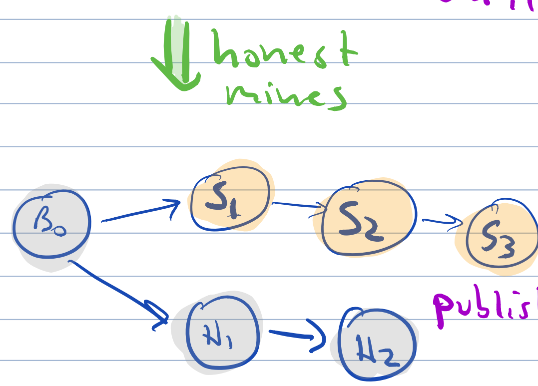
now either we get to S_3 before H_1 is published or not.





Still in it to win it!

Some missing states here.



publish!

Basic Algorithm

Initialization

public chain ← all known blocks
private chain ← all known blocks
branch_len ← ∅

We mine at the head of the private chain.

Two scenarios: (A) Selfish miner finds a block.
(OR)
(B) Honest miners find a block

(A)

$\Delta \leftarrow$ len(private) − len(public) — Difference before new block was mined.

append new block to private

branch_len += 1

IF $\Delta == \emptyset$ and branch_len == 2
THEN publish branch
branch_len ← ∅

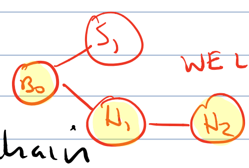
} we won after a tie

(B)

$\Delta \leftarrow$ len(private) − len(public)
append new block to public

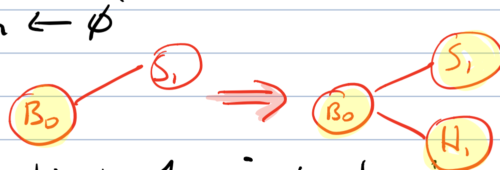
yellow means published

IF ($\Delta == \emptyset$)
THEN private chain ← public chain
branch_len ← ∅



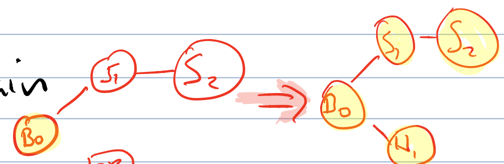
WE LOST!

ELSE IF ($\Delta == 1$)
THEN publish last block of private chain

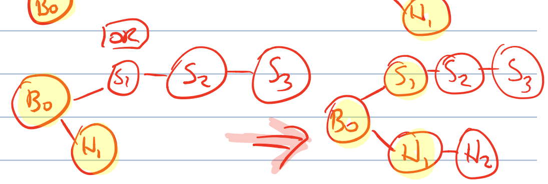


go for a tie!

ELSE IF ($\Delta == 2$)
THEN publish all of the private chain
branch_len ← ∅



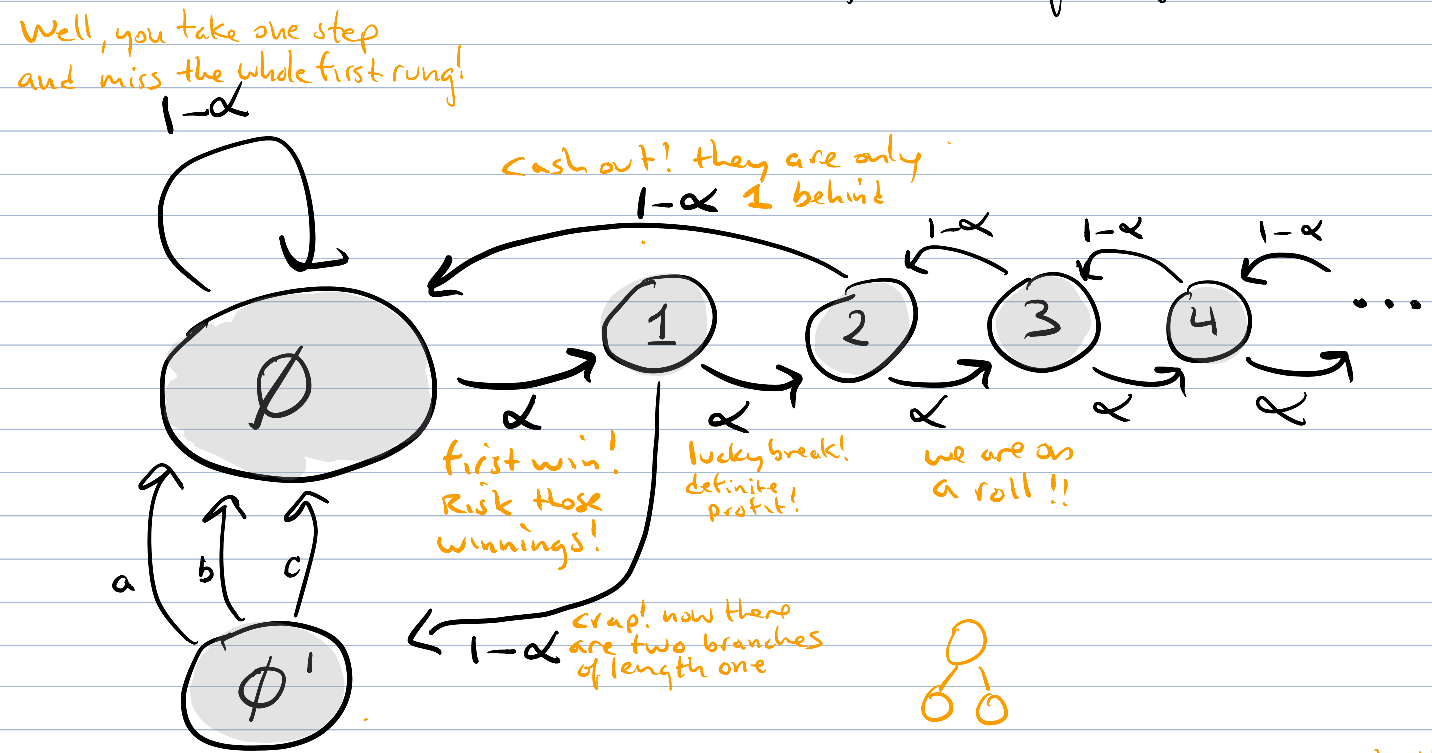
ELSE
publish the first unpublished block in private chain



State Machine

$\alpha = q$ above; α is variable used in paper

α = mining power of selfish miners



$a = (1-\alpha)(1-\gamma)$ } with prob $(1-\alpha)\cdot(1-\gamma)$, honest will add to honest branch; we lost!

$b = (1-\alpha)\gamma$ } with prob $(1-\alpha)\gamma$, honest will add to private/attacker branch; cash out!

$c = \alpha$ ← with prob α, we are now 1 ahead; so cash out!

With eclipse attacks, $\gamma = 1$.

Above I claimed that $\hat{q} = \frac{q}{1-q}$. $\hat{\alpha} = \frac{\alpha}{1-\alpha}$

The true result is

Revenue of selfish is

$$R = \frac{\alpha(1-\alpha)^2(4\alpha+\gamma(1-2\alpha))-\alpha^3}{1-\alpha(1+(2-\alpha)\alpha)} \leq \frac{\alpha}{1-\alpha}$$

Saphirshtein has shown that no selfish mining strategy is better than $\frac{\alpha}{1-\alpha}$.

And also has shown that there exists a strategy better than the equation above.