ECE 595: Foundations of Blockchain Systems **October 28, 2020**

Lecture 9

Lecturer: Sreeram Kannan *Scribe: Liu Cao*

Outline: This lecture covers two aspects:

1. The liveness under the case of $\lambda_h > \lambda_a$ and $\Delta = 0$;
2. The safety and liveness under the case of general Δ.

9.1 Liveness

The definition of liveness is that new honest transactions get added to the chain. This implies that any new (valid) transaction issued by a honest client should be added to the chain. Note that if this does not happen, you can have censoring. This mechanism thus creates anti-censoring, i.e., any new transaction issued by a honest client should be eventually added to the chain. Here, we should notice that the liveness is not defined in terms of blocks but in terms of transactions.

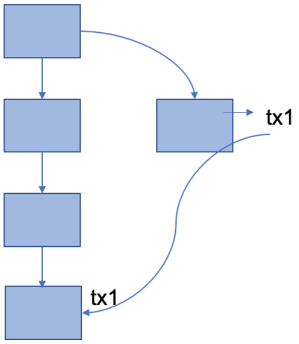

Figure 9.1: Liveness is in terms of transactions

9.2 Chain growth

Chain growth is one of aspects to prove the liveness. One clear way in which the Liveness can be blocked is that the total length of the chain is not growing anymore for whatever reasons, i.e., the length of the chain remains the same and no new blocks are added to the chain.

Chain growth is a condition that new blocks are added. Every honest block increments the longest chain by 1. If we denote $L_v(t)$ is the length of chain at node v at time t, $L_v(t)$ should be non-decreasing in t because the honest nodes never accept a shorter chain. Hence, the chain grows at least at rate λ_h(blocks/sec). Since if $\lambda_a > \lambda_h$, as Fig 9.2 shows, there is an issue that the adversary chain may not contain any transaction issued by honest clients when the adversary chain is accepted. Therefore, chain growth is not enough to guarantee the liveness.

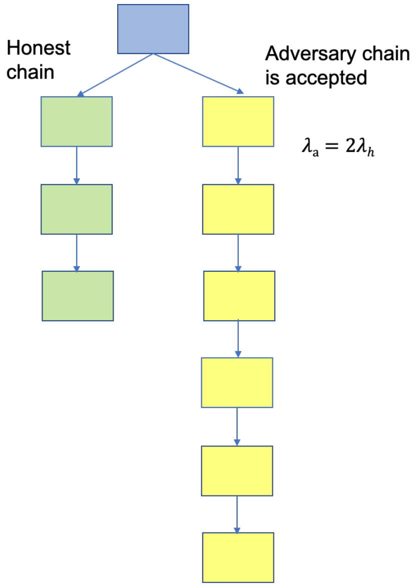

Figure 9.2: One example for the chain growth

9.3 Chain quality

Chain quality is another aspect to prove the liveness. The chain quality γ denotes the fraction of honest blocks in the longest chain. The ideal value of γ should be equal to 1 while the potentially realizable (fair) value of γ is given by

$$\gamma = \frac{\lambda_h}{\lambda_a + \lambda_h}.$$

Any non-zero lower bound is still useful, which gives some liveness. It is because some fraction of blocks are honest, which means these blocks are willing to include the honest transactions.

9.3.1 Attack on chain quality

Attack on chain quality is something which lowers the chain quality below the fair value. This attack is called **selfish mining**.

Let's see an example, which is illustrated in Fig 9.3 shows. We create an adversary who is trying to get more blocks into the chain than what is expected. The normal behavior is to broadcast as soon as mining. The honest nodes always do that because they will send it to everybody as soon as a block is mined. However, the adversary may not do it. Thus, the adversary behavior is to hold the block in private. If the length of adversary chain is the same as the honest chain, the adversary will continue to keep mining until the adversary chain takes advantage of honest chain. In the figure, the adversary mined two blocks $3'$ and $4'$ and released both blocks as soon as the honest block 4 is released. Here, we always assume the adversary can rush its blocks ahead of honest nodes, i.e., the adversary needs this rushing behavior to attack the network like this. **Tie-breaking rule** that nodes favor earlier chain is used for selecting one of chains when the length of both chains are the same. Since the adversary is well connected in the network, it can get its blocks ahead faster than the honest nodes. Thus, the two honest blocks 3 and 4 were kicked out of the chain. In this run, the total number of adversary blocks is 2 while the total number of honest blocks is 4. From the perspective of fair allocation, the fraction of adversary blocks should be 2/6 while the fraction of honest blocks should be 4/6. However, for the actual chain in the end of game, the fraction of adversary blocks should be 2/4 while the fraction of honest blocks should be 2/4. Therefore, the attack is trying to change the fraction of blocks that the adversary got in the chain, being higher than the representative chain quality.

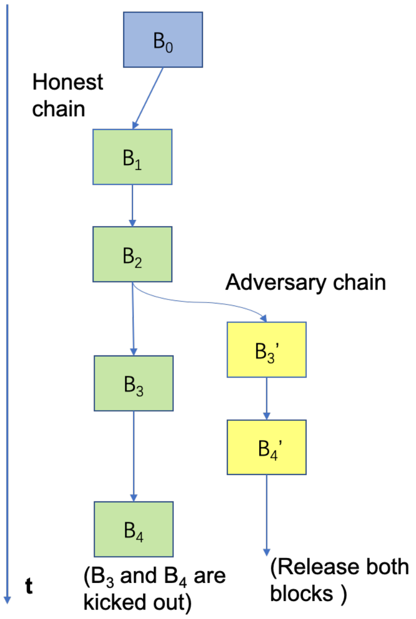

Figure 9.3: One example for attack on chain quality

There are some scenarios which should be discussed:

1. **Scenario 1**: If the block $4'$ is mined after the block 4, the adversary will lose the advantage as soon as the block 4 is released, and its own block $3'$ is wasted.

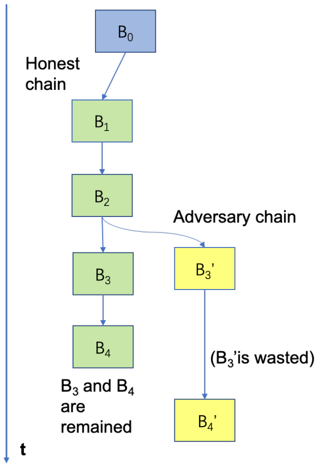

Figure 9.4: Scenario 1

2. **Scenario 2**: If the block $4'$ is mined before the block 3, the adversary will have the advantage as soon as the block 3 is released.

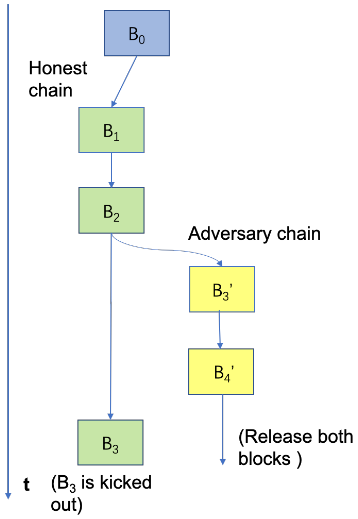

Figure 9.5: Scenario 2

3. **Scenario 3**: If the block $3'$ is mined before the block 3, the adversary will also have the advantage as soon as the block 3 is released.

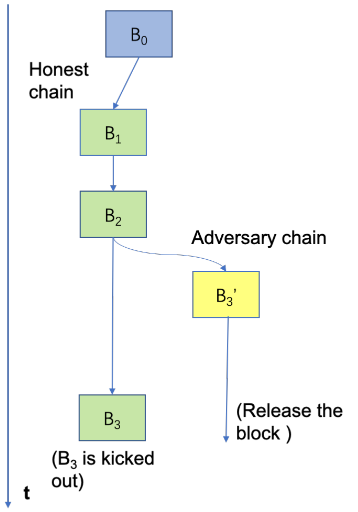

Figure 9.6: Scenario 3

4. **Scenario 4**: If the block $3'$ is mined after the block 3, the adversary will switch to start mining the block $4'$.

As we can see, calculating the optimal selfish mining strategy is non-trivial. The incentives of selfish mining for the adversary are:

1. Break the protocol.

2. Gain unfair rewards.

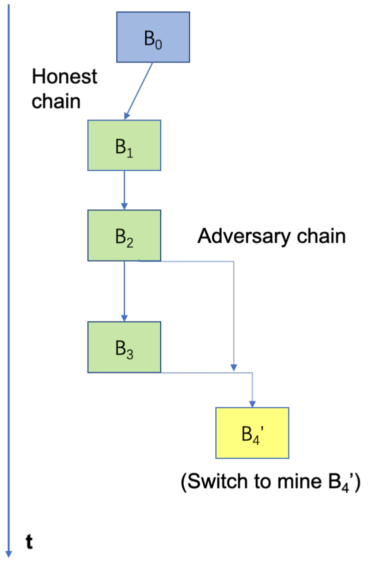

Figure 9.7: Scenario 4

9.3.2 Analysis of chain quality

Let's assume T time has passed. Denote $n_h(T)$ as the number of honest blocks in the chain at time T, and $n(T)$ is the total number of blocks in the chain at time T. Thus, the chain quality is given by

$$\gamma = \frac{n_h(T)}{n(T)},$$

when T is large. The lower bound and upper bound of $n(T)$ are

$$\lambda_h T \leq n(T) \leq (\lambda_a + \lambda_h)T.$$

Note that chain length increases every time there is a honest block, but it is not guarantee that all honest blocks are in the chain.
Define $n_a(T)$ is the number of adversary blocks in the chain at time T. The upper bound of $n_a(T)$ is

$$n_a(T) \leq \lambda_a T(1+\epsilon),$$

where ϵ is very small. Therefore, we have

$$\begin{aligned} n_h(T) &= n(T) - n_a(T) \\ &\geq \lambda_h T - \lambda_a T \\ &= (\lambda_h - \lambda_a)T. \end{aligned}$$

Based on $n(T) \leq (\lambda_a + \lambda_h)T$, the lower bound of chain quality γ is given by

$$\gamma = \frac{n_h(T)}{n(T)} \geq \frac{\lambda_h - \lambda_a}{\lambda_h + \lambda_a}.$$

This indicates that in order to kick out 1 honest block, at least 1 adversary block is required. Hence, at least $\lambda_h T - \lambda_a T$ honest blocks will remain in the chain. If $\lambda_h > \lambda_a$, then $\gamma > 0$.
Thus far, we can see the chain growth and chain quality implies the liveness.

9.4 Security for $\Delta > 0$

Key ideas of the previous proof for $\Delta = 0$ are as follow:

1. Property 1: No two honest blocks are at the same height.
2. Property 2: Every honest block forces the longest chain to grow.

For $\Delta > 0$, Property 1 and Property 2 are violated. Therefore, the key ideas for $\Delta > 0$:

1. Property 1*: No two loners are at the same height.
2. Property 2*: Every loner increases the chain height.

Loner is a block which is born when no other honest blocks are born (silence) within Δ before and after. ”Silence before” implies that the block builds on the longest chain held by the honest nodes. ”Silence after” implies that the block is heard by all honest nodes before they build any other blocks. In the earlier proof, we assume $\lambda_h > \lambda_a$, now we assume $\lambda_l > \lambda_a$ where λ_l is the rate of loners. The probability of event {A block is a loner} is equivalent to The probability of event {Silence before and silence after}, which is given by [1]

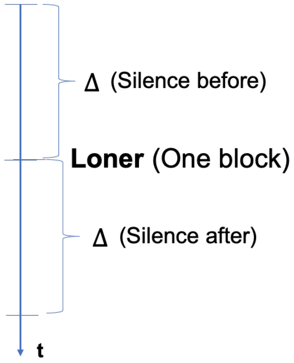

Figure 9.8: Scenario 1

$$Pr\{\text{A block is a loner}\} = [e^{-\lambda_h \Delta}][e^{-\lambda_h \Delta}] = e^{-2\lambda_h \Delta}.$$

Hence, the rate of loners λ_l is expressed as

$$\lambda_l = \lambda_h e^{-2\lambda_h \Delta}.$$

If $\lambda_l > \lambda_a$, it will imply the security.
The difference between the loner and censored process: The loner is a stronger condition than censored process because the loner enforces the silence on both sides.

References

[1] Alberto Leon-Garcia. Probability, statistics, and random processes for electrical engineering. 2017.