ECE 595: Foundations of Blockchain Systems **October 19, 2020**

Lecture 6

Lecturer: Sreeram Kannan *Scribe: Bowen Xue*

Outline: This lecture covers security of blockchains. We first explain the Adversarial model, and then give a definition to security of the blockchain; in the end, we use the adversarial model to analyze security of the consensus layer with regard to a special attack, private chain attack.

6.1 Adversarial model

In the adversarial model, there are two types of nodes. **Honest** nodes are blindly following all rules specified by a designed protocol, and the rest of the nodes are called **adversaries** who can do whatever actions desired for them. For making the adversarial model more concrete, we state the adversarial action space in the following axes:

- **where to mine blocks:** adversary can decide to mine at any height of the ledger
- **what to put inside blocks:** adversary can put any transactions inside any blocks
- **when to broadcast blocks:** adversary can choose any time to release its blocks
- **network delay control:** adversary can deliver a block in arbitrary orders within $(t, t + \Delta)$

Since the consensus layer relies on services provided by the P2P layer, we need clarification about the P2P properties before accurately defining the adversarial action space. We assume P2P enables **Δ-delay broadcast property**: if a block reaches some honest nodes at time t, then the block reaches all honest nodes before $t + \Delta$, where Δ is a fixed delay time. When the **network delay control** attack is used for a block generated at time t, adversaries can arbitrarily decide the delivering time of the block to any honest nodes before the block is eventually seen by all honest nodes at time $t + \Delta$. In summary, adversarial model defines powerful adversarial nodes who can freely decide any actions contained in the space, and such powerful model is useful because a large space contains more adversarial situations in the real world, and therefore our protocol is more useful if it can tolerate those harder conditions.

6.2 Security: Safety and Liveness

Traditionally, the security of a distributed system consists of two properties: **Safety** and **Liveness**. Safety is also called persistence, which means any transaction confirmed by an honest node will remain confirmed in all future. A system with only Safety property can sometimes be useless. For example, a system is useless to anyone if it stops accepting any new blocks after the genesis block is created by an honest node. This requires us to think of another useful property, Liveness: any honest transaction will eventually be confirmed. For example, if a transaction from an honest node is rejected by some adversarial nodes, the honest node should be able to resend the transaction to all other nodes until it gets confirmed. In addition to Safety and Liveness, a new property useful to the blockchain but ignored by the traditional definition is called **Fair order**: Fair order requires the order of transactions to be consistent to their arrival time. For example, there is a valuable paint sold for a fixed price in the system, it would be unfair if Bob gets the paint by cutting in line after many buyers already sent their transactions. Security (=Safety + Liveness) gives no guarantee on the relative order in the final ledger. In the following sections, we are restricted to the traditional security definition. We will discuss Fair order and its precise meaning later in the course.

6.3 Private chain attack

Adversary can conduct any attacks contained in the Adversarial model, in this lecture we focus on a specific attack considered in the original bitcoin paper: Private chain attack. A private chain attack is pictorially summarized in Fig 6.1, where there are two chains coming from the genesis block, and all blocks in the chains are labeled by their height after the genesis block. Blue chain are created by honest nodes and its ledger are visible by all nodes in the network; red chain is created by adversarial nodes who secretly mine and chain the blocks together. The private chain attack is successful if and only if there exist a $\ell \geq k$ such that an adversary mines ℓ blocks before honest nodes mine their ℓ blocks, where k corresponds to the k-deep confirmation rule. We will analyze the consensus security in two situations: without and with network latency Δ. But before that, let's characterize the mining process using tools from probability.

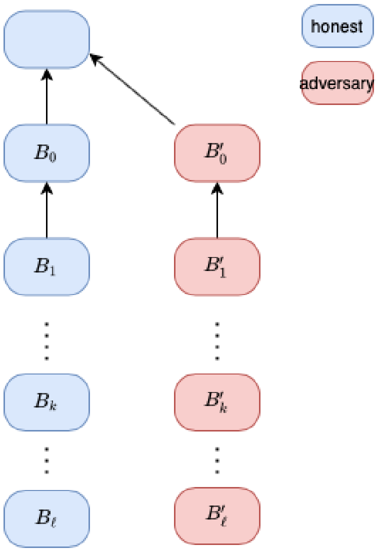

Figure 6.1: Private Chain attack

6.3.1 Model for mining

The event of successfully mining a block can be characterized by a geometric random variable which measures the number of Bernoulli trials for reaching one successful event. This model is reasonable because every hash computation can be considered as a Bernoulli trial with probability of success matching its mining power; and in practice, the mining uses a pseudo random generator SHA-256 which generates independent outputs as good as a real random source. A unique character of the geometric random variable is the memoryless property: the distribution of the waiting time until the next successful block does not depend on how much time has elapsed in the past. When the time slot becomes extremely small, a geometric random variable converges to an exponential random variable with the mean equals to $\frac{1}{\lambda}$, where λ is the block generation rate with a unit of (blocks per second). A counting process $\{N(t), t \geq 0\}$ represents the total number of events that have occurred up to time t (detailed definition see [1]). Because the number of blocks in a chain can be modeled with a counting process, $\{N(t), t \geq 0\}$ and the interarrival times in the counting process are independent and identically distributed exponential random variables, the height of a chain follows a Poisson process. Going forward, let's use the notation T_i^h be the mining time of an honest block $i \geq 1$, with an underlying exponential random variable with rate λ_h; similarly, let T_i^a be the mining time for adversaries with rate λ_a.

6.3.2 security without network delay

We are interested in the event E_ℓ when an adversary mines ℓ blocks before the honest nodes mine ℓ blocks, which can be written as $E_\ell = \{\sum_{i=1}^{\ell} T_i^a \leq \sum_{i=1}^{\ell} T_i^h\}$. To further simplify the equation, let's define a new random variable $D_i = T_i^h - T_i^a$, and $E_\ell = \{\sum_{i=1}^{\ell} D_i \geq 0\}$. The probability of the event is hence $\Pr\left[E_\ell\right] = \Pr\left[\sum_{i=1}^{\ell} D_i \geq 0\right] = \Pr\left[\frac{1}{\ell}\sum_{i=1}^{\ell} D_i \geq 0\right]$. The expected value of D_i is expressed as,

$$\mathbb{E}[D_i] = \mathbb{E}\left[T^h - T^a\right] = \frac{1}{\lambda_h} - \frac{1}{\lambda_a}$$

If $\lambda_h < \lambda_a$, then $\mathbb{E}\left[D_i\right] > 0$, the attack succeeds with high probability, because in expectation adversary launches the private attack by mining more blocks. We now focus on the other case $\lambda_h > \lambda_a$, and use Chernoff bound to derive an upper limit on the probability of E_ℓ

$$\begin{aligned}
\Pr\left[E_\ell\right] &= \Pr\left[\frac{1}{\ell}\sum_{i=1}^{\ell} D_i \geq 0\right] \\
&= \Pr\left[\sum_{i=1}^{\ell} D_i \geq 0\right] \\
&= \Pr\left[s\sum_{i=1}^{\ell} D_i \geq 0\right] \quad \text{; note that } s > 0 \\
&= \Pr\left[e^{s\sum_{i=1}^{\ell} D_i} \geq 1\right] \\
&\leq \mathbb{E}\left[e^{s\sum_{i=1}^{\ell} D_i}\right] \\
&= \mathbb{E}\left[\Pi_{i=1}^{\ell} e^{sD_i}\right] \\
&= \Pi_{i=1}^{\ell}\, \mathbb{E}\left[e^{sD_i}\right] \quad \text{; as } D_i \text{ are independent}
\end{aligned}$$

s is a parameter greater than 0. The inequality in the fifth line comes from Markov Inequality: If X is a nonnegative random variable, then for any $a > 0$, $\Pr\left[X \geq a\right] \leq \frac{\mathbb{E}X}{a}$. Its proof is fairly straightforward.

Proof.

$$\mathbb{E}[X] = \sum_i i \Pr\left[X = i\right] \geq \sum_{i \geq a} i \Pr\left[X = i\right] \geq \sum_{i \geq a} a \Pr\left[X = i\right] = a \Pr\left[X \geq a\right]$$

□

To compute $\mathbb{E}\left[e^{sD_i}\right]$,

$$\begin{aligned}\mathbb{E}\left[e^{sD_i}\right] &= \mathbb{E}\left[e^{sT_i^h}e^{-sT_i^a}\right] \\ &= \mathbb{E}\left[e^{sT_i^h}\right]\mathbb{E}\left[e^{-sT_i^a}\right] \\ &= \frac{\lambda_h}{\lambda_h - s}\frac{\lambda_a}{\lambda_a + s}\end{aligned}$$

The last equation is just integration. For example, $\mathbb{E}[e^{sx}] = \int \lambda e^{sx}e^{-\lambda x}\,dx = \frac{\lambda}{\lambda - s}$. Therefore for all $s > 0$

$$\Pr\left[E_\ell\right] \leq \Pi_{i=1}^{\ell}\left(\frac{\lambda_h}{\lambda_h - s}\frac{\lambda_a}{\lambda_a + s}\right) = \left[\frac{\lambda_h}{\lambda_h - s}\frac{\lambda_a}{\lambda_a + s}\right]^\ell$$

By our assumption $\lambda_h > \lambda_a$, and let's set $s = \frac{\lambda_h - \lambda_a}{2}$

$$\Pr\left[E_\ell\right] \leq \left[\frac{4\lambda_h\lambda_a}{(\lambda_a + \lambda_h)^2}\right]^\ell$$

One can prove $\frac{4\lambda_h\lambda_a}{(\lambda_a+\lambda_h)^2} < 1$ if $\lambda_h \neq \lambda_a$

Proof.

$$\begin{aligned}(\lambda_a + \lambda_h)^2 - 4\lambda_h\lambda_a &= \lambda_a^2 - 2\lambda_h\lambda_a + \lambda_h^2 \\ &= (\lambda_a - \lambda_h)^2 \\ &> 0\end{aligned}$$

□

Therefore

$$\Pr\left[E_\ell\right] \leq \left[\frac{4\lambda_h\lambda_a}{(\lambda_a + \lambda_h)^2}\right]^\ell = \left[e^{-c}\right]^\ell$$

where c has the form

$$c = 2\log(\lambda_a + \lambda_h) - \log(4\lambda_a\lambda_h)$$

So far we upper bound the probability that adversary mines ℓ blocks faster than honest nodes mines ℓ blocks. Now we upper bound the probability that adversary is ever faster than honest nodes for any $\ell > k$ by using union bounds

$$\begin{aligned}\Pr\left[\text{private chain attack is successful}\right] &= \Pr\left[\exists \ell \geq k : E_\ell \text{ happens}\right] \\ &= \Pr\left[\cup_{\ell=k}^{\infty} E_\ell\right] \\ &\leq \sum_{\ell=k}^{\infty}\Pr\left[E_\ell\right] \quad \text{; union bound} \\ &= \sum_{\ell=k}^{\infty} e^{-c\ell} \\ &= \frac{1}{1 - e^{-c}}e^{-ck}\end{aligned}$$

Once the private chain attack is successful, all honest blocks can be deconfirmed; so safety property no longer holds. For avoiding such attack under $\lambda_h > \lambda_a$, we can increase the confirmation depth k, until the probability of a successful attack becomes extremely small.

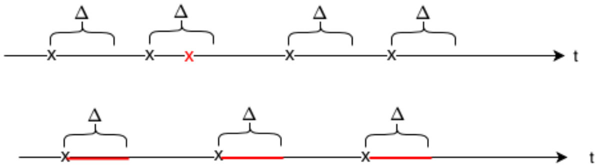

Figure 6.2: Top: Honest mining process with latency, where the red cross is a block excluded from the blockchain due to forking. Bottom: A censored mining process, where the red line is the silencing period

6.3.3 security with network latency Δ

With network latency Δ, two(or more) randomly mined blocks may arrive to the network within the Δ time, and create a situation called **forking** where only one of the blocks is confirmed in the chain. One of the time domain realization is depicted in Fig 6.2. Because two(or more) arrivals within Δ only has one block confirmed in the chain, we model the new process by manually silencing a Δ duration right after a block is mined, so a new block can only be generated after this period, shown in the bottom part of Fig 6.2. To compute the rate for each arrival in the new censored process, we need to consider both the silencing part and the mining part. The silencing part is simply Δ; since the mining event is a memoryless exponential random variable, the distribution for it to mine a block after waiting the Δ period is still exponentially distributed. Hence it can be expressed as (with reuse of notation from previous section)

$$\mathbb{E}\left[T^h\right] = \Delta + \frac{1}{\lambda_h}$$

Hence the rate of growth of the honest chain is adjusted to $\frac{1}{\mathbb{E}\left[T^h\right]} = \frac{\lambda_h}{1+\Delta\lambda_h}$ in the unit of (block per sec). We can view the $\Delta\lambda_h$ as a penalty factor due to forking. Continuing our security analysis for the delayed network, we first need to make sure event E_ℓ does not happen in expectation, which translates to

$$\begin{aligned} \mathbb{E}[T_i^h] &< \mathbb{E}[T_i^a] \\ \Delta + \frac{1}{\lambda_h} &< \frac{1}{\lambda_a} \\ \lambda_a &< \frac{\lambda_h}{1+\Delta\lambda_h} \end{aligned}$$

The security condition for the system is $\lambda_a < \frac{\lambda_h}{1+\Delta\lambda_h}$, and we can apply the similar analysis in the previous section to show the probability of a successful private chain attack deceases exponentially in relation to k if the security condition holds. Taking bitcoin as an example, suppose network latency is $\Delta = 10$ sec, the honest mining rate $\lambda_h = \frac{1}{600sec}$, the new non-wasted honest power is $\frac{\lambda_h}{1+\Delta\lambda_h} = \frac{\lambda_h}{1+10/600} = \frac{\lambda_h}{1.0166}$. Therefore if we use a large k, the system is secure if

$$\lambda_h > 1.0166\lambda_a$$

Another way to describe the relation is via fraction of adversarial power $\beta = \frac{\lambda_a}{\lambda_a+\lambda_h}$. In the bitcoin case, the system is secure with a large k when $\beta < \frac{\lambda_a}{\lambda_a+1.0166\lambda_a} = 0.495$. In the next lecture, we will visit other types of attacks.

References

[1] Hossein Pishro-Nik. counting process. Available at https://www.probabilitycourse.com/chapter11/11_1_1_counting_processes.php.