EE 374: Blockchain Foundations **Stanford, Winter 2023**

Lecture 16: PoS Longest Chain

March 9, 2023

Lecturer: Prof. David Tse

1 Overview

From last lecture, to prevent grinding attacks, we proposed a new PoS equation for validating blocks in the PoS longest chain protocol. Our latest PoS equation for winning a block is as follows:

$$H(s_0||pk||r) < T \tag{1}$$

where $s_0 = H(G)$ is the hash of the genesis block, pk is the public key of the winner of the block, r is the round at which the block is won, and T is the proof-of-stake target.

In this lecture, we will develop the full protocol of PoS longest chain, and discuss the intuitions behind the security theorems of PoS longest chain. Lastly, we will point out the drawback of our protocol, which is being vulnerable to predictive attacks.

2 PoW vs PoS Header Chain

We will review the block structure and proof equations for PoW and PoS in order to have a clearer understanding of the protocol and help us compare the differences when we talk about the security theorems in the next section.

Generally, in PoW, the block header for the ith block B_i in a chain, starting from the genesis, includes the following:

- s_{i-1} : the hash of B_{i-1}, providing the linking to the previous block
- x_i : the Merkle tree root of transactions in block B_i
- ctr_i : the nonce of the block.

The PoW equation which needs to be satisfied for validity is

$$H(s_{i-1}||x_i||ctr_i) < T \tag{2}$$

In our newly devised PoS longest chain protocol, the information included in the header for block B_i is:

- s_{i-1} : the hash of B_{i-1}, providing the linking to the previous block
- x_i : the Merkle tree root of transactions in block B_i
- $pk^{(i)}$: public key of the winner of the i-th block

- r_i : the round in which the i-th block is produced
- $\sigma_i = \text{Sign}(sk^{(i)}, s_{i-1}||x_i||pk^{(i)}||r_i)$: the signature of the owner of $pk^{(i)}$ on the block header

The PoS equation is (1).

3 Security Theorems

Recall that for PoW, the security theorem is as follows

Theorem 3.1 (PoW Security (informal))**.** *Suppose we have n total parties with t of them being dishonest. Then if $t < n/2$, PoW longest chain protocol is safe and live for some choice of value of f and ϵ .*

Let us first state the corresponding security theorem for PoS longest chain (we will see later why the theorem holds).

Theorem 3.2 (PoS Security (informal))**.** *Suppose we have n total parties with t of them being dishonest. Then if $t < n/2$, PoS longest chain is safe and live for some choice of the value of f and ϵ.*

The essence and intuition behind the PoW theorem is that the number of honest blocks is proportional to the honest mining rate, and the number of adversarial blocks is proportional to adversarial mining rate. Therefore, given enough time, if honest majority holds, then there have to be more honest blocks than dishonest blocks with high probability. This fact, in combination with the pairing lemma, is used in the proof of common prefix.

However, notice that this fact is not true anymore in PoS. To see this, consider two PoS blocks B and $\tilde{B}$:

$$B = (s_{i-1}, x, pk, r, \sigma) \text{ and } \tilde{B} = (\tilde{s}_{i-1}, \tilde{x}, pk, r, \tilde{\sigma})$$

Notice that there is no difference between what goes into the PoS equation when checking the validity of B and $\tilde{B}$ (i.e. s_0, pk, and r). This means that the adversary $\mathcal{A}$ can create many different blocks by winning the PoS lottery once at a particular round. We call such blocks *equivocations*. The number of adversarial blocks can therefore be arbitrarily larger than the number of times the adversary wins the PoS lottery, while the number of honest nodes produce is exactly the number of times they win the lottery. This potentially gives the adversary significantly more power.

Here's an example attack that uses this power of the adversary.

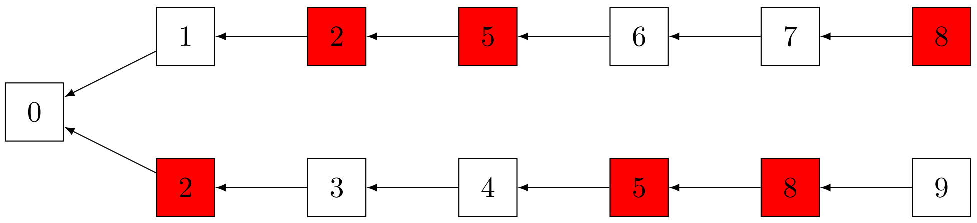

Figure 1: Example attack that uses the adversary's power of equivocation.

In Figure 1, white blocks are honest and red blocks are adversarial, and the number in the block refers to the round r when the block is won.

Assume that the adversary has the ability to choose which previous block honest nodes propose on when there are multiple longest chains (recall the rushing adversary from Lecture 8). The attack works as follows: when an honest node produces a block (1), the adversary matches it with another block (2). Then, when honest node produces a block (3) that extends (2), the adversary can reuse (2) to extend (1). This works because the information of the previous block (s_1) is not included in the PoS equation, so the adversary could choose whichever chain to extend.

With the above example, the adversary is able to break common prefix (let's say $k = 5$) even though the adversary only wins blocks in half the number of slots as the honest nodes (i.e., $t \approx \frac{1}{3}n$.

Note that this attack requires that the adversary wins one block every time the honest nodes win two blocks. What is the probability that this pattern continues for a long time? It is negligible!. So the example attack doesn't contradict Theorem 3.2, since security needs not hold on events of negligible probability. On the other hand, what this attack demonstrates is that the *proof technique* we used to prove PoW security based on purely counting the number of blocks is not sufficient to proof the PoS security Theorem 3.2. New techniques are needed.

4 Everything is a Race & Nakamoto Always Wins

Recall that a Nakamoto race is when the adversary secretly builds a chain, and once the hidden chain is longer than k blocks, and it is longer than the public chain which all honest nodes follows, the adversary publishes it in order to break common prefix.

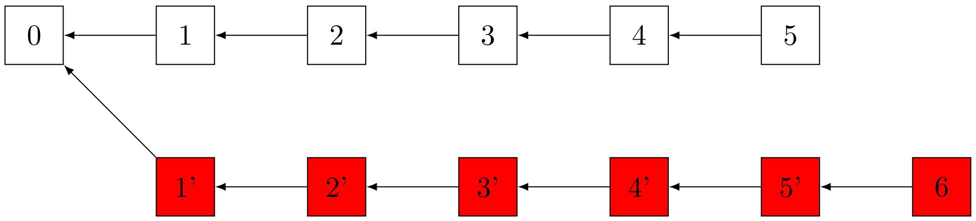

Figure 2: Nakamoto Race

Realize that for this attack, equivocations will not help the adversary because she cannot put two equivocating blocks, which are necessarily in the same round, in sequence (the round indices have to be strictly increasing along any valid chain). With this, we can see a very important insight that equivocating blocks can only be used in parallel chains. Thus, we can try to reduce every attack to some sort of attack that uses a sequence (race) and show that the security theorem still holds. We accomplish that by first partitioning the blocktree under an arbitrarily attacks into adversary subtrees.

As Figure 4 shows, in the blocktree partitioning process, we first build a canonical sequence of honest blocks which we call a virtual honest chain. The blocks in the virtual chain are not necessarily chained together, but only ordered by height. Then for each honest block, we define an adversarial subtree as the tree consisting of all adversarial blocks that are connected to the honest block through only adversarial blocks (or directly). Due to equivocations, each time the adversary wins a block, she can add it to all branches, but cannot add more than once to each branch. This

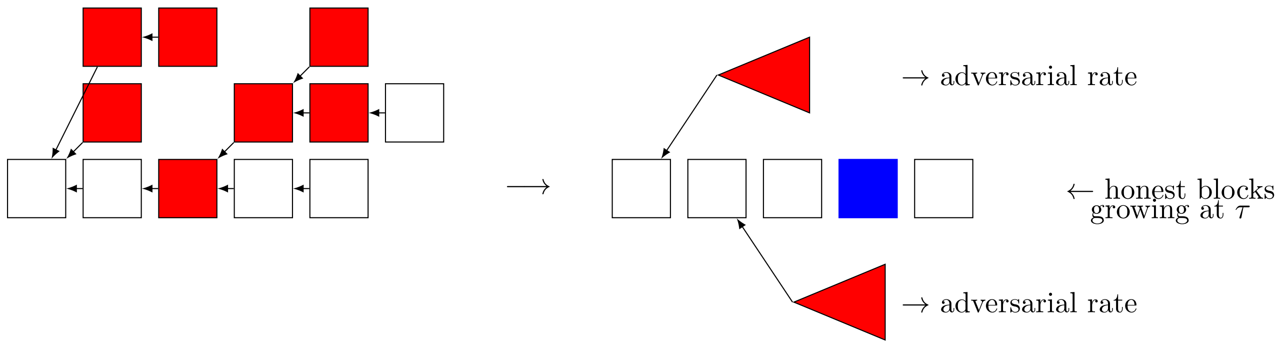

Figure 3: Example blocktree partitioning. Recall that τ is the chain growth parameter. An example Nakamoto block is highlighted in blue.

limits all adversarial subtrees to grow at the rate at which adversarial rounds are won. Based on the Chain Growth Lemma, honest blocks grow at the rate of at least τ. If we have honest majority, then τ is greater than the adversarial rate. Then, each of the adversarial subtrees grows slower than the canonical chain. As a result, we can prove that once in a while, there will be an honest block such that after the arrival of this block, none of the adversarial trees that originate from earlier honest blocks will be able to catch up to the virtual honest chain after this block. We call this type of block *Nakomoto block*. It is further shown that it is guaranteed that Nakamoto blocks will stay in he canonical longest chain forever and that the existence of these blocks suffice to guarantee the safety and liveness of the PoS longest chain protocol. Check the reference [1] for more details about this proof.

5 Drawback of the PoS Security Theorem

During our previous discussion of the security theorem of our current PoS protocol, we have always assumed that we have a static adversary (adversary who chooses the t corrupt nodes at the beginning of the protocol). With our current PoS equation (1), an adversary knows, for r from 0 to infinity, exactly who is going to win each round. Therefore, in the face of an adaptive adversary, who tries to make predictions for the future and corrupt the winning nodes, our protocol becomes highly exploitable since our current protocol is very predictable. This will be solved in the next lecture.

References

[1] A. Dembo, S. Kannan, E. N. Tas, D. Tse, P. Viswanath, X. Wang, and O. Zeitouni. Everything is a race and Nakamoto always wins. In *Conference on Computer and Communications Security*, CCS '20, page 859–878. ACM, 2020.