Zano hybrid PoS/PoW analysis and improvement proposal.

Andrey N Sabelnikov (andre@zano.org)
Special thanks for big contribution to this research to Maxwell Sanchez (www.veriblock.org/)
June 2019
www.zano.org

Cumulative Difficulty Adjustment

As was claimed in original Zano whitepaper, any attack on Zano's hybrid parity protocol, would require an attacker to obtain a majority of both hashing power and a majority of coins. Deep analysing of PoS consensus showed that it needed a few improvements to be resistant to these sorts of attacks.

To enforce this hybrid parity protocol, the core always prefers a chain with alternating work types and we have developed a method of adjusting the weighting of successive PoS or PoW blocks such that the core will always prefer the chain with alternating work.

The following rules were applied:

• Cumulative difficulty (CD) is used in determining which chain is "better".

• The cumulative difficulty for each chain is the weighted sum of each block's difficulties:

$$CD = \sum_{i=0}^{h} w_i d_i$$

where:

w_i — is a weight of particular difficulty of the block;

d_i — difficulty for this height in the current chain.

• Weights are chosen to favor a chain in which blocks alternate in type: $PoS - PoW - PoS - PoW$ etc. In this ideal case $w_i = 1$ for each i.

• Weights decrease at rate $r = 0.75$ in cases of successive blocks of the same type.

For example, a chain PoS - PoW - PoW - PoW - PoS - PoS has the following cumulative difficulty:

$$CD = (d_1 + d_2 + r \cdot d_3 + r^2 \cdot d_4 + d_5 + r \cdot d_6)$$

These rules may lead to a situation where a 10 block-long chain will have less cumulative difficulty than an 8 block-long one. This may be surprising, but it derives from the assumption that an attacker with only one overwhelming type.

Problem of linking PoS/PoW difficulties.

The nature of PoW difficulty and PoS difficulty is completely different. PoW difficulty reflects the estimated number of hashing operations needed to reach target. PoS difficulty reflects the amount of coins involved at a particular moment in PoS-mining ("staking"). Each type of difficulty is controlled independently, and to ensure that the network will choose a chain with greater efforts from both sides (PoS and PoW) we need to establish an algorithm which evaluates summary efforts in a proper way.

The problem is that balance between PoW and PoS difficulty is constantly changing and primitive algorithms doesn't provide correct protection against manipulation.

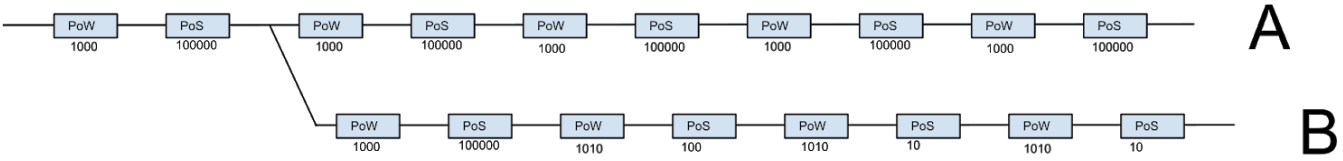
First Zano implementation calculated difficulty adjustment by multiplying current PoS difficulty to the difficulty of the last PoW block divided to current PoS difficulty. This brought PoS difficulty to PoW basis:

$$d_i^{adj} = d_i^{pow}$$

$$d_i^{adj} = d_i^{pos} * \left(\frac{d_{prev}^{pow}}{d_i^{pos}} \right)$$

Researching different attack vectors showed that with this approach it's still possible to fulfill double spend attack by only by having 51% PoW certain amount of PoS (depends on how far should attacker to go backward to perform attack: less coins -> longer period to have longer time frame for timestamp enumeration -> more expensive attack).

Consider these two sequences as a very simplified example:



In this example PoW and PoS blocks alternate sequentially and the sequence factor is always equal to 1.

Cumulative difficulty for block i is:

$$CD_i = CD_{i-1} + Sf * d_i^{adj}$$

where:

CD_{i-1} — cumulative difficulty calculated on height i - 1

Sf — "sequence factor", coefficient which set penalty for a block as:

$$Sf = r^{n-1}$$

where n is "sequence number", number of blocks of the same type goes in a row, and r is sequence decrease rate (assume r = 0.75).

Using this formula for chain "A", we obtain:

$$CD_A = 1000 + (100000 \cdot \tfrac{1000}{100000}) + 1000 + (100000 \cdot \tfrac{1000}{100000}) + 1000 + (100000 \cdot \tfrac{1000}{100000}) + 1000 + (100000 \cdot \tfrac{1000}{100000}) + 1000 + (100000 \cdot \tfrac{1000}{100000}) = 10000$$

For chain "B" we have:

$$CD_B = 1000 + (100000 \cdot \tfrac{1010}{100000}) + 1000 + (100000 \cdot \tfrac{1000}{100000}) + 1010 + (100 \cdot \tfrac{1010}{100}) + 1010 + (10 \cdot \tfrac{1010}{10}) + 1010 + (10 \cdot \tfrac{1010}{10}) = 10060$$

It's clear that $CD_B > CD_A$, and, moreover CD_B doesn't take into account the amount of PoS miners' contribution. This type of PoS "blind" binding to PoW lets PoW miner to drop PoS difficulty in alternative chain and then adjusted difficulty formula will evaluate PoS work as nearly equal to mainchain. Hard part for an attacker here is to generate first PoS blocks in chain before PoS difficulty got dropped. It can be done by using a longer timeframe which starts from the past to enumerate more timestamps for every coin in the stake, which then makes it

more expensive to catch up with the main chain. Or it can be done by re-mining PoW blocks until staking coins win a chance to make a block. Both methods depends on how much of a stake an attacker has.

As it turned out, the problem is in the way we get d_i^{adj} - this parameter should be protected from manipulation. If we consider the definition of d_i^{adj} as some sort of average of long period, let's say median of monthly PoW and PoS difficulties ratio, then this only makes an attack more expensive - an attacker would need more efforts to handle the gap between the top of the main chain and some point in the middle of 30 days, needed to shift an adjustment median. But still, the statement that "attacker needs 51% + some minor stake" is true.

Proposal

The idea of this proposal is to put into analysis only a work which is made on the distance of split and for evaluation of PoS work to take ratio of PoW difficulty and PoS difficulty at the point of split between two competing chains. With this approach adjusted cumulative difficulty exists only for given sub-chain started from the point of split. Let's define adjustment coefficient at point of split K' as:

$$K' = \frac{d_i^{pow}}{d_i^{pos}}$$

where:

d_i^{pow} — current PoW difficulty at the point of split

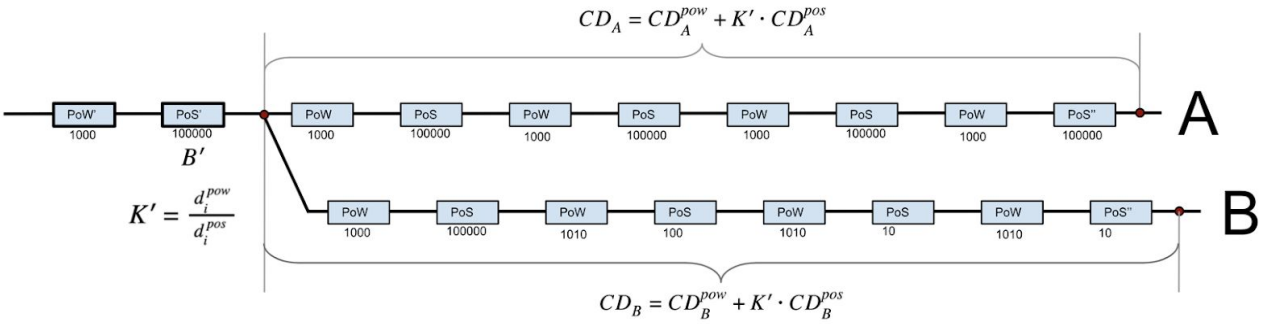
d_i^{pos} — current PoS difficulty at the point of split

Then cumulative difficulty for given sub-chain S split from the main chain after block B' is:

$$CD_s = CD_s^{pow} + K' \cdot CD_s^{pos}$$

where $CD_s^{pow} = \sum_i d_i^{pow}$ and $CD_s^{pos} = \sum_j d_j^{pos}$

Illustration provided down below:



For the given example cumulative difficulties for both chains would be calculated as the following:

$$CD_A^{pow} = 1000 + 1000 + 1000 + 1000 = 4000$$
$$CD_A^{pos} = 100000 + 100000 + 100000 + 100000 = 400000$$
$$CD_A^{adj} = 4000 + 400000 \cdot K' = 4000 + 400000 \cdot (\frac{1000}{100000}) = 8000$$

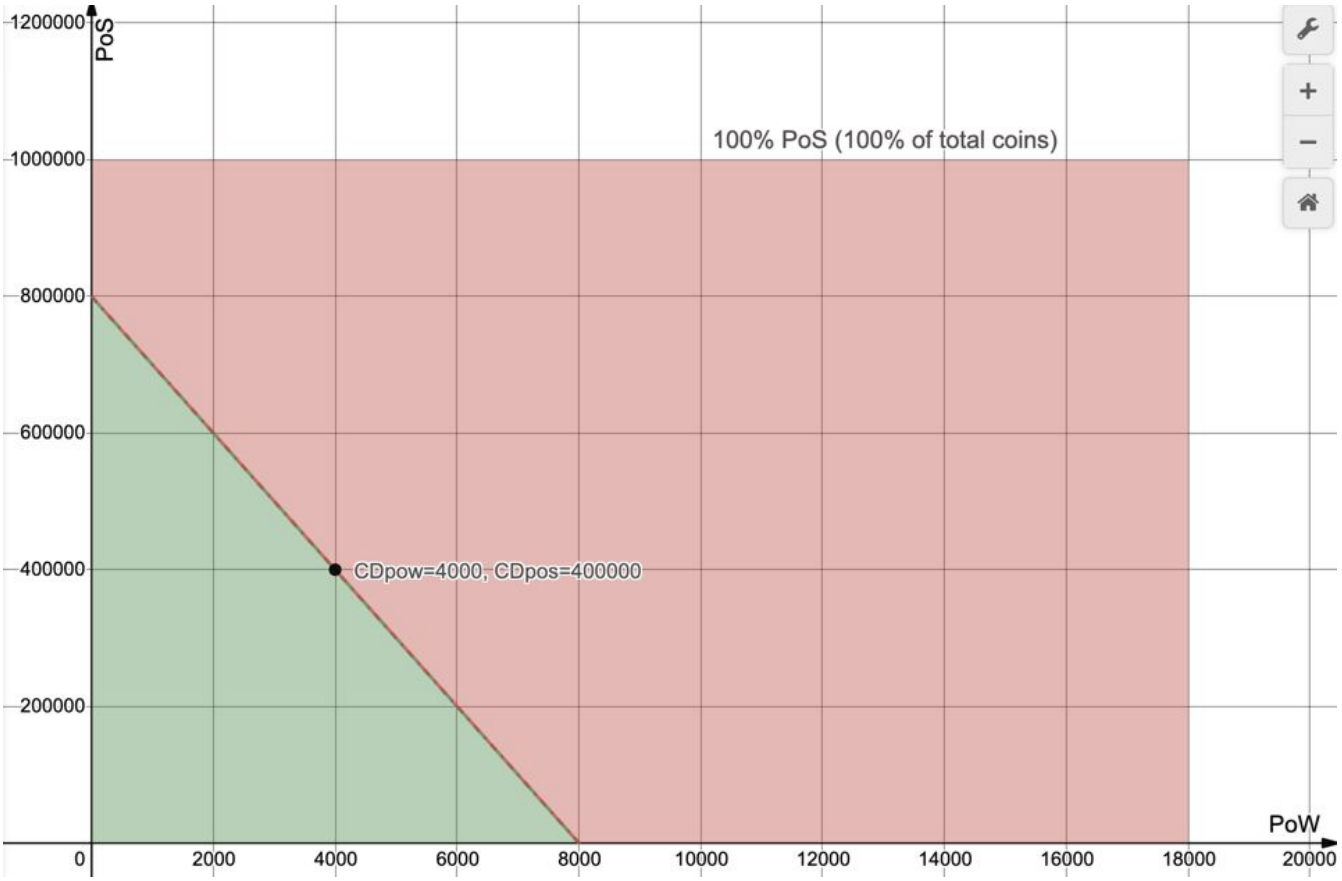
For chain "B" against "A" we are getting:

$$CD_B^{pow} = 1000 + 1010 + 1010 + 1010 = 4030$$
$$CD_B^{pos} = 100000 + 100 + 10 + 10 = 100120$$
$$CD_B^{adj} = 4030 + 100120 \cdot K' = 4000 + 100120 \cdot (\frac{1000}{100000}) = 5001$$

CD_A^{adj} now properly reflects PoS contribution and looks way greater than CD_B^{adj}. Let's try to see how much resources would be needed to commit double spend attack with CD_s formula. If we do analysis based on numbers from the example above, then we can describe it with this equation:

$$8000 = CD_{PoS} + \frac{1}{100} \cdot CD_{PoW}$$

With the graph presented above you can see the red zone which represent an area of potential double spends and the green area is safe. In this graph we assumed that amount of coins used in PoS mining in chain A was only 40% of total coins in circulation, just to show that PoS "hash power" is a limited resource.



This graph shows that if a potential attacker has nearly 100% of one of the type of resources (PoW or PoS) and minority of other - then he can commit double spend attack by creating alternative chain with superior CD_s. In order to make the formula more resistant to the manipulation of difficulty through only one of the resources type(PoW or PoS) to the detriment of the other, let's introduce two balancing coefficients for the PoW and PoS and respectively, which will take into account the ratio of cumulative difficulties on two alternative chains. In this case, total cumulative difficulty can only be calculated relative to another alternative chain, but cannot exist by itself.

$$K' = \frac{d_i^{pow}}{d_i^{pos}} \;;$$

$$K_{PoW(X \to Y)} = \frac{CD_{PoW(X)}}{CD_{PoW(Y)}}$$

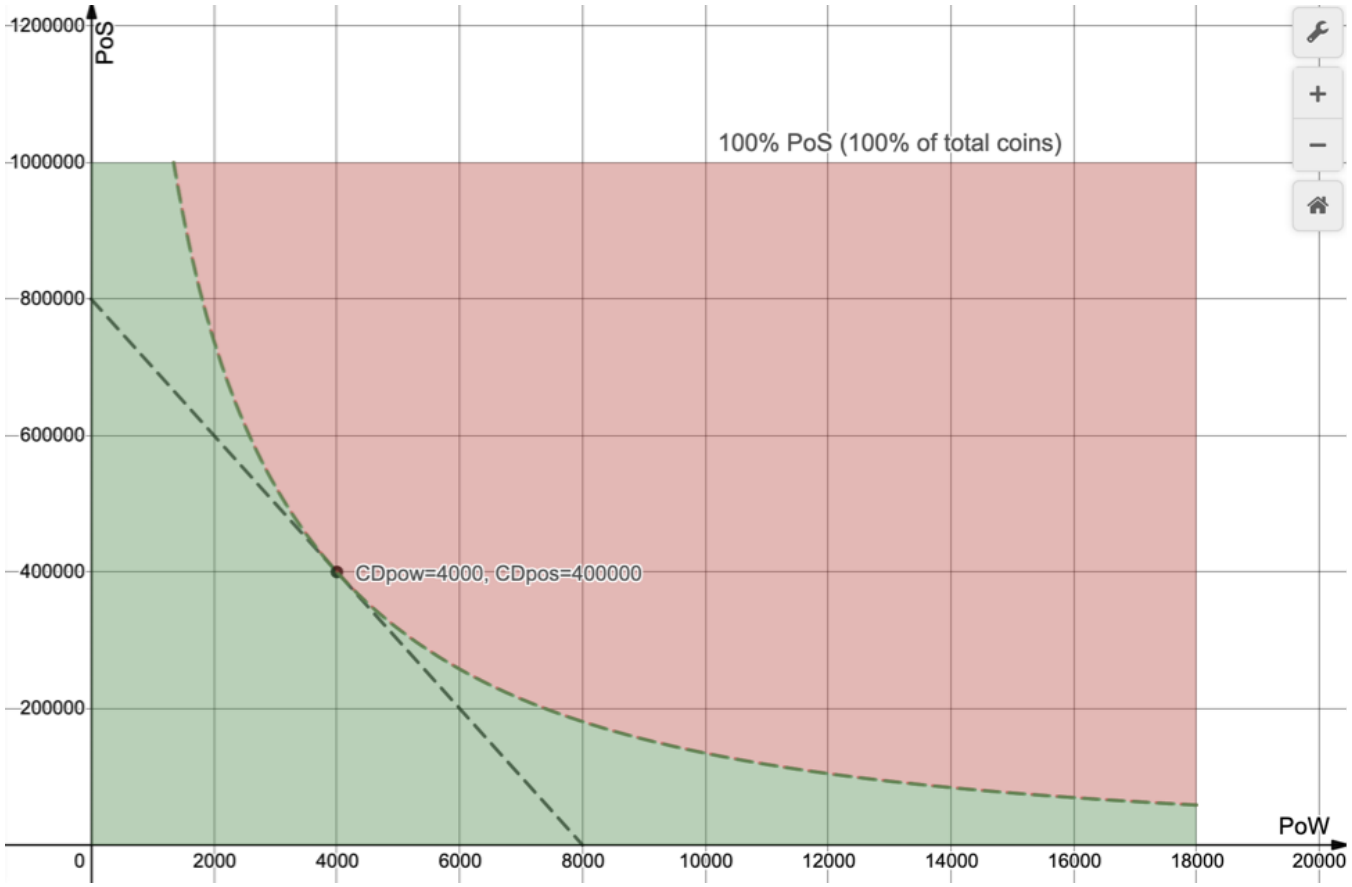
$$K_{PoS(X \to Y)} = \frac{CD_{PoS(X)}}{CD_{PoS(Y)}}$$

$$\begin{cases} CD_{A \to B} = K_{PoW(A \to B)} \cdot K_{PoS(A \to B)} \cdot \left(CD_{PoW(A)} + CD_{PoS(A)} \cdot K' \right) \\ CD_{B \to A} = K_{PoW(B \to A)} \cdot K_{PoS(B \to A)} \cdot \left(CD_{PoW(B)} + CD_{PoS(B)} \cdot K' \right) \end{cases}$$

Now, to see resulting curve which describe boundary of potential “double spend” attack, based on the same numbers as in the example above, we can use both sides of the system of equations in the following inequality:

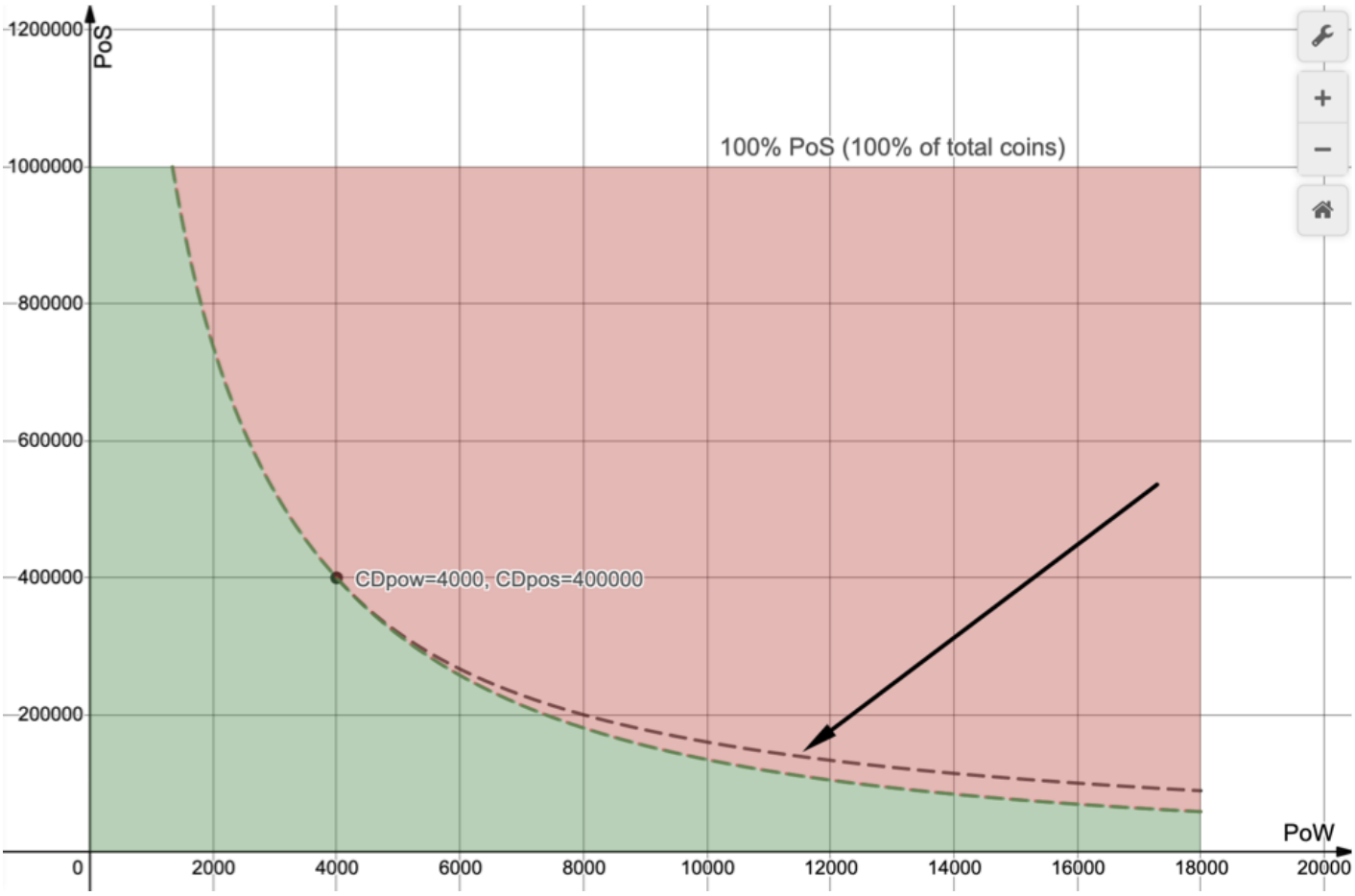
$$K_{PoW(A \to B)} \cdot K_{PoS(A \to B)} \cdot \left(CD_{PoW(A)} + CD_{PoS(A)} \cdot K' \right) < K_{PoW(B \to A)} \cdot K_{PoS(B \to A)} \cdot \left(CD_{PoW(B)} + CD_{PoS(B)} \cdot K' \right)$$

$$K_{PoW(A \to B)} \cdot K_{PoS(A \to B)} \cdot \left(4000 + 400000 \cdot \frac{1}{100} \right) < K_{PoW(B \to A)} \cdot K_{PoS(B \to A)} \cdot \left(CD_{PoW(B)} + CD_{PoS(B)} \cdot \frac{1}{100} \right)$$



PoS power multiplication through PoW power problem

Let's assume PoW miner has majority of hashrate and can use his/her hashpower to let him/her multiply his chances to find PoS block by not announcing all mined PoW blocks but announcing only those which let him/her win PoS blocks. In this case having 100% hashrate (twice more than needed for performing double spend in pure PoW system) let him do double spend by having only 25% of PoS power, with 200% needed only 12.5% of PoS power and so on. Nevertheless, such manipulations do not create the possibility to go beyond the curve defined by the formula of relative complexity:



Conclusions

In our opinion, the proposed solution creates a stable equilibrium between PoW and PoS, ensuring decentralization and high level of resistance to 51% attacks.

Graphs and formulas can be checked at this link:
https://www.desmos.com/calculator/nhzf7mbtes