NuCypher KMS: Mining

Michael Egorov*

NuCypher

(Dated: April 13, 2018)

This paper describes mining mechanisms and economics in NuCypher KMS. It includes inflation rates, mechanisms to incentivise long-term stakers and estimates of the number of tokens generated by nodes running in typical modes. Also, optimal strategies for stakers who may be affected by market volatility are proposed.

I. MOTIVATION

In the future, NuCypher KMS will probably be fully paid by network fees. But initially, when the adoption isn't yet high, miners who run the nodes necessary for network operation and keep re-encryption keys, will need to be subsidised. This will be done through an inflation schedule, where all the inflation is given back to miners.

Distribution of rewards should have the following properties:

- All the inflation is distributed to stakers who run the nodes, proportionally to their stake;
- The amount of work (and, hence, fees earned) is proportional to stake;
- Stakers are incentivized (by a higher reward rate) to run long-term nodes;
- High inflation doesn't depreciate the price in order to keep liquidity for new stakers;
- Stakers are incentivized to stay online all the time.

In the paper we address all these points, calculate expected earnings of miners who run nodes and devise optimal mining strategies.

II. HISTORICAL EXAMPLES OF INFLATION

Let's review inflation schedules of different cryptocurrency projects: DASH [1] and ZCash [2].

FIG. 1: Historical price of ZCash in logarithmic scale. Note the minimum at 23 Feb 2017

DASH has a hybrid of Proof-of-Work (POW) and Proof-of-Stake (POS). It has 45% of inflation going to POW miners, 45% to staking master nodes, and 10% reserved for budget proposals [3]. After the first year, its emission was 18.42% APR, decreasing by 1/14 every 383 days. With this setting, 60% of DASH coins are locked in masternodes for staking, according to the node statistics. It's unclear how inflation rate affects the price (and if it does here), but the useful data point is that there are 60% of coins locked for staking. Perhaps, that is something to expect in a network where staking is an option.

ZCash is very interesting because it started with an extremely high inflation rate. This caused a short-term price drop (even though the market capitalization was growing) (Fig. 1). But at some point (23 Feb 2017), the price started going up. ZCash block rewards yield 50 ZEC every 10 min, and ZEC supply at Feb 23 was

* michael@nucypher.com

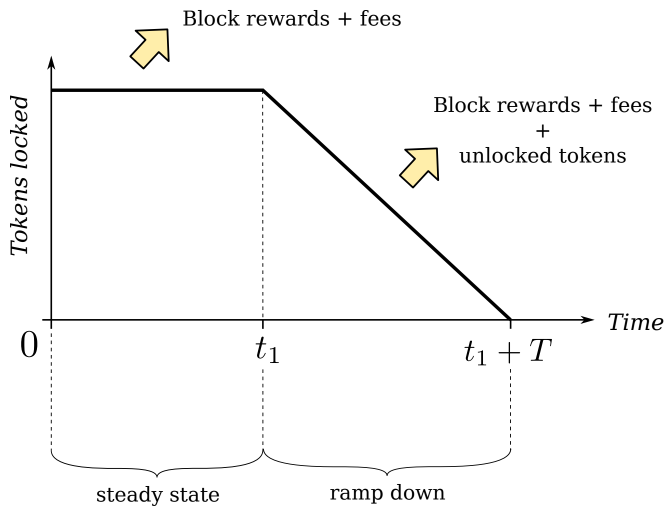

FIG. 2: Mining modes: *steady state* (before t_1) and *ramp down* (from t_1 till $t_1 + T$). The ramp down (or vesting) time T is chosen by the miner (and he gets higher staking rewards when T is higher).

727k ZEC. This corresponds to 360% APR. It is even more remarkable given the fact that miners who mined ZEC are probably those who dump and exchange the proceeds into something else (and also pay electricity bills). This gives us information about what would be the maximum allowable inflation which still doesn't create too much downward pressure on the price.

III. MINING PROTOCOL

A miner commits to stay available for at least time T. If the tokens were locked for this whole time T and abruptly unlocked after that, we'd have a problem: a huge increase of tokens in circulation right after the time T that crashes the market.

In order to avoid that, we introduce two modes of miner operation: *steady state* and *ramp down*. When miner is in steady state, his base tokens stay locked. Block rewards are created, and they aren't necessarily locked (unless the miner chooses to add newly minted tokens to the locked amount).

At any point in time, the miner can switch to *ramp down*. This would start unlocking the staked tokens linearly, over time T. Both, *steady state* mode and *ramp down* starting at the point t_1 are shown at (Fig. 2). So, the time T doesn't mean the total time during which miner's tokens are locked, but rather how quickly do the tokens get unlocked once he starts unlocking.

IV. GENERAL INFLATION PROPERTIES

A. Initial inflation

Let's assume that we'll have the same number of tokens locked as DASH: $\lambda = 60\%$. Then we'll have $1 - \lambda = 40\%$ in circulation. If we have inflation rate I, then adjusted inflation rate (e.g. inflation as if the locked tokens didn't exist) of tokens in circulation will be:

$$I^* = \frac{I}{1 - \lambda}, \tag{1}$$

and we should be comparing I^* with historical examples of inflation. If we take $I^* = 350\%$ (turnover point of ZCash price in an overall bullish market), the corresponding inflation I will be 140% APR.

To be on the safe side, we choose the starting inflation to be $I_0 = 100\%$ APR (or, in other words, 1/365 per day).

B. Inflation decay

Initially, the inflation subsidises mining, but payments for re-encryption services will generate the majority or all the revenues of miners in the long run. If all miners have the same, maximum reward rate, we choose the inflation rate to decay by factor of 2 in $T_{1/2} = 2$ years. The inflation, depending on time passed from the Genesis t, looks like:

$$I(t) = I_0 \cdot 2^{-\frac{t}{T_{1/2}}} = I_0 \exp\left[-\ln 2 \frac{t}{T_{1/2}}\right]. \tag{2}$$

In this case, the dependence of the token supply on the time t is:

$$S(t) = S_0 + \int_0^t I(t)\, dt = S_0 + \frac{I_0 T_{1/2}}{\ln 2}\left[1 - 2^{-\frac{t}{T_{1/2}}}\right], \tag{3}$$

Let's call relative initial annual inflation i_0, and then $I_0 = i_0 S_0$. For 100% APR, $i_0 = 1$ and $I_0 = S_0$ per year, and the maximum number of tokens which will ever be created is:

$$S_{\max} = S(\infty) = S_0 \left(1 + \frac{i_0 T_{1/2}}{\ln 2}\right) \approx 3.89\, S_0, \quad (4)$$

where S_0 is initial number of tokens.

C. Implementation of the exponential decay in a smart contract

Complex functions like exponentials, if implemented in smart contracts, would be quite costly. Fortunately, the exponential is a solution of a differential equation where inflation is proportional to the amount of not yet mined tokens:

$$I(t) = \frac{\ln 2}{T_{1/2}} \left(S_{\max} - S(t)\right) \quad (5)$$

$$dS = I(t)\, dt, \quad (6)$$

where $S(t)$ is the current token supply with $S(0) = S_0$ and the time step dt can actually be equal to the mining period (1 day). Each mining node can trivially calculate its dS in a smart contract using very few operations and the coin supply S from the last period. So, the amount of tokens mined for the node i and the time period t will be:

$$ds_{i,t} = \frac{l_i}{L} \frac{\ln 2}{T_{1/2}} \left(S_{\max} - S_{t-1}\right), \quad (7)$$

$$dS_t = \sum_i ds_{i,t}, \quad (8)$$

where l_i is the number of tokens locked by the miner i, L is the total number of tokens locked. Instead of calculating all the sum over i, each miner i can add her portion $ds_{i,t}$.

D. Mining rate and staking time

We want to incentivize miners to keep serving re-encryption policies for at least 1 year. However, short-term stakers are still useful and should be rewarded. We will give the full reward ($\kappa = 1$) to the stakers who are committed to stake at least $T_1 = 1$ year, however those who stake for $T_{\min} = 1$ month will get close to half the reward ($\kappa \approx 0.54$). The individual daily reward rate for a miner looks as:

$$\kappa = \left(0.5 + 0.5 \frac{\min(T_i, T_1)}{T_1}\right) \quad (9)$$

$$T_{i,\text{initial}} \geq T_{\min}, \quad (10)$$

$$\delta s_{i,t} = \kappa \frac{l_i}{L} \frac{\ln 2}{T_{1/2}} \left(S_{\max} - S_{t-1}\right). \quad (11)$$

$$(12)$$

The unlocking time T_i means either the unlocking time if the miner is in steady state mode, or the time left to unlock the tokens if the miner is in ramp down mode. The initial T_i cannot be smaller than 1 month, but it can become smaller than that during the ramp down.

This has implications on the global token economy. Firstly, if stakers, despite smaller reward, prefer to stake for shorter, that creates smaller daily token supply. Since this usually should coincide with bear market, reducing the issuance during that time sounds like a good idea.

Interestingly enough, $\kappa < 1$ prolongs the reward half-decay time $T^*_{1/2} = T_{1/2}/\kappa^*$, where κ^* is the mean staking parameter. If all the stakers have $\kappa^* = \kappa = 0.5$, this prolongs $T_{1/2}$ to be 4 years instead of 2.

The total supply over time (Eq. 3) at $\kappa^* \neq 1$ will then look like:

$$S(t) = S_0 \left[1 + \frac{i_0 \kappa^* T^*_{1/2}}{\ln 2} \left(1 - 2^{-\frac{t}{T^*_{1/2}}}\right)\right]. \quad (13)$$

V. MINING STRATEGIES AND EXPECTED REWARDS

In this section, we look at three possibilities: miner taking all the block rewards, miner adding all the rewards to what's currently staked and miner spinning the node down to get all the tokens unlocked after the time T (as well as the rewards). Every of these possibilities could have different distributions of κ. Let's consider $\kappa = 1$ and $\kappa = 0.5$ as two marginal values. Let's take amount of tokens locked to be $\lambda = 60\%$, as in DASH. We'll plot graphs of daily rewards, as well as calculate the rewards during the first year in each of these scenarios.

A. Withdraw mining compensation

This is the simplest case. The total amount of tokens staked in the network can be expressed as $L = \lambda S$. The amount of stake stays constant in this case and equal to $s_i = l$, and the mining rate (e.g. the cumulative rewards) is:

$$\frac{dr}{dt} = \kappa \frac{l}{\lambda S(t)} \frac{\ln 2}{T_{1/2}} \left(S_{\max} - S(t)\right). \tag{14}$$

When we substitute $S(t)$ from Eq. 13 and integrate over time, we find total rewards:

$$r(t) = l \frac{\kappa}{\kappa^* \lambda} \ln \frac{S(t)}{S_0}. \tag{15}$$

If $\kappa = 1$ (staking for 1 year+) and $\lambda = 60\%$ (60% of all nodes in the network are staking), miner's compensation starts from 0.46% per day in NKMS tokens, or 100.2% during the first year of staking.

We should note that if other miners stake for less than a year ($\kappa^* < 1$), the inflation rate decays slower, and the compensation over a given period will be higher.

B. Restake mining compensation

Instead of taking all the mining compensation, it could be restaked into the node in order to get even more jobs and, provided that the hardware is maintained powerful enough to cope with those, get even higher compensation. In this case, the actual stake l is constantly increasing with time:

$$\frac{dl}{dt} = \kappa \frac{l}{\lambda S(t)} \frac{\ln 2}{T_{1/2}} \left(S_{\max} - S(t)\right). \tag{16}$$

If we substitute $S(t)$ from Eq. 13 and solve this differential equation against l, we get:

$$l(t) = l(0) \left[\frac{S(t)}{S_0}\right]^{\frac{\kappa}{\kappa^* \lambda}}. \tag{17}$$

If $\kappa = 1$ (staking for 1 year+) and $\lambda = 60\%$ (60% of all nodes in the network are staking), miner's compensation starts from 0.46% per day in NKMS tokens, or $l(1) - l(0) = 177.5\%$ during the first year of staking.

C. Take mining rewards and spindown

When the node spins down, its rewards are constantly decreased because the time left to unlock becomes smaller and smaller, effectively decreasing κ gradually towards 0.5, as well as the stake is linearly decreased towards zero.

VI. EDGE CASES: CONNECTION PROBLEMS, VESTING DURING UNLOCKING

If the miner is found to be non-operational and/or cannot confirm the activity, the tokens aren't unlocked during the mining period when that happens (and the rewards aren't mined). It's not too bad for the miner if that happens, however one cannot lock tokens for mining and get them unlocked, even without getting staking rewards. So if the miner commits to stake for at least a year, it implies a year of operation.

Another question is what happens if during the spindown time, the miner decides to add some tokens to his stake. Once spindown has started, the absolute spindown rate dl/dt is specified. Adding tokens to the stake during spindown doesn't change this rate, unless keeping the rate constant makes the unlocking time larger than T: in this case, the spindown rate is increased. In other words, if the staker adds to the stake during the spindown, the initial unlocking rate dl/dt and the rate after adding to the stake $(dl/dt)'$ are:

$$\frac{dl}{dt} = \frac{l_0}{T}, \tag{18}$$

$$\left(\frac{dl}{dt}\right)' = \min\left(\frac{dl}{dt}, \frac{l}{T}\right); \tag{19}$$

where l_0 — initial stake, l — stake which is still locked, T — unlocking time.

VII. TLDR

A. How much will I be earnining if I run a node

It depends on how early you start (the earlier, the better), and for how long you commit to provide re-encryption services. If you commit to work for 1 month, you'll be getting approximately 54% of what you'll be getting if you commit for 1 year or more. Also the compensation is inversely proportional to the total amount of tokens staked by all the participants. Finally, if you choose to automatically add the tokens earned to the stake, this will increase the rewards because your stake will be increasing as well.

For example, if 60% of tokens in the system are always staked, you'd be earning 0.46% in day 1. With this amount of stakers, if you withdraw all the earned tokens instead of re-staking them, in the first year you'd earn 100.2% of your stake. But if you restake all your tokens, in the same first year you'd earn 177.5% of the tokens you staked.

B. How many tokens will be in the existence

We'll start with 1 billion tokens, and the maximum amount of tokens ever mined will be 3.89 billions.

C. What's the inflation rate

The inflation rate will depend on how many short-term miners and long-term miners are in the system. Depending on this, the initial inflation will be between 50% APR (when all miners are very short term) and 100% APR (when everyone commits for a long term). The inflation will be decaying exponentially every day, halving in time between 2 years (when all the miners are long term) and 4 years (when all the miners are short term).

---

[1] Evan Duffield and Daniel Diaz, "Dash: A privacy-centric crypto-currency," (2015).
[2] "Zcash documentation," .
[3] "Official dash documentation: Emission rate," .