KOREPoS Official Whitepaper

Revision 1.1

Abstract

A Bitcoin/PIVX fork focused on privacy and anonymity that uses a newly created Proof-of-Stake algorithm to generate new coins for the network. This solution also implements full validation offering the best security model a blockchain can implement.

Introduction

Security and anonymity has been Kore's focus. Therefore Kore has built the best model to assure people are safe using the network. People nowadays are concerned about their privacy and data safety. Many governments are imposing sanctions and restrictions on individual freedom. The KORE team is trying to deliver solutions to empower everyone through information technology.

All information traffic through nodes utilizes the Tor network. The onion router requires three layers of cryptographic encryption and a relay circuit that transports data throughout the network. The relays classification is; guard (entrance), middle and exit peer. Each relay decrypts a layer of encryption to reveal the next relay in the circuit, which passes the remaining encrypted data onto the next relay. This makes the data tamper proof and reliable, as well as safe. Using this model, anyone who tries to get the data will be stopped at an encryption wall.

The Tor Network has been proven the best security model to route application data throughout networks. The Kore network is built inside Tor using Socks5 as proxy (Pluggable transports are configurable such as OBFS4). Therefore, all requests and responses are relayed and encrypted, without leaking your IP address which makes it anonymous and safe. As long as the Tor network is secure, Kore's applications will be secure as well.

Kore Proof-of-Stake Algorithm

A problem that many traditional Proof-of-Work blockchains try to solve is the amount of energy spent by the mining process and the need of specialized equipment. There are a few technologies that are trying to solve this. However, none of them seem to take every problem related to a decentralized consensus into consideration. Nor do they seem to consider the lack of security when there's no work involved in the process of creating new coins.
We analyzed many so called Proof-of-Stake coins and did not find any point in the code that proved an actual stake of coins, nor was there a check for those supposed staked coins. It is worth noting that at the time this document was written, the only Proof-of-Stake protocol that people believe will work as it should, is still under development by the developers of the Ethereum Network, Vitalik Buterin et al. KORE developers believe KOREPoS successfully mitigates these issues.

Behavior

New coins are generated by full-nodes that are set-up and running on the network. These nodes must have a minimum amount of coins to be able to mint new coins. This last requisite is due to the economic incentives related to supporting the network and will be discussed later in this document.
Existing coins are used in the minting process and to be considered to the task, they should follow these requirements:

- A *coin*[1] must have a minimum *coin age*[2] of 25 blocks;

[1] A coin is an output of a transaction. A coin can have any value from a minimum amount of 0.00000001.
[2] The coin age means the number of blocks that has been added since this coin was inserted in a block.

- The coin must be unspent and spendable according to the network rules;
- The coin must not be multi-sig, even if the full-node controls all the keys;
- The coin must be *P2PKH*[3];

Minting coins need a balance of at least 25 KORE, the balance must be in the same address to be considered for a stake. After the full-node has a list of which coins could be used in the minting process, it must check if any of them fulfills a last requirement regarding its hash. The formula used to perform this check is discussed in depth at a later point in this document.
Between the selected eligible coins, the first one to fulfill all requirements (kernel) is used to generate new ones.
The block containing the newly minted coins and the staked coins have the following format:

The first transaction contains the *Coinbase*[4], which has 1 input and 3 outputs:

- Inputs:
  - Block height followed by a zero and extra nonce;
- Outputs:
  - Dev fund reward;
  - Minter reward;

The second transaction, referred by as the Locking TX, locks the coins used to prove you had balance at moment of the minting and has the following format:

- Input:
  - *Kernel*[5] transaction;
  - Any other transaction that belongs to the same address (up to 25 transactions)
- Output:
  - Locked coins (up to 2500 KORE)
  - Change in case excess balance of the inputs address is above the locked 2500 KORE

The coins, up to 2500 KORE, are locked for 2 hours(ish). This means that this coin cannot be spent or staked again until this time lock expires. This is guaranteed by Script Rules which uses a sequence timelock that is calculated based on median time past. Meaning, unless the time has passed it can neither be spent nor staked.
It is worth mentioning that any other coin that passes the first set of requirements and belongs to the same address will be *merged*[6].

Kernel

To be considered a Kernel the coin must be part of an address that has at least 25 KORE as balance and hashed as follows:

$$StakeModifier + OriginBlockTime + CoinUniqueID + TransactionTime$$

The Stake Modifier is a big number represented in hexadecimal form.

[3] Pay to Public Key Hash means the address of this payment must be verifiable.
[4] The first transaction of a new block, that contains minted coins.
[5] Kernel is an input that satisfies the rules to mint new coins.
[6] Merge coins by using them as inputs for the transaction, and only one output to the same address.

Origin block time represents the timestamp of the block where this coin was inserted in the blockchain. The coin unique id is the union of the coin hash (txid) and its position on its parent transaction. Transaction time is the current time when there is an attempt to create a stake.

Stake Modifier

The purpose of the stake modifier is to prevent a coin owner from computing future proof-of-stake from being generated by this block of coin during the time the transaction is being confirmed. To meet kernel protocol, the coin must hash with a future stake modifier to generate the proof. The stake modifier consists of bits, each of which is contributed from a selected block of a given block group in the past. The selection of a block is based on a hash of the block's proof-hash and the previous stake modifier. The stake modifier is recomputed at a fixed time interval instead of every block. This makes it more difficult for an attacker to gain control of additional bits in the stake modifier, even after generating a chain of blocks.

Full Validation

We check every part of the block and base transactions creation when we receive a new block from a peer, this way we can validate if any consensus rule has been modified. These are the rules being checked now:

- Amount of generated coins (coinbase);
- Amount of coins rewarded to Dev Fund;
- Amount of coins rewarded to the minter;
- Total reward sum is correct (including tx fees);
- Coins have been locked;
- The total input amount of the staking transaction is equal to its outputs;

Economics

Given that the old KORE blockchain has been running for a few years before the changes discussed in this document are applied, it is difficult to address the interest rate problem in such a manner that allows users with a low balance to earn a fair amount of new coins each time they are able to mint as well as users that have a high balance (thousands of coins) to earn accordingly.
pow(1.44e14 - pow(moneySupplyFloat, 2), (double)1 / 2) / 1.436e7;

Monetary Policy

New KORE is generated for the network using the following formula:

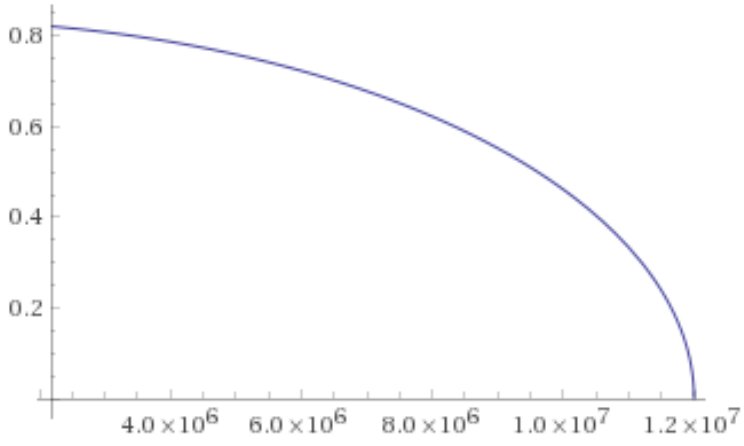
$$coinbase = \frac{\sqrt{1.44 * 10^{14} - MoneySupply^2}}{1.436 * 10^7}$$

Equation will be discussed in depth at a later point in this document.

The network monetary base is composed of exactly 12,000,000.00000000 KORE. No coins will ever be minted above this limit. We have calculated the final minting reward will occur in 37 to 43 years based on block time fluctuation.

Inflation

Any economic system must control the coin inflation, so the current market value does not floor due to too many coins being introduced in a short period of time.
To control the inflation Bitcoin uses a halving system that cuts the amount of generated coins per block by half every 210,000 blocks. This gives Bitcoin a fixed inflation rate based on an issuance schedule.
Taking into consideration that for the past few years, with the old PoS, inflation was not regulated since the coins generated were proportional to the balance of the minter. Our new KOREPoS monetary base limit, to start, has an inflation rate of 16.00% per year that decreases over time, based on the current money supply. The following chart describes the curve over time.



Reward vs Supply

Dev Funding

People might want to develop some interesting use or idea for KORE, this is why we save 10% of the generated coins for those projects. We might also have several projects competing for this fund, our roadmap will explain what this projects are.
The formula to get the developer fund value is very simple:

$$devfund = coinbase * \frac{10}{100}$$

Incentive

We currently have one incentive for the full-nodes that maintain in-wallet balance and are staking, we are working on a second incentive for people that help our services be up and running.

Full-nodes

Full nodes are responsible for validating blocks and creating new coins, if staking is enabled (must follow the rules) through the Coinbase transaction and the locking transaction. Because of that, we feel the need to give a fair incentive for the full-node wallets that keep at least the minimum balance.
The formula used to get how much of the Coinbase distribution will be given to the minter is as follows:

$$minter = (coinbase - devfund)$$

Limits

To allow every user on the network to participate in a fair way, we will limit the minimum amount of coin needed to mint, as well as the maximum amount that could be locked for each minting operation. The bottom limit of 25 KORE is what is needed to create a stake transaction and the upper limit is 100x the minimum required fund.

Chances

To balance the reward structure people with balances just above the lower limit and those above the upper limit, we decided to raise the chance of generating new coins according to the balance[7] held by address that contains the Kernel found. More chances will lower difficulty to meet the target required by the network.
The chances will increase by multiples of 25 as the following:

$$CanStake = \frac{Kernel}{Chances} < NetworkDifficult$$

Once a stake has been made, coins used to generate the stake (up to 2500 KORE) are locked for two hours before they can be used to stake or spent again but the change (Sum of address balance - stake locked) can be spent but not stake till 25 confirmation.

[7] KernelAddressBalance is the sum of the first 25 highest UTxOs