Subjective Proof of Work

Daniel Larimer

***Abstract*—Ever since Satoshi Nakamoto introduced the world to the concept of digitally mining cryptocurrency, people have been intrigued by the idea of issuing a currency to those who do work that helps the community. In the case of Bitcoin, people were rewarded for providing computational power necessary to secure the network against counterfeit ledgers. Advancements in blockchain technology have given us faster, more predictable solutions for producing blocks and securing the public record, but so far nothing has been able to capture people's imagination like Bitcoin's proof of work (mining) did. This paper will introduce the next generation of proof of work, where currency is distributed based upon a more subjective measure of work than the objective measure used in Bitcoin mining.**

I. Background

The value of a cryptocurrency is derived from many factors, but perhaps the most significant factor is *network effect*. Every person who joins the network increases the value of the network for everyone else involved. Prior to the creation of specialized hardware, Bitcoin mining gave everyone who was interested an easy way to earn Bitcoin. Once people had Bitcoin they had a financial interest in the future success of the currency.

We believe the perceived fairness of Bitcoin's distribution significantly contributed to the success of Bitcoin. Everyone had an equal opportunity to participate and the work done to earn the Bitcoin was easily and *objectively verified* by all.

Now that Bitcoin mining has become a professional business, it no longer serves the purpose of growing the network effect by distributing the currency to everyone on an equal opportunity basis. Mining has turned into a long-term commitment where users must invest money to purchase specialized hardware that is lucky to mine enough Bitcoin in its useful life to replace the Bitcoin used to purchase the hardware. The purpose and/or benefit of mining has changed from being about *fair distribution* to *incentivising loyalty and commitment* from those who have made a long-term capital commitment to hardware. This commitment has a dramatic impact on how companies invest in and promote Bitcoin.

When Bitcoin mining was accessible to all the claim was made that Bitcoin was censorship resistant. This property of mining is no longer true in a world where two specialized mining operations based in China control over 51% of the mining power. Experience has shown that mining operators can filter transactions and have financial incentive to reduce the number of transactions they include in blocks. To achieve censorship resistance requires having a much larger pool of people whom have the ability to include transactions into the blockchain.

A. Alternative Distribution Models

Since Bitcoin, many other cryptocurrencies have introduced a range of distribution models. The first wave of alternatives changed the mining algorithm in an attempt to maintain a fair and equal-opportunity distribution model. Some currencies such as Primecoin attempted to generate a useful byproduct (discovering chains of prime numbers). Eventually many currencies abandoned proof of work and instead allocated all of the currency up front to those who helped build or fund the development of the currency.

Under BitShares, the currency holders were given the opportunity to vote on how to allocate funds to improve the protocol (up to 6% of supply per year). Despite the "equal opportunity" for anyone to earn new BitShares by getting votes, the barrier to entry is similar to the barrier to entry of getting elected to a public office. Only the most active and well-known members of the community working on relatively large projects can attract enough votes to get paid. Even then the allocation of funds results in unpleasant political debates where the losers decide to sell their BitShares and leave the community. A better solution is needed.

II. Introducing Subjective Proof of Work

As we have learned from Bitcoin, proof of work should serve several purposes:

1) Distributing currency to proportional to contribution
2) Deciding who should produce blocks
3) Deciding when blocks should be produced
4) Ensuring long term commitment to the currencies success.
5) Making an alternative ledgers of equal quality expensive to produce

Subjective Proof of Work presents an alternative approach to solving these problems that improves upon fully *objective* Proof of Work systems. Of particular note is the ability to have highly reliable and irreversible transactions with confirmation times as low as a few seconds along with a uniform block production interval.

A. Applications

The applications of a currency implementing *subjective* proof of work are far wider than any *objective* proof of work system because they can be applied to build a community around any concept that has a sufficiently defined purpose. When individuals join a community they buy into a particular set of beliefs and can vote to reinforce the community values or purpose.

In effect, the criteria by which work is evaluated is completely subjective and its definition lives outside the source code itself. One community may wish to reward artists, another poets, and another comedians. Other communities may choose to reward charitable causes or help advance political agendas.

The value each currency achieves depends upon the demand for influence within a particular community and how large the market believes each community can get. Unlike prior systems, subjective proof of work enables a community to collectively fund the development of whatever it finds valuable and enables the monetization of previously non monetizable time.

B. Rewarding Commitment

At the end of the day a currency is given value by the community of people who decide to *hold* and *trade* it. When the community is growing and the value is rising everyone is happy to buy and hold a currency, but the true test of the strength of the community, and therefore the currency, is how committed individuals are to holding the currency when times get tough. Once a currency starts to lose value every individual is faced with the dilemma of "selling while they can" or hold and risk becoming a "bag holder". If a currency introduces stake-weighted voting into the mix, it becomes even more critical to reward long-term commitment.

Under *Subjective* Proof of Work only those who commit to hold the currency for at at least a year are given a voice in the political process. The cost of locking up capital for a year is as real and measurable as the cost of electricity and computational difficulty of traditional Bitcoin mining and therefore can be considered *proof of work*. Few people will willingly do work (give up their liquidity) without any kind of compensation, just like few people would run up their electric bills mining Bitcoin without any compensation. Therefore, those who commit to hold the currency should be paid through the issuance of new currency. The economic result is to transfer value from those who want liquidity to those who give it up. Ultimately this mirrors the process of investing capital in Bitcoin mining hardware with the small caveat that you are guaranteed an increase in the amount of currency you hold.

C. Distributing Currency

There are two ways people can get involved with a cryptocurrency community: they can *buy in*, or they can *work in*. In both cases users are adding value to the currency, however, the vast majority of people have more *free time* than they do *spare cash*. Imagine the goal of bootstrapping a currency in a poor community with no actual cash but plenty of *time*. If people can earn currency by working for one another then they will bootstrap value through mutual exchange facilitated by a fair accounting/currency system.

Distributing a currency to as many people as possible in a manner that is generally perceived as fair is a challenging task. The tasks that can be entirely evaluated by an objective computer algorithm are limited in nature and generally speaking have limited positive external benefits. In the case of Bitcoin-style mining, it can result in the production of specialized hardware and cause people to invest time developing more efficient algorithms. It may even help find prime numbers, but none of these things provide meaningful value to society or the currency holding community at large. More importantly, economies of scale and market forces will end up excluding everyone but experts from participating in this kind of distribution. Ultimately, computation-based mining is just another way of *buying in* because it requires money to pay the electric bill necessary to do the work.

In order to give everyone an equal opportunity to get involved and earn the currency people must be given an opportunity to work. The challenge is how to judge the relative quality and quantity of work that individuals provide and to do so in a way that efficiently allocates rewards to millions of users. This requires the introduction of a scalable voting process. In particular it requires that authority to allocate funds must be as distributed and decentralized as possible.

The first step in rewarding millions of users is to commit to distributing a fixed amount of currency regardless of how much work is actually done or how users vote. This changes the question from being "Should we pay?" to "Whom should we pay?" and signals to the market that money is being distributed and is being auctioned off to whoever "bids" the most work. This is similar to Bitcoin committing to award 50 BTC to whoever finds the most difficult hashes. Like Bitcoin, all work must be done prior-to payout and nothing should be paid speculatively on the promise to do work in the future.

The next step is to reward everyone who does anything even remotely positive with something. This is accomplished by ranking all work done and distributing proportionally to its value. The more competitive the market becomes, the more difficult (higher quality or quantity) it becomes to earn the same payout.

D. Voting on Distribution of Currency

Assume there is a fixed amount of money to distribute, and that those who have a long-term vested interest in the future value and utility of the currency are the ones who must decide how to allocate it. Every vesting user casts their votes on who did the best work and at the end of the day the available money for that day is divided proportional to the votes such that everyone with even one net positive vote gets something.

The naive voting process creates a Prisoner's Dilemma whereby each individual voter has incentive to vote for themselves at the expense of the larger community goal. If every voter defects by voting for themselves then no currency will end up distributed and the currency as a whole will fail to gain network effect. On the other hand, if only one voter defects then that voter would win undeserved profits while having minimal effect on the overall value of the currency.

In order to realign incentives and discourage individuals from simply voting for themselves, money must be distributed in a nonlinear manner. For example a quadratic function in votes, i.e., someone with twice the votes of someone else could receive four times the payout and someone with three times the votes could receive nine times the payout. In other words, the reward is proportional to votes^2 rather than votes. This mirrors the value of network effect which grows with n^2 the number of participants.

Assuming all users have equal stake, someone who only receives their own vote will receive much less than someone who receives votes from 100 different users. This encourages users to *cooperate* to vote for the same things to maximize the payout. This system also creates financial incentive to *collude* where everyone votes on one thing and then divides the reward equally among themselves.

While *cooperation* to distribute funds to the best work is the desired goal, *collusion* that undermines this objective should be minimized. There are two kinds of *collusion*, the most straightforward is one user simply buying a larger stake than others, and the other involves coordinating a large number of smaller stakeholders to work together. Larger stakeholders can have the voting influence of 100 or even 1000 smaller stakeholders which means they have even greater incentive to defect by voting for themselves than they had under a linear distribution.

Regardless of how much money any one individual has, there are always many other individuals with similar wealth. Even the wealthiest individual rarely has much more than the next couple wealthiest combined. Furthermore, those who are have a large investment in a community and thus have the most to lose by attempting to game the voting for themselves. It would be like the CEO of a company deciding to stop paying salaries so he could pocket all of the profits. Everyone would leave to work for other companies and the company would become worthless, leaving the CEO bankrupt rather than wealthy.

Fortunately, any work that is getting a large concentration of votes is also gaining the most scrutiny (publicity). Through the addition of *negative-voting* it is possible for many smaller stakeholders to nullify the voting power of collusive groups or defecting large stakeholders. Furthermore, large-stakeholders have more to lose if the currency falls in value due to abuse than they might gain by voting for themselves. In fact, honest large stakeholders are likely to be more effective by policing abuse and using negative voting than they would be by voting for smaller contributions.

The use of *negative-voting* to keep people from abusing the system leverages the crab *mentality* that many people have when it is perceived that one individual is profiting at the expense of everyone else. While crab mentality normally refers to short-sighted people keeping good people down, it is also what allows good people to keep bad people down. The only "problem" with crab mentality is when people *wrongly* believe someone is profiting at everyone else's expense.

> **The Story of the Crab Bucket**
> A man was walking along the beach and saw another man fishing in the surf with a bait bucket beside him. As he drew closer, he saw that the bait bucket had no lid and had live crabs inside.
>
> ”Why don't you cover your bait bucket so the crabs won't escape?”, he said.
>
> ”You don't understand.”, the man replied, ”If there is one crab in the bucket it would surely crawl out very quickly. However, when there are many crabs in the bucket, if one tries to crawl up the side, the others grab hold of it and pull it back down so that it will share the same fate as the rest of them.”
>
> So it is with people. If one tries to do something different, get better grades, improve herself, escape her environment, or dream big dreams, other people will try to drag her back down to share their fate.

Eliminating "abuse" is not possible and shouldn't be the goal. Even those who are attempting to "abuse" the system are still doing work. Any compensation they get for their successful attempts at abuse or collusion is at least as valuable for the purpose of distributing the currency as the make-work system employed by traditional Bitcoin mining or the collusive mining done via mining pools. All that is necessary is to ensure that abuse isn't so rampant that it undermines the incentive to do real work in support of the currency.

The goal of building a community currency is to get more "crabs in the bucket". Going to extreme measures to eliminate all abuse is like attempting to put a lid on the bucket to prevent a few crabs from escaping and comes at the expense of making it harder to add new crabs to the bucket. It is sufficient to make the walls slippery and give the other crabs sufficient power to prevent others from escaping. Â

E. Rate Limiting Voting

A major part of minimizing abuse is the rate-limiting of voting. Individual users can only read and evaluate so many work items per day. Any attempt to vote more frequently than this is a sign of automation and potential abuse. Through rate limiting, stakeholders who vote more frequently have each vote count for less than stakeholders who vote less frequently. Attempts to divide tokens among multiple accounts also divides influence and therefore does not result in a net increase in influence nor bypass the rate-limit imposed on voting. Â Â

The charts below show how a user's voting power decreases every time they vote and then regenerates as time passes without voting. These charts use nominal time unit and could be made to scale to any targeted voting rate. Note that voting power rapidly drops off during periods of continuous voting, and then slowly recovers.

Voting power is multiplied by a user's vesting tokens to determine how many shares in the reward pool should be allocated to a given work item.

F. Delayed Payouts

To further prevent abuse, all payouts are delayed a stake-weighted average of 24 hours from the time each vote was cast. This ensures that large stakeholders cannot snipe payouts by voting at the last second before other voters (aka crabs) have a chance to negate the potential abuse. Once 24 hours have past without additional votes a payout is made to the user and

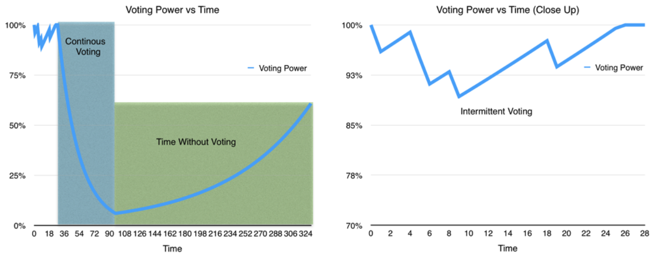

Fig. 1. Illustration of voting power over time

all votes are reset to 0. If votes come in after the payout the process begins again.

The chart below shows how the voting period expiration changes in response to new positive and negative votes being applied. New votes extend the payout period in proportion to how large they are relative to all votes that have gone before. Around time 40 a large number of new votes were added which extended the voting period by 12 hours, subsequent smaller votes had far less impact on the voting period.

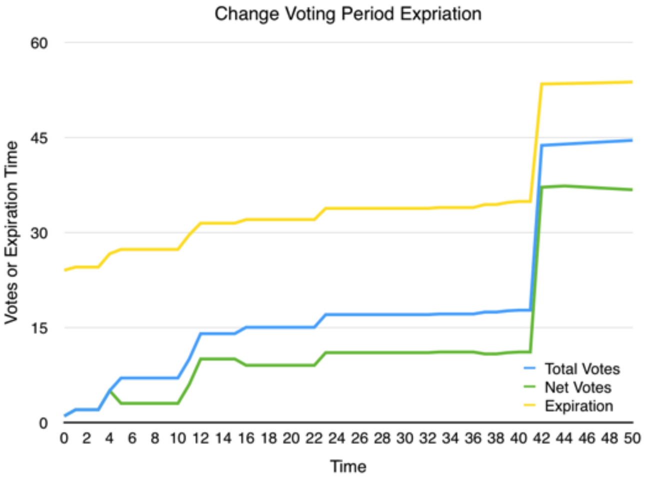

Fig. 2. Illustration of the voting period expiration

G. Payout Distribution

Assuming a normal distribution of voting then most people will have an "average" number of votes, while relatively few people have a lot of votes. Under the distribution model described where percent of payout depends upon

$$\text{payout percentage} = \text{votes}^2 / \sum \text{votes}^2 \quad (1)$$

then we can estimate that 40% of payout will be distributed among the top 10% of the work, 70% among the top 20% of the work, and 95% among the top 50% of the work. The bottom 50% of the work submissions would receive just 5% of the payout.

Obviously a normal distribution isn't exactly how votes will be distributed as median number of votes is likely to be very different than the average due to there being more work submissions with low votes than work with high votes.

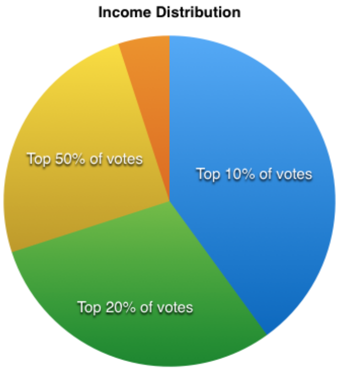

Fig. 3. Illustration of Income Distribution

The result of this allocation is that some users will generate substantial income from their work while others will get a token amount. This follows the principle that you can get more work done by paying one person $100,000 per year than paying 100,000 people $1 per year. Everyone is competing to receive the most votes.

The economic effect of this is similar to a lottery where people over-estimate their probability of getting votes and thus do more work than the expected value of their reward and thereby maximize the total amount of work performed in service of the community. The fact that everyone "wins something" plays on the same psychology that casinos use to keep people gambling. In other words, small rewards help reinforce the idea that it is possible to earn bigger rewards.

III. Implementation Details

The challenge with all blockchain concepts is to find an implementation approach that efficiently implements the desired economic policies. What follows is the basic implementation strategy for how to implement the various components using constant-time algorithms.

The first part of the design is to specify the ratio of long-term holders to short-term holders. This is a balancing act between the demand for commitment and the need for liquidity. For the purposes of this paper we will assume a target ratio 50% vesting / 50% liquid. To avoid confusion, we define *inflation* to mean an increase in the total supply of tokens rather than loss of purchasing power.

The network needs to create economic incentives that automatically push the ratio toward 50/50. It does this by *inflating* the currency and distributing 50% of the inflated currency proportional among the vesting users. For the sake of simplicity we will assume that the currency supply doubles every year (100% supply inflation). If only 25% of the users are currently vesting then they will earn an inflation-adjusted return of 50% APR as their collective ownership grows from 25% to 37.5% over the first year. Assuming no one changes their allocations between vesting and liquid tokens then over time the vesting users will approach, but never quite reach, 50% ownership of the tokens. As vested owners approach 50% their inflation-adjusted rate of return approaches 0%. If at any point of time more than 50% of the token holders opt to become vested, then it is possible for users to see a negative inflation-adjusted rate of return. The inflation rate can be used to control how quickly the system converges on 50/50 allocation between vesting and liquid.

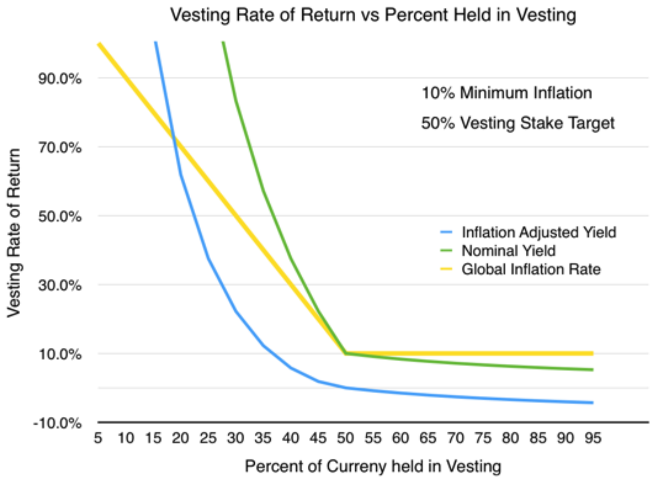

Fig. 4. Illustration of Vesting Rate of Return

The selection of inflation rate also impacts the effectiveness of the token as a currency and controls the rate at which the currency can grow by funding new work. Early on in the life of a token the inflation rate needs to be very high to promote growth and investment. All other things being equal, the inflation rate needs to match the growth rate of the communities economy or it will start to negatively impact its currency function. An inflation rate of 0 and infinity both result in a complete unraveling of the community economy. With no inflation it becomes impossible for the community to fund growth through subjective proof of work. With infinite inflation it becomes impossible to use the token as a currency due to rapidly falling exchange rates.

As the network approaches the desired ratio between vesting and liquid tokens the inflation rate should fall to its minimum value. As the vesting/liquid ratio gets further from the desired target the inflation rate should increase until it reaches its maximum value.

Unlike traditional currencies, all long-term savers are protected from inflation when they commit to hold for over a year and usually they will earn a positive inflation-adjusted rate of return. Only in extreme cases where everyone wishes to be a long-term holder do the vesting users see a slight negative inflation adjusted rate of return.

All prior statements assume a constant market capitalization of the base currency; however, in the real world this valuation will change constantly. The situation when vesting users have a negative inflation adjusted rate of return indicates that there is high demand for long-term commitment to the currency which will likely result in a growing market capitalization that could potentially result in a positive real (relative to purchasing power) rate of return for all vesting users. If demand is strong enough, even non-vesting users may see capital gains on their currency.

The opposite is also true, if the market capitalization is falling faster than the inflation-adjusted rate of return on the vesting balances, then these users may still see a negative real rate of return.

IV. Block Production

To understand the block production algorithm used by subjective proof of work, it is helpful to refer to BitShares implementation *delegated proof of stake*. When it comes to producing blocks, nothing is more regular than the Delegated Proof of Stake (DPOS) model introduced by BitShares. Under DPOS a number of individuals are elected by BTS shareholders to produce blocks by approval voting. The active set of block produces is shuffled and then each is given an opportunity to produce a single block. This is known as a round. After every round, the set of active block producers is updated and reshuffled. Blocks are produced on a fixed schedule of every 3 seconds and the nodes are synchronized via the network time protocol (NTP).

Under the Subjective Proof of Work model, vesting stake is considered to represent proof-of-work similar to the purchase of mining equipment. Vesting stake can then be used to delegate its block-producing power in a manner similar to mining pools or voting proxies in DPOS.

There are several ways to determine the how big the set of active block producers should be. Take the top N by vote, define a minimum voting threshold, or have the stakeholders vote on what N should be. All of these approaches create two classes of users and subject participating in the block production process to a political process. This creates a barrier to entry that could ultimately result in censorship and centralization.

Under Subjective Proof of Work we would like a system where anyone who wants to contribute to the network by producing blocks has an opportunity to do so. The challenge is to maintain the reliability and consistency that is provided by DPOS while opening up opportunities for all. Individuals with little stake produce far less often and have far less incentive to be reliable. There is also a need to have consistently reliable measure of irreversibility so the network must get as much direct or delegated work confirming every block as quickly as possible.

We propose, that the active set of block producers consists of the top N accounts elected by approval voting, plus W accounts selected by objective proof of work (e.g. classical mining) and R of the remaining M accounts *nominated* by approval voting, e.g.

$$M = N + W + R \tag{2}$$

For the sake of having hard numbers to work with we will assume the top 19 accounts by approval voting, 1 account by objective proof of work, and 1 account that didn't make it into the top 19 by approval voting. We will also assume 2 second block times.

This means that there is one objective proof-of-work confirmation every 62 seconds on average. It also means that everyone who is running a full node has an opportunity to produce a block (and include any valid transaction) once every 62 seconds on average.

V. Scheduling Block Producers with low Vote Priority

The top N block producers are given high vote priority and get to produce blocks every round. The remaining individuals who wish to produce a block must take turns getting scheduled about once per minute. The goal is to efficiently schedule these remaining producers proportional to the votes they have received. Someone with twice as many votes should be scheduled twice as often. The algorithm chosen is a variation of weighted fair queueing (WFQ). WFQ assigns each block producer a virtual time based upon their vote weight and how long they have been waiting to produce a block. Any time votes change the virtual time is updated to reflect the new priority. At the start of each round, the block producer with the lowest virtual time is selected to produce a block and their position is moved to the back of the queue.

To ensure network reliability, everyone wishing to produce blocks must pay a fee equal to 10x the current block reward to get on the schedule rotation. If two blocks are missed in a row they are removed from the schedule. Anyone wishing to participate in block production must successfully produce at least 10 blocks to break even or they will ultimately lose money. Someone who is unable to get enough votes to be scheduled frequently enough to make a profit shouldn't sign up for producing blocks in the first place. Those who are in the top 19 are not subject to removal except by users updating their vote. This is a security measure designed to protect the network against certain kinds of attacks.

A. Integrating Objective Proof of Work

We would like to have the benefits of Objective Proof of Work (OPOW, aka mining) without the downsides such as unpredictable block production time, mining pool centralization, or the potential for recent blocks to be orphaned. The primary benefits of OPOW include:

An objective measure of quality that is expensive to forge
A financial incentive to optimize a useful/necessary computer algorithm A distribution model that attracts tech-savvy users

Unlike traditional mining, block production time is separated from the time when work is performed. When a solution is found that meets the target difficulty, a transaction is submitted to the network and included by the current block producer. To be included, the OPOW must be derived from the current head block. The user is then added to a queue to be included in a future block production round. The target difficulty becomes a function of the queue length. A simple algorithm (as it is implemented in Bitcoin) would require a number of leading 0 bits equal to the number of producers in the queue.

B. Resistancy against Pooled Mining

With two-second block times, a OPOW miner needs to operate a node with minimal network latency so it can get the new head-block as quickly as possible and then submit its result to the network with enough time to propagate to the next block producer. The introduction of a mining pool would add additional latency that would dramatically reduce the percentage of time available to actually do work.

C. Mining Algorithm

While any mining algorithm could be used, we would like to introduce a new algorithm that has several beneficial properties. The mining algorithm requires proof that the miner possess the private key for the account that will ultimately produce the block and receive the reward. The algorithm also requires the user to do an elliptic curve signature verification, the optimization of which will benefit the validation of all transactions and lower the cost of operating the network in the long run.

The algorithm:

```
Require: $\mathrm{HASH}(\cdot)$ to be a secure cryptographic hash function
  (SHA256 or better)
Require: $p$ to be the producer's private key
Require: $P$ to be the producer's public key
  $i \leftarrow$ Head Block ID
  $S \leftarrow \mathrm{SIGN}(p, \mathrm{HASH(i)})$
  $H \leftarrow \mathrm{HASH(i + nonce)}$
  $K \leftarrow \mathrm{RECOVER\_PUBLIC\_KEY}(H, S)$
  $\mathrm{POW} \leftarrow \mathrm{HASH(K)}$
Ensure: $\mathrm{POW} \leq$ target difficulty
Ensure: $K = P$
```

To be valid the POW must be less than the target difficulty and $\mathrm{RECOVER\_PUBLIC\_KEY}(H, S)$ must equal P. The miner introduces randomness in either the selection of the nonce or via the randomness required for elliptic curve signature generation. This, combined with the private key selection should ensure that no two miners are searching the same work space.

By starting and ending the POW with a cryptographically secure hash function we can ensure that any vulnerabilities or computational shortcuts that may exist in the $\mathrm{RECOVER\_PUBLIC\_KEY}(H, S)$ algorithm or SIGN algorithm will ultimately cause the POW algorithm to revert back to a simple HASH-based POW.

D. Vesting Rewards

Traditional OPOW mining algorithms create new tokens that are liquid the same day. This means the decision to mine depends only on whether or not it is profitable today. By requiring all mining rewards to vest for one year the system removes the opportunistic miners who jump in and out of mining based upon short term price or difficulty changes and leave only those who have a long-term belief in the value of the currency. This long-term outlook means miners have greater incentive to be reliable producers and act in a way that adds value to the network and should result in a more stable contributions.

E. Mining Efficiency and Waste

Those who follow the proof of work vs proof of stake debates have heard the concerns that OPOW mining is wasteful and non-productive. We agree and believe that OPOW is an unnecessary addition to the overall subjective proof of work model. The network and consensus can be equally secure and censorship resistant with or without it. Whether or not mining is wasteful depends upon the degree to which it helps a particular currency with user acquisition or results in the production of more efficient algorithms. It also depends upon how much of the currency is awarded to those who do OPOW mining relative to Subjective Proof of Work.

VI. Conclusion

We have presented a new consensus algorithm and currency distribution method that is designed to maximize user participation, minimize centralization, reward long-term commitment, resist censorship, while providing high reliability, low variability, and low latency transaction confirmations. Many different communities can use the same consensus algorithm and code to achieve an infinite variety of goals.