LAVA

I. Background

A) Revolution in distributed systems and model of collaboration

Distributed Ledger Technology (DLT) is a decentralized ledger maintenance and update mechanism, a mode of ledger organization that combines peer-to-peer networks, consensus mechanisms, and cryptography. Blockchain, as a typical DLT example, features a dendrogram that defines a valid ledger through specific selection rules (by specifying such items as the longest chain or the vote with highest weight). In Blockchain, blocks are connected end to end and direct to a unique way, making the whole system traceable and verifiable and hence forming the technical basis for trust construction of decentralized networks. For whatever purpose the Blockchain is designed (mere payment or smart contract), it has in the first place changed the model of human collaboration. Financial asset ledger, computing resources, as well as storage resources all can be mobilized in a trustless manner throughout the network. Given its natural network effect, it is expected to be able to shape or reconstruct new industries.

The consensus mechanism has solved the problem of how not-mutually-trusted nodes or individuals can reach consensuses in a decentralized network. This problem, technically or economically, has always been at the core of the distributed ledger technology. Therefore, the design of a universally recognized consensus mechanism is the ultimate Holy Grail for each working in the field of decentralized networks. And the revolutionary impact of DLT technology represented by Blockchain on human beings must be built on a consensus mechanism that is robust, energy efficient, fair, open and can maintain a high degree of decentralization on a long-term basis.

The society is always keeping an eye on whether emerging technologies can **solve those real social problems** and whether they can truly improve the efficiency of social productivity. Undoubtedly, the impact of information technology on both relation and efficiency of social productivity has been verified and is still continuously deepening. While **storage** and **computing** are the very two fundamental factors of the revolution in information technology productivity. Thus, we believe that distributed ledger technology not only needs to provide a mechanism for building consensus and de-trusting (ie. **value decentralization**), but also should build decentralized infrastructure for the two core productivity factors (ie. **storage decentralization** and **computation decentralization**).

B) Lava’s vision

Lava is a digital and cryptographic infrastructure based on **Proof-of-Capacity (PoC)**. Also, it embodies **Root-of-Trust** and **Top-level Indexing** mechanism for the global storage space.

PoC is a consensus mechanism with high security but low energy consumption, and of great fairness and openness. It is therefore conducive to building a stronger trustless on-chain ecology and assembling a wider range of consensus and value. Lava has adopted an improved PoC mechanism, namely *Lava-Firestone*, greatly relaxing the hardware barriers set to maintain a decentralized network, and making it easy and cost-effecient for anyone to utilize handy but idle storage devices to participate in block forging.

The whole ecology is designed for a complete close loop: Lava builds consensus from global storage space, and in return gives feedback to the space with the trust-value gained from consensus. The concept of “building” comes from the PoC consensus mechanism adopted by the main chain. It requires a large and distributed storage networks to contribute “capacity power” to create Root-of-Trust on the blockchain. And the trust-value feedback, based on the decentralized trust facility, Lava, is widely used as a general and core open protocol for indexing and mobilizing global storage resources by third-party applications and services through such proven mature technical solutions as cross-chain extension, virtual token coloring, and distributed content-addressable storage networks.

II. Introducing a PoC-based consensus mechanism

A) Brief on principles of PoC

PoC is a capacity-based consensus mechanism. Miners increase the forge rate by providing greater storage capacity.

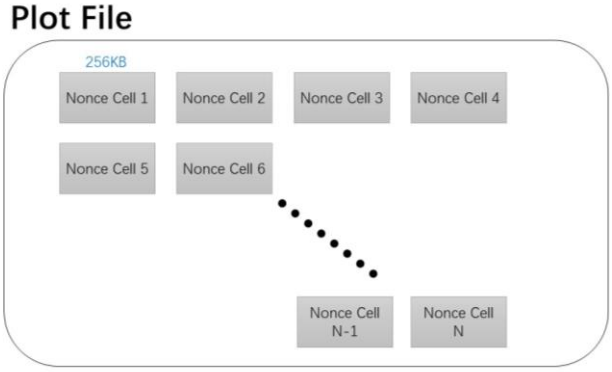
PoC miners rely on statically stored data to participate in the competition of block forging. These special data are arrays of sequenced calculations based on a particular hash algorithm. They are generated by the miners through calculation in advance and are cryptographically bound to their addresses. Data processed in this way are called a *Plot file*.

In the forging process, the consensus mechanism randomly specifies a location in the data array of a *Plot file* through the very data from to-be-generated block. Forging competitors then retrieve the corresponding data in their own *Plot file* and generate a *Deadline*. The *Deadline* indicates a mandatory time delay that nodes have to wait before broadcasting the newly-forged block. Therefore, generating the minimum *Deadline* means seizing the forging.

Considering the fact that *Plot files* can be generated at one time yet stored and reused for long, and the work required for the forging is but network broadcasting, retrieval and simple verification computing, the consumption of high-powered computing resources and energy by PoC mechanism can be lowered to a minimum. Under the same assumption, the power consumption required for PoC operation is far less than a fraction by the PoW.

B) Generating a Plot file

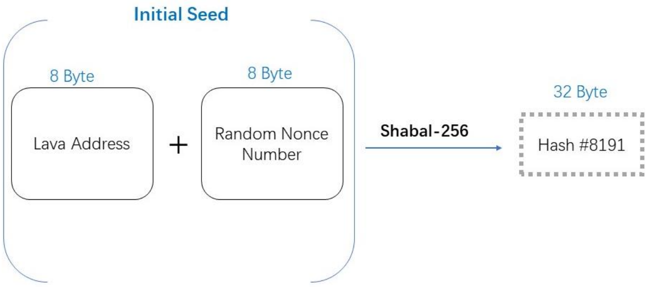
A ***Plot file*** is a data array made up of a series of hash calculation results through permutation and combination. Each elementary data array unit in a Plot file is called a ***Nonce Cell***. The data capacity of each *Cell* is fixed at 256 KB. The larger the storage of the miner, the more the *Cell* can be stored, which naturally will increase the success rate of block forging.



The *Cell* generation will involve the 256-bit ***SHABAL256*** hash function. *SHABAL256*, which is a hash algorithm featured by slow calculation process, well fits the requirements of the PoC algorithm.

The starting point for generating a *Cell* is the user address. The *address* (8Byte) is spliced with a *Nonce Number* (8Byte) to form an *Initial Seed* (16Byte).

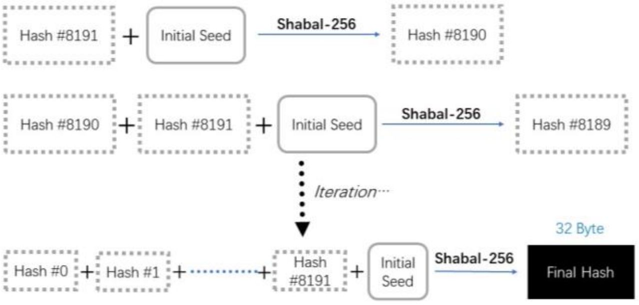
Perform a *SHABAL256* calculation on the *Initial Seed* and get the first hash result #8191;



Add #8191 before the *Initial Seed* to form another seed (#8191+*Initial Seed*), and perform the *SHABAL256* calculation again to get the second hash result #8190;

Add #8190 before the previous seed to form a new seed (#8190+#8191+Initial Seed), and perform the *SHABAL256* calculation for the third time to get the third hash result #8189;

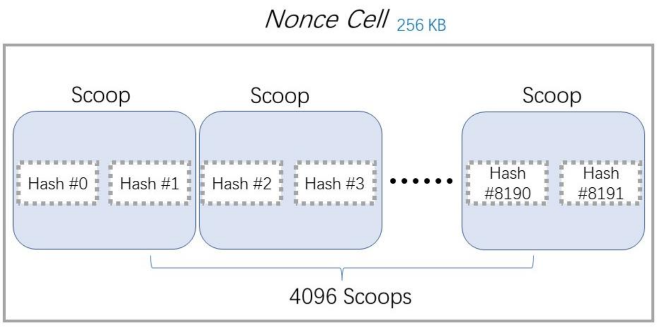
Analogously, each time the previous hash result is added before the latest seed until the very last seed is generated (#0+#1+...+#8190+#8191+*Initial Seed*), and the *SHABAL256* calculation is performed again to get the ***Final Hash***.



Each hash result is 32Bytes in length. Once the seed length exceeds 4096Bytes in hash calculation, only the last 4096Byte will be taken.

Then the calculated 8192 hash results (#0, #1...#8191) get XORed in *Final Hash*, and the according 8192 results are saved (still labelled into #0, #1...#8191).

Following this, group the 8192 hash results into adjacent pairs. Each group is called a ***Scoop***, so 4096 *Scoops* are obtained, which are filled in the *Cell*. The *Cell* construction has to this far been completed.



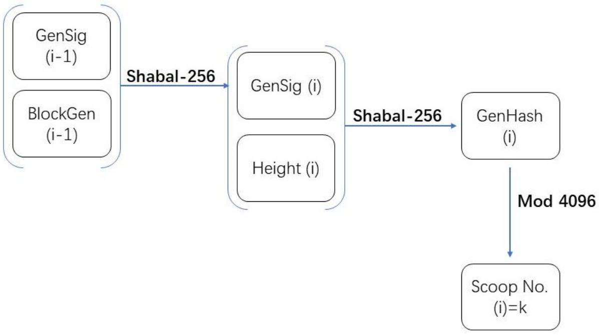
In *Cell* generation, the computer must use cache to record all calculation results in the process to obtain the *Final Hash*. Since every Cell contains 8192 *SHABAL256* hash results, all 32Bytes in length, each *Cell* will occupy a fixed 256KB space.

Repeat *Cell* generation process and rearrange all the *Cells* to fill the Plot file. The *Plot file* is at this point completely ready.

C) Block forging and verification

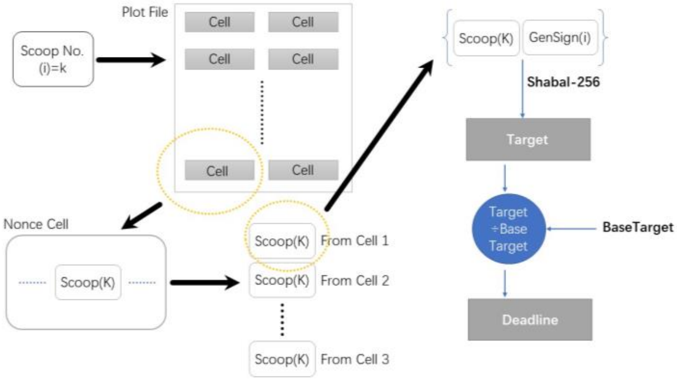
When the *Plot file* is there, miners can now start the block forging.

When forging, the Miner program receives the broadcasted transaction and packs it to produce a to-be-generated block. The program will get the current *difficulty*, the *height* and the *Generation Signature* from the block. The *SHABAL256* calculation is then performed on *GenSig+height* to obtain the ***GenHash*** which is used to modulate 4096 to obtain the *Scoop Number* in this forging.



The Miner program then retrieves and extracts all *Scoop* data corresponding to the *Scoop Number* in the *Plot file*, attaches the *GenSig* to the data and performs the *SHABAL256* calculation to obtain the ***Target***. The *Target* is then divided by a system parameter ***BaseTarget*** representing the *difficulty* to obtain the *Deadline* (8Byte).

The ***Deadline*** represents a mandatory wait time, the delay a new block timestamp needs to wait after the previous block timestamp. Blocks generated before the delay is met are seen illegal and will not be accepted by the network.



If the Miner program receives a new block broadcasted by the network, it will verify the data provided by the block, including the *Initial Seed* and *Deadline* of the selected *Cell*. And the Miner program of any network node can complete the verification in a short time.

III. Lava-Firestone Consensus

A) Overview of the Consensus

1) The purpose of the Consensus

Consensus is an intermediate process designed to solve the problem of how an asynchronous-communication network cluster achieves consistency. It is assumed that there should be a certain proportion of trustworthy individuals in the network during the consensus process for it will be difficult to reach consensus in networks where cheaters (or attackers) dominate. In a more general sense, the trustworthy individuals of a network are contained in behavioral norms through some material rewards or penalties in the whole consensus process.

In a peer-to-peer and decentralized network, the more distributed the structural basis of a consensus is, the lower the efficiency of the consensus, but the higher the reliability will tend to be. Likewise, the more centralized the structural basis of a consensus, the higher the efficiency of the consensus, but the lower the reliability, and the whole system will tend to be more centralized. Reliability is also generally interpreted as the "degree of decentralization", hence a good consensus mechanism should well balance efficiency and reliability by staying open, fair, and friendly to hardware and running environment.

2) Consensus practice:

PBFT (Practical Byzantine Fault Tolerance) is a consensus mechanism based on Byzantine Fault Tolerance. PBFT can achieve (N-1)/3 fault tolerance while maintaining its security. However, the time complexity required by PBFT at O(N^2) has made it not scalable. In addition, PBFT to some extent is a permissioned network, which also impairs its openness.

Proof of Work (PoW) is a **competition-based** and **reward-based** consensus mechanism. The *HashCash* concept proposed by Adam Back in the 1990s has adopted the PoW to screen those non-trustworthy in a system, which is later widely used to filter junk mails. The more difficult a PoW, the more efficiently cheaters will be excluded. It is hence a solution that can achieve a relatively ideal consensus effect. However, it is also proved that PoW had shown two primary defects in applications: a) the energy consumption to maintain the consensus is huge; b) the computing power **going ASIC** is inevitable and irreversible.

Proof of Stake (PoS) reduces or avoids the competition among computing powers for accounting rights by giving appropriate weight to token holders. Compared with PoW whose reliability relies on holders' tempt for material rewards, PoS, however, maintains consensus through holders' fear of loss. However, PoS lacks continuous openness because the initial distribution of stake is somewhat exclusive.

B) Problems addressed by PoC

Proof of Capacity (PoC) with robust, open, and clean features provides a superior consensus mechanism for decentralized systems.

1) The robustness of PoC:

The PoC consensus requires miners to prove to the network the storage capacity they hold, and incentivize them into participating in the competition of block forging to maintain the security and credibility of the whole decentralized network. The implementation process of the PoC consensus determines:

- PoC is a consensus of competition;
- PoC is a consensus of randomness;
- PoC miners will have to work (or invest cost), and their work (or invested cost) can be verified by third parties at a lower cost.

The above features have ensured that the PoC mechanism is an extremely robust consensus. Robustness here is used to evaluate the insensitivity of the control system to system features or external parameters, that is, the ability to maintain stable and efficient operation even under interference from internal and external systems, especially from some other uncontrollable factors.

PoC miners need to get the storage media they are having or controlling continuously occupied during the forging process and prepare *Plot files* in advance. Since it will be impossible for the miners to create *Deadline* data that can win over in the forging competition through real-time high-powered computing without *Plot files*. This is because the *Deadline* data is determined by the very information of the current block and is also determined by the *Nonce* data. The *Nonce* data force the miners to strictly follow a certain order in the forging and are determined by an algorithm sensitive to storage space. Therefore, the PoC forging is random at the micro level. In general terms, the chance of cheating in PoC is as slim as that of a hash collision.

In addition, the PoC miners need to submit the necessary raw data during the *Deadline* generation along with the to-be-generated block, so the complete process of the *Deadline* generation can be verified by a third party. Anyone at any time can backtrack the validity of the forging at any historical state, further ensuring the credibility and anti-tampering of the PoC.

Also, the highly competitive forging process has led to a significant increase in the cost of cheating, so that participants will have no incentive to take the risk. If evaluated only from a competitive perspective, the PoC consensus is highly similar to the PoW consensus. The latter is a typical consensus mechanism that screens cheaters through workload competition. The advantage of it lies in the fact that the establishment of consensus depends on resources which cannot be replicated in the short term, so some irreversible cost in relation to trust will exist.

2) Openness of PoC:

PoC is also a consensus mechanism with excellent openness. Some openness requirements include:

- Consensus sets the threshold for miners as low as possible;
- The criteria of the threshold are stable or at least within stable expectations;
- Potential participants can join the consensus or withdraw from it **at any time**;
- Consensus can be **infinitely scalable** in relation to forging power;

PoC can meet any of the above.

PoC requires miners to provide storage media and access the Internet to participate in forging. Participants in this consensus mechanism can almost get a return equal to their storage space (which we can also call "computing power"), and the value can be quantified by estimating the forging power of the whole network, so there is no nonlinear return problem.

Besides, PoC participants are free to join or exit the forging process without any constraints. Considering the process of the *Plot file* preparation, newly added large-capacity forging power to the network will not be able to complete the power calculation in a very short time and hence will cause a power mutation. In fact, this mechanism only smoothens the possible power change, and does not substantially cause any obstacle to the entry or exit of power. Moreover, regardless of the capacity of the network's existing computing power, PoC can accommodate newly added computing power anytime, anywhere, without a cap.

Though PoW also has the above features, its requirements for high-powered computing hardware have led to an irreversible problem of hardware going ASIC. This is because the hash computing power is sensitive to the architecture of the computing hardware. For instance, the efficiency of CPU architecture lower than the GPU architecture, the efficiency of GPU architecture lower than the FPGA customized circuit, and the FPGA customized circuit lower than the large-scale integrated ASIC device. And this is impossible to avoid. Therefore, we believe that due to this high sensitivity of PoW to hardware, the consensus will lead to the participants being too much specialized, centralized and also irreversible, which is apparently detrimental to the openness of the consensus.

Further, we should be aware that:

- Openness is a crucial basis for **fairness**;
- Openness is the basis for maintaining **decentralization**;

Therefore, PoC well meets the openness requirements of the consensus protocol and is therefore beneficial to the realization and long-term operation of the decentralized system.

3) Cleanness of PoC:

The cleanness of PoC is mainly differentiated from PoW. As mentioned above, PoC and PoW both are consensuses of competition, so they are also comparable in terms of cleanness. While non-competitive consensuses, such as PoS, DPoS and PBFT, have logic flaws in terms of decentralization and system openness. That's why they share no comparability with this issue regard.

Generally, if PoW is positioned as a "hot" forging, PoC is a "cold" one, a process that is almost static.

The principle is that the PoC forging requires the miners to prepare the *Plot file* in advance and allow it for long-term repeated use. In the forging process, miners do not need to repeat the calculation to generate *Plot files*. Only a few simple searches and verifications will be enough. The demand for electrical energy in the forging is therefore contained at a minimum, that is, at the level of maintaining normal operation of the storage device.

As mentioned above, PoW can be infinitely scalable in terms of forging computing power. Yet evaluated from the perspective of energy consumption and economic concerns, this property is in fact ungrounded. Electrical energy is a limited and socially meaningful resource, but the underlying resources used to generate electrical energy are not always renewable. This fact turns the PoW consensus from an issue of virtual domain into a social concern, and the sustainability of its extension is therefore questioned. While the "static forging" of PoC has largely spared us the trouble to worry about energy consumption.

C) Implementation levels of meaningful storage

1) Concerns about "meaningless storage":

It is widespread concerned that cryptocurrency based on PoW wastes a lot of computing resources and power. Similarly, PoC is not entirely exempted from this problem. Since PoC requires miners to provide storage media as indispensable forging data, it will also be unavoidable for these storage media to be occupied by a large number of arrays of hash value.

So, is this behavior truly meaningful? We will try to explain at different levels Lava's understanding of "meaningful storage" and "decentralized storage ecology", and the philosophy of solving this problem.

2) Maximizing the utilization of space resources:

Competitive mining (PoW and PoC) takes advantage of short-term non-replicable resources (computing units, physical devices of storage units) as well as the accumulation of time (or space) in exchange for forging rights. The occupation of the resources forms the basis of decentralized trust objectively. While Blockchain, expected to change the model of collaboration and business, is hence highly dependent on the trustless infrastructure. Therefore, the computing equipment invested by PoW and the storage space invested by PoC both are like the large amount of asphalt used in the paving of highways, rather than simply a waste. The saving of power resources by PoC of course will still demonstrate great social value.

The PoC consensus advocated by Lava encourages the use of surplus storage space at a material and incentive level, which is a demonstration of maximizing resource utilization. Storage space is a **tangible** resource. And laying idle for such resources cannot be more universal, which leads to a weakening of equipment utilization and a waste of social resources. The coming of "One CPU One Vote" Nakamoto has envisaged at the beginning of the design of Bitcoin is precisely due to his original vision of "using the idle computing power to maintain the peer-to-peer electronic cash system". However, PoW owing to its defects has turned itself going ASIC an irreversible steer, causing the current status que of it featuring high energy consumption and hardware threshold. Therefore, in this sense, PoC goes further than PoW and realizes the vision of Nakamoto more fully and thoroughly.

3) Lava as the infrastructure for decentralized storage:

We believe that what mentioned above is but theoretical interpretations of the consensus process consuming necessary resources, which yet does not truly present Lava's vision out. "Meaningful" storage must help to generate **real social value**, regardless of whether it is implemented in a decentralized manner or not. Lava plans to in stages implement:

- Building global storage space consensuses through the PoC mechanism and becoming a decentralized credential (“Root-of-Trust”);
- Feeding the accumulated trust value into storage-based applications and services, especially the trust infrastructure for decentralized storage applications and services;
- As a trust infrastructure (Lava Infra), assuming the responsibility of the top-level indexing of global storage (Lava as Top-level Indexing);
- Embedding decentralized storage application and service ecosystem (Lava Layer 2 Embedment Protocol Architecture) into the Lava mainchain, and implementing cross-chain embedding in communication technology, asset flow and economic incentives.

Lava’s vision is based on the maturity of a series of underlying technologies and ecosystems, including but not limited to:

- Cross-chain asset trading technology based on *2-way peg*;
- Script-based technology of *cross-chain atomic swap*;
- Off-chain extension solutions, *payment channel* and *state channel* technologies;
- Distributed storage networks based on content addressing;
- DHT mechanism and variants of distributed storage networks;
- Trusted proof mechanisms for file storage and retrieval, including *Proof-of-Replication* and *Proof-of-Spacetime*;
- Web-scale serverless computing architecture, including *Lambda* and *Fargate*.

For example, the Lava mainchain can conduct Layer 2 embedding to spatially unrestricted and an unlimited number of distributed content-addressable storage networks through a cross-chain solution. It provides a trustless environment through the “space power” of the main chain to combine and record the top-level indexing, resource mobilizing and corresponding transaction behaviors of distributed storage network resources. Deriving in this way, protocol architectures based on the Lava as Top-level Indexing and Lava Layer 2 Embedment can support the following scenarios:

- Recording the transaction of idle storage resources, mobilizing, and managing assets;
- Decentralized DNS system based on distributed content-addressable storage networks;
- Trusted decentralized data deposit;

- Trusted computing environment;
- Mobilizing and settlement of distributed storage tasks based on centralized service;

In view of the fact that this paper is to elaborate mainly on the Lava main chain and the PoC-based Lava-Firestone consensus mechanism, the above-mentioned technical scenarios are not fully unfolded here. If you are interested, please follow the development progress of the project and the **Lava Foundation** for more details.

D) D) Lava-Firestone Consensus

1) Brief on the Lava-Firestone Consensus:

Lava has adopted an improved PoC consensus mechanism, the **Lava-Firestone Consensus**. This Consensus has been re-designed, refactored, and optimized in terms of the following factors:

- Optimizing the distribution model of PoC consensus and *Plot file* generation algorithm to make it more stable and suitable for long-term ecology development;
- Preferring to build on trust-infrastructure based on PoC in early stage, whereas tending to encourage meaningful storage and applications or services based on decentralized storage subsequently;
- Considering technology solutions based on cross-chain, off-chain extension, Layer 2 embedding, and economic incentives;
- Considering setting up a system built-in credential--**Firestone**;
- • Considering the Firestone-based token coloring;

2) Firestone:

The mechanism has introduced a concept called *Virtual Layering* in an innovative way, which is known as "Firestone."

The Firestone features:

- A system built-in credential whose design is based on *Virtual Layering*;
- A representative of occupying resources of the Lava system;
- A representative of contribution to the Lava ecology
- The governance right credential of Lava Ecology;
- The carrier of voting rights in the on-chain governance system;
- The economic right credential of Lava Ecology;
- A non-permanent, customizable and non-fungible credential.

Distribution, use and circulation of Firestone:

- By any participant who contributes to system security and consensus building;
- By system members who contribute to meaningful storage ecology;
- Generated and distributed by individuals, organizations or institutions based on their own credit endorsements through *Coloring* mechanism;
- Firestone is mainly generated by freezing or mortgaging Lava;
- The use and circulation of Firestone are both customizable and non-fungible since decentralized storage or computing applications and services based on the Lava ecosystem are also non-fungible. And Lava applications or service providers can also define different usage rules.

3) Firestone generation and reward mechanism of the ecology development:

The period from the beginning of Lava Blockchain going online to before the meaningful storage ecology gets matured is defined as the ecology development. To address the cold start of the trust infrastructure during this period, system participants can obtain Firestone by initiating a transaction to freeze Lava.

The mechanism adopts a "***Dynamic Freezing***" to generate Firestone:

- The Lava Blockchain is divided by every 2048 blocks, and each divided result is called a ***Slot***.
- The user obtains Firestone by initiating a transaction to freeze Lava.
- The Firestone obtained by the user in the N-1th *Slot* is valid only in the Nth (next) *Slot*. When the Nth *Slot* is over, the Firestone is automatically abolished, and the frozen funds will be returned.
- During the ecology development, the full block reward consists of 50% basic reward and 50% ecology reward. When miners have successfully forged the block, they obtain the basic reward unconditionally. If they consume an activated Firestone, they can also obtain the additional ecology reward.

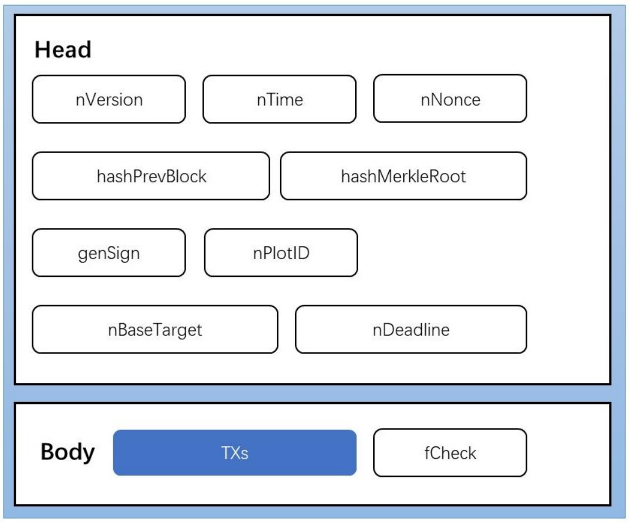
When users initiate a freezing transaction, they need to obtain Firestone at a dynamically-adjusting ***freeze ratio***. The *freeze ratio* can be easily understood as the "price" of Firestone, that is, how many Lavas needed to be frozen in exchange of a Firestone.

The *freeze ratio* is adjusted dynamically at the beginning of each *Slot*. When the number of Firestones in the previous *Slot* exceeds the target value of 2048, the *freeze rate* of the *Slot* will increase by 5%. And when the number of Firestones in the previous *Slot* is less than the target value of 2048, the *freeze ratio* will decrease by 5%.

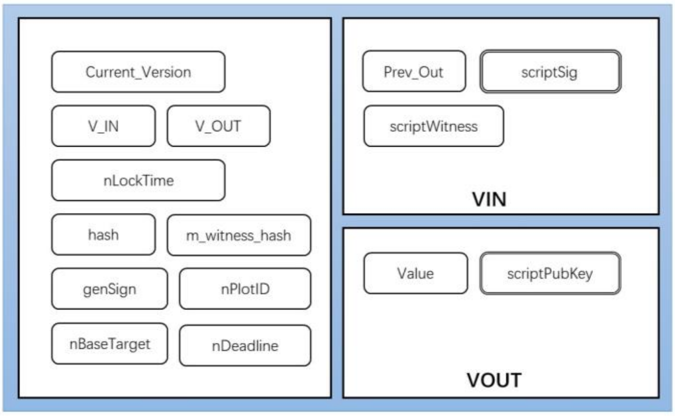
IV. Technology Implementation

A) Data structure of Lava Blockchain

1) Block structure:



2) Transaction structure:



B) Generation Attack

Generation Attack is a kind of attack against networks that adopt storage-based consensus. Malicious miners can on demand generate a large amount of storage resource information through high-efficiency but low-cost software to successfully “deceive” the node authentication at the consensus network layer. This approach allows those malicious miners to make a lot of block gains (the higher the storage space, the higher the returns).

In a PoC-based ecological network, the most important resource for miners is the hard drive (or other storage medium). While how to prevent malicious miners from *Generation Attack*, that is, behaviors of submitting *nonce ID* with the value generated from lower-cost software to cause unfairness in PoC hard-disk ecology, is an important attack scenario every PoC network needs to consider.

In the PoC scenario, the forming of Generation Attack is clearly defined:

- Definition 1.1. (*Generation Attack*): In PoC networks, there are authentication role *Sv*, storage role *Sp*, and resource R declared by *Sp*. In each round of PoC block generation, *Sp* needs *Sv* to authenticate the validity of nonce ID and accurately measure the storage resource R.

Definition 3.1 clearly points out two key points. The first is that *Sp* needs to submit the *nonce ID* within the block time of each round. The second is that *Sv* can clearly measure the storage resource R at the consensus level. Lava invests some deep thinking on these two points:

1) Control over blocktime:

By adjusting the difficulty value *BaseTarget*, Lava limits the expected blocktime to 4 minutes. This value is expected through the *Plotting* to legally produce the *nonce ID* and obtain hard disk scan speed. The deducting process is as follows:

Derivation 1.1. (*BlockTime*): Due to the particularity of the *Shabal256* algorithm (unable to calculate in parallel), if a malicious miner launches a *Generation Attack*, it normally takes 300ms for a single-core CPU common on the market to form a nonce and pass the self-authentication. That means 800 submissions can be generated per round, but valid submissions must be well below 800 (because the value range for deadline is 0~2^64-1, and the valid deadline is 86400). For the current PoC networks, the average 48TB computing power will submit *(2~5)*3* times of *nonce ID* in each round of block generation, as a typical PoC network of hundreds of PB's computing power, so it is almost impossible to attack the computing power of the whole network merely through *Generation Attack*.

2) Measuring storage resource R:

Although there is basically no possibility of *Generation Attack* by malicious miners as long as the first necessary condition is not met, but Lava still considers the measurement of storage resources as another necessary condition to ensure the security and robustness of the entire Lava network.

Definition 1.2. (*Power Estimation*) In PoC networks, there is the following mathematical relationship between storage role *Sp*, authentication role *Sv*, *Sp* ID submission times per round *originalNConf*, submitted ID deadline and the size of storage space estimated by *Sv* for *Sp* plotSize:

$$plotSize=coef(originalNConf)*240*(nConf-1)/\sum_{i=1}^{m} BT_i*Target_i)/GenesisBaseTarget$$

$$coef(x) = 1 - (m - x - 1)/((x + 1) * log(m/(m - x - 1)))$$

- *m*: manually selected chunks, that is, the total number of blocks for 1 day, which is generally set at 360 (based on the expected value of the block time of the whole network);
- *originalNConf*: times of target dataset submission by the miner in *m* blocks.
- nConf: selected times of target dataset submission by the miner in *m* blocks, which is generally considered to be equal to *originalNConf*. And the singularity and repeated data in fastblock are filtered in the datasets.
- Sum *(BT*Target)*: the product sum of the first *m* blocks BaseTarget and Target, where

$$Target = BaseTarget * deadline$$

- GenesisBaseTarget: with fixed value at 18325193796L

The wallet nodes of Lava networks have applied this formula which estimates the computing power of each miner in a scientific and reasonable way and can authenticate each *minerAddress* at the consensus level. This makes it even more difficult for malicious miners to fake computing power through *Generation Attack*.

C) Parasite Chain Attack

Parasite chain attack refers to evil behaviors of malicious wallet nodes, which secretly authenticate each other, generate blocks, form a private chain, transact with an address of the main chain at some time, block-generate the transaction and ultimately pass the mutual authentication without being discovered by the main chain. It will later merge into the main chain and create a "*double spend*", leading to the modification of the transaction while wasting twice the cost of the whole network nodes to authenticate the transaction.

To solve this problem, Lava has introduced the *CDF (Cumulative Difficulty Algorithm)* at the *Blacklist* level.

From the perspective of mathematical modeling, if the entire network nodes or malicious nodes want to attack the PoC network, they will at the consensus level need to overcome the challenge when merged by the main chain, that is, to estimate the cumulative difficulty of the merger at the consensus level, to increase the number of malicious nodes and hence to upscale the cost of the parasite chain acting evil (requiring longer blocks to ensure their cumulative difficulty).

Although we believe that an attack with such a large amount of maliciously-collaborated nodes is unlikely to occur for a large global network, the limit on the number of malicious nodes drawn from *Byzantine Fault Tolerance* is there for the whole Lave network, whatever election algorithm is used.

The trusted layer of Lava has additional security measures. Exploration nodes may select suitable and accessible nodes from the entire set of nodes in the Lava network to form a trusted set. Suppose *N* as the total number of nodes in the Lava network, and *T* as the number of suitable and accessible nodes of trusted layer, where $T \subset N$ (*T* is the subset of *N*). In this way, the maliciously-collaborated nodes will have to accept 33% of the entire Lava network nodes *N*, not just the subset *T*.

In addition, all members of the set *N* are often reviewed as part of the Lava networks, and if they act evil, they will be blacklisted by other nodes in the network, which effectively prevents them from being selected by the subset T.

D) **Blockchain parameters**

- Maximum Supply: 332,800,000 **LV**
- Target BlockTime: 4min
- Issuance Param:
  ·A bitcoin-like halving style of issuance with a fixed halving period;
  · Initial Block Reward (before first halving): 640 LV if a Firestone is consumed by the miner, or 320 LV if not;
  · Halving period: 260,000 Block Height (which last approximately two years);
  · Max Supply will be fulfilled with all of Lava produced full-amount;
- Firestone Slot: per 2048 block height
- Initial distribution*:
  ·Tech team 2%;
  ·Community development start-up fund 2%;
  · Lava Foundation, for mid to long-term ecology development 3%;
  ……Total 7%**

*The initial distribution is completed in the **Genesis Block** as **an additional issuance** against minable coins.

**This percentage is under the assumption that all of Lava (LV) will be produced full-amount, which is 332,800,000 LV.

Reference

[1] Satoshi Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[2] "Technical Information about Mining and Block Forging", [Online] Available: https://burstwiki.org/en/technical-information-about-mining-and-block-forging/

[3] Adam Back, "Hashcash - a denial of service counter-measure," 2002. [Online]. Available: http://www.hashcash.org/papers/hashcash.pdf

[4] "Survey of Consensus Protocols", Abdul Wahab, Waqas Memood.

[5] "Proof of Space from Staked Expanders", Ling Ren, Srinivas Devadas. Massachusetts Institute of Technology, Cambridge, MA renling, devadas@mit.edu

[6] "Subchains: A Technique to Scale Bitcoin and Improve the User Experience", Peter R. Rizun. 2015. [Online] Avalable: https://www.bitcoinunlimited.info/resources/subchains.pdf