LIGHT · FAST · TRUST

TrustNote

The TrustME-PoW Consensus Scheme:

Decentralized, Network Partition Tolerance, Minable

TrustNote Institute of Technology

April 2018

Disclaimer

TrustNote Institute of Technology and Research & Development section hereby declare that, this package is under MIT open source software license and this software distributed without any warranty. TrustNote Institute of Technology declares that we are **NOT** responsible for direct, indirect, incidental, or consequential damages resulting from any defect, error, or failure to perform. This package is **experimental** and a **work-in-progress**, use at your own risk. The contents of this report are in implementation phase, thus TrustNote can update (add/remove packages) any time without informing the users. Finally, we declare that, TrustNote White paper and all other technical reports related to TrustNote **only** can be accessed from:

- https://github.com/trustnote/document
- https://trustnote.org/

We do not guarantee the faulty or misleading data available in documents downloaded from any other website rather than two official websites introduced above.

Contact Us

Business Enquires: foundation@trustnote.org

Technical Support: community@trustnote.org

Copyright

Contents

Glossary

- **Node**: Refers to any active user installed TrustNote client (any version: phone, pc, etc.) and having a valid wallet address.
- **Unit**: Refers to any type of messages generated by the nodes including: Transactions messages, text messages and etc.
- **DAG**: Directed Acyclic Graph is a finite directed graph with no directed cycles.
- **Full Node**: Refers to Cloud Host Server/Workstation, and PC, which maintaining synchronization and verification of ledger data.
- **Super Node**: Refers to Mining Systems, Cloud Host Server/Workstation, and PC, which paying the deposit, and running the TrustME-PoW mining program.
- **Parent Unit**: Refers to units generated at an earlier time and Child Units can reference them.
- **Child Unit**: Refers to units generated at a later time and referencing one or more parent unit.
- **MC**: a single chain along Child-Parent links within the DAG which is determined by applying the Parent Selection Algorithm recursively
- **MCI**: Main Chain Index
- **Attestor**: Refers to a Super node, which participates in a round of consensus and successfully obtains Attestation power.
- **PoW Unit**: Refers to unit containing Equihash solution.
- **TrustME unit**: Used to determine the MC and its first message is a TrustME message.
- **Micro-Node**: Refers to Microcontrollers and Smart Cards.
- **Light Node**: Refers to Smartphone and Tablet PC.
- **Fine-grained PoW consensus**: Each Super Node independently starts PoW, and there is no direct bound between nodes.
- **Coarse-grained PoW consensus**: The Super Node periodically starts PoW. Each round selects a certain number of Super Nodes as Attestors. These Attestors only belong to that specific round. Once that round finishes, the Attestors automatically lose their Attestation powers.

Introduction

TrustNote is a minable public DAG-ledger with an innovative, two-tier consensus mechanism designed to be "lightweight, efficient, and trustworthy". Such two-tier consensus mechanisms can improve transaction throughput and reduce transaction confirmation delay, which effectively solving the problem of "Excessive Bifurcation" and "Double Spending". TrustME-PoW enables support for high concurrency transactions, fast transaction confirmation, and decentralized transaction unit strict sequencing mechanism. Even more, it also provides an important capability for TrustNote to support advanced declarative smart contracts and Micro-Nodes. This report explains super node management protocol and TrustME-PoW consensus mechanism.

1.1 Node Taxonomy and Topology

The TrustNote network supports four types of nodes: Super-nodes, Full-nodes, Light-nodes, and Micro-nodes; these four types of nodes comparison presented in the table below.

Table 0-1 Comparison of four types of node

	Super Node	Full Node	Light Node	Micro-Node
ledger	full ledger	full ledger	light ledger	N/A
transaction	√	√	√	commissioned
DAG consensus	√	√	indirect	×
TrustME-PoW	√	×	×	×
TrustME-BA	√	×	×	×
Hosting Micro-Node	√	×	×	×
deployment	+ Mining Systems + Cloud Host + Server/Workstation + PC	+ Cloud Host + Server/Workstation + PC	+ Smartphone + Tablet PC	+ MCU + Smart Card

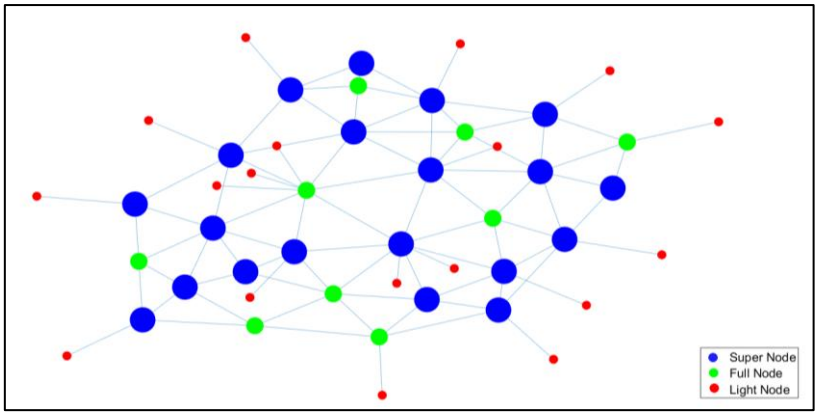

Figure 1-1 P2P network Connection diagram between nodes

1.2 TrustNote Protocol Stack

Table 0-2 TrustNote Protocol Stack

Super-Node Management Protocol	This report				Super Node
TrustME-PoW Consensus Protocol	This report				
Micro-Node Access Protocol	N/A				
Delegate Unit-Creation Protocol	N/A				
P2P Gossip Protocol	N/A			Full Node	
DAG Ledger Synchronization Protocol	N/A		Lite Node		
Units Reference and Validation Protocol	N/A				
Declarative Smart Contract Protocol	N/A				
Cryptographic Algorithms	TR-2018-01	Micro Node			
Micro-Node Security Protocol	N/A				

1.3 TrustME-PoW Scheme Overview

- Decentralized Attestor selection algorithm.
- Users who want to become Super Nodes are required to pay the deposit. Full Nodes can become Super Nodes and join into TrustME-PoW by paying enough TTT (Trust Note's altcoin) to their deposit contracts which would be created by themselves.
- Super Node Clients are enjoying the premium package.
- Super Node Clients will be running multi-threaded platform (Full ledger thread, Super Node thread, Attestation thread, etc.). see figure 1-2

- Full ledger thread let the Super Node clients to synchronize and verify the transaction units.
- Super Node thread let the Super Node clients to run Equihash. Once the super node thread finds the Equihash solution and after a specific procedure, they will receive Attestation power.
- Equihash solutions will be filtered by Difficulty filter. Also, Difficulty Adjustment performed by modified version of Digishield v3/v4 algorithm.
- Attestation thread will be started once Super Node client can establish a stable unit (containing Equihash solution) on the MC of DAG-ledger and its result would be evaluated as high performer.
- Attestation rewards provided for Super Node clients who are having an active Attestation thread in a consensus round. Calculation of the rewards according to the share of issued stable TrustME units located on MC.

TrustME-PoW Protocol Sketch

Full Ledger Thread

On arrival of the new units

(a) Verify whether the unit hash is wrong or right.
(b) Verify that the header fee is correct.
(c) Verify whether message array is empty or not.
(d) Check duplicated units.
(e) Validate message hash tree.
(f) Validate that the parent nodes are reasonable.
(g) Validate Authors and verify the signatures.
(h) Check if the unit is a TrustME unit:
- If it's a TrustME unit, determine and update the MC.

On stabilization of a unit:

(a) Validate and verify the message/s.
(b) Verify the UTXO (if the Node has enough TTT or not).
(c) If the unit fails at any steps the unit header will be stored on the DAG-ledger but the message will be.
(d) Check if the unit is a PoW unit:
- If it's a TrustME unit, update the Attestor list.

On demand from user:

(a) Create and gossip a deposit unit.
(b) Wait the deposit unit to become stable.
(c) Send or wait for another node/s to send enough TTT to the deposit address.
(d) If there are enough TTT in the deposit address starts up the super node thread.

For each transaction about deposit:

(a) If the Super node generated any invalid unit.
- Check silent-locking time
- payout_address
- reward_receiver_address
(b) If the super node, didn't generate any TrustME unit for a certain time.
(c) If any of the above fails, the silent-locking time will be reset.

For each PoW unit created:

(a) Check the PoW solution generated by other Super Nodes.
(b) Wait for twenty PoW solution.
(c) Determine the Attestors' priority in next consensus round.

Super Node Thread

At the beginning of each consensus round:

(a) Calculate the Public seed based on:
$$Seed^{i}_{public} = blake2_{256}(Seed^{i-1}_{public})$$
(b) Generate the node specific seed using public key.
$$Seed^{(i,n)}_{public} = blake2_{256}(Key^{n}_{public} || Seed^{i}_{public})$$
(c) Calculate the difficulty.
(d) Run Equihash and get the solution (TR-2018-01).
(e) Generate and gossip PoW Unit.
(f) Super Node starts the procedure from the beginning again to obtain the opportunity to take part in more consensus rounds.

When the PoW Unit becomes stable:

(a) Check if the Super node will be on top **eighteen** list:
- If on the list, it will receive the Attestation power.
- Determine its own priority.
- Trigger the Attestor thread.

Attestor Thread

At any time:

(a) Generates the TrustME unit according to its priority and should starts sending messages on its turn within five seconds.
(b) Check if there are enough Attestors selected for the next round.
(c) If there is not goes to (a).
(d) If there are enough Attestors for next consensus round, it will be off the list.

When the next consensus round Attestors is determined:

(a) Calculate the Attestation rewards for Attestors of previous consensus round.
(b) Search among the Attestors of current consensus round with the priority equal to one.
(c) The first Attestor of current consensus round sends out the first TrustME unit containing the Coinbase message.

If all the normal units are stable:

(a) Don't generate new TrustME unit until new normal units arrive.

Figure 1-2 TrustME-PoW Protocol Sketch

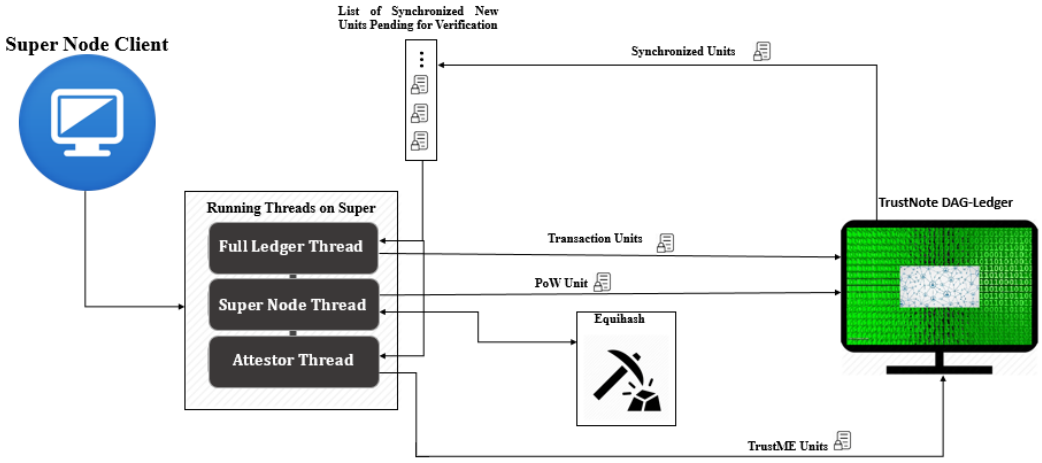

Figure 1-3 Illustration of the underlying mechanism of TrustME-PoW

1.4 Report Organization

The remainder of this technical report is organized as follows. In chapter 2, the importance and methodology of Super nodes will be explained. Also, more detail about how a user can become a super node and the procedure the user should be through explained. Finally, we will explain why the user should pay the deposit to become a Super Node and the procedure the user would be through to get it back. In chapter 3 the innovative TrustME-PoW consensus with all detail will be explained. Even more, the reason why we have Attestors, what Attestors are doing, how Attestors benefit from helping the maintenance of the DAG-ledger and so on explained carefully. The procedure of calculation of the rewards and the procedure of sending rewards to Attestors also explained in this chapter. Finally, in chapter 4 the required steps that TrustNote should be taken to upgrade the network from Witnesses mode to TrustME-PoW mode fully explained.

Super Node

2.1 Motivation

Why do we need super node?

- Select Attestor in a decentralized way.
- Verification and analysis of the new units.

- Increasing the total computing power of the whole network.
- The more super nodes, the higher security of network.
- Super nodes and adoption of a Coarse-grained network algorithm make the whole network, partition tolerant.

2.2 Methodology

Super nodes playing a key role in verification of the transactions, selection of parent units, transaction units sequencing, and micro-node protocols. In order to encourage nodes with high computing storage and network resources to become Super nodes, TrustNote has a total bonus of 500 million MN. The total amount of bonuses is equal to the total circulation in the previous period. These bonuses prized during 30 years to those Super Nodes who will help in maintenance of the TrustNote DAG-ledger.

Full nodes can become Super node as well, while light nodes and micro-nodes cannot become Super nodes. To become a Super node, first it is required to download and install the Super node client program, then pay the deposit contract and assign the address that the deposit will goes to it later. After the Super Node obtains the Attestation power, it must synchronize and verify the rapid growth of the DAG-ledger data, while participating in the competition for the next round of Attestation at the same time. Even more, it will become the agent of the micro node in the future. Therefore, Super nodes need to have high computing performance. Super Node resource consumption:

- Synchronous DAG-ledger consumes network, memory, and hard disk IO resources.
- Verification of ledger consumes CPU, memory, and hard disk IO resources.
- PoW consumes RAM and CPU/GPU resources.
- Database operations consume hard disk IO, CPU, and RAM resources.

2.3 Deposit Mechanism

The Super node can generate normal units and TrustME units. The address of the generated TrustME unit must be the same as the one used to generate the deposit contract. There is only one Super node address for each deposit contract. If a Super node wants to have more than one competing Attestors' address (such as a mining pool), it needs to generate and pay multiple deposit contracts. By considering the safety issues of the Super Node itself and security of the network, the deposit required for the Super Node assigned as 50,000 MN.

The main purpose of the deposit is to guarantee the trustworthy of the super nodes, if any super node tries to violate the rules the deposit value will be reduced; also, it reduces the malicious node's attack on the consensus mechanism. When there is a sufficient number of TTTs in the deposit contract address issued by a certain node, the node can immediately participate in the TrustME consensus to obtain Attestation power in certain rounds and its Attestation rewards. In order to prevent the loss of the Super nodes' deposit and the witch attack due to hacking, the Super nodes deposit has a silent locking time (SLT) when the Super nodes' deposit is frozen in a smart contract. The Super node deposit silent locking time is 17,280 consensus rounds (about 60 days). If the Super node has not sent any TrustME unit or wrong unit within the lock time, all the TTTs in the deposit contract can be transferred to the specific address assigned by that Super node in the smart contract, otherwise the silent locking time reset.

Super nodes are always online and vulnerable to attack. Therefore, a shorter consensus round duration should be set. If it is set to 5 minutes, the probability of the Attestor node being invalid or being hacked is reduced significantly. In order to improve the security of the Super Node funds, the deposit contract sets the deposit payable address (payout_address) and the Attestation bonus receive address (reward_receiver_address), which may be totally unrelated to HD (Hierarchical Deterministic) wallet addresses. Super Node should

introduce different addresses as the deposit_address, payout_address, and reward_receiver_address. Even more, Super Nodes are required to use the cold wallet to increase the security of the Super node deposit and Attestation bonus. In addition, to provide a higher security for Attestors, TrustNote platform made it possible to give the Attestation power to any other node which its address is set in the PoW unit (Equihash message).

Deposit Contract Message

```
messages:[{
        app:'Deposit',
        payload_location:'inline',
        payload_hash:'hash of payload',
        payload:{
                silent-locking time: '17280',
                payout_address: 'Wallet Address of Node',
                reward_receiver_address: 'Wallet Address of Node'
        }
}]
```

Main Chain

In this section it is explained that:

- What the Main Chain is and introducing some basic concepts about it.
- How the main chain is determined and what the best parent is.
- How we determine the best parent and what the procedure for main chain stabilization is and more.

TrustNote does not have a block limitation. Any node can generate a new unit (e.g. a transaction) at any time and broadcast it to other nodes. The unit can select one or more previous units as parent units, each unit validates the parent units and adds the parent units' hash value to its own header. In this manner, the parent-child relationship between them can be expanded into a directed acyclic graph (DAG). Directed Acyclic Graph (DAG) is a finite directed graph with a topological

ordering (a sequence of the vertices such that every edge is directed from earlier to later in the sequence). The Acyclic property of the graph means there is no way to start at any unit u and follow a sequence of edges that finally loops back to u again.

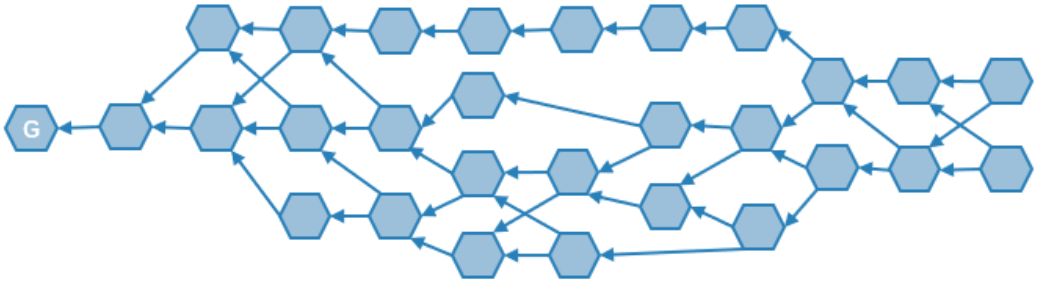

Figure 0-1TrustNote simplified DAG-ledger

Assume an attacker wants to modify a unit. The attacker will be successful if it would have conspiracy with all other nodes to gossip the new ledger which containing the double-spent unit; which with the existence of honest nodes it's impossible. Even more, assuming that the digital signature is unforgeable, the greater the number of nodes on the entire network, the more difficult it is for an attacker to tamper a unit. Consequently, It's about to impossible for an attacker to do so.

3.1 Basic Concept

- **Unit Inclusion**: If "A" verifies "B", the unit "A" header contains the hash of the unit "B", so we say the unit "A" contains the unit "B" which indicates that the unit "B" can be reached from the unit "A" in the direction of the arrow. This inclusion relationship between the nodes is represented by an arrow between vertices "A" and "B" on the DAG. If unit "A" contains "B" and the minimum path between the two units is "1", then "A" is said to contain "B" directly; if unit "A" contains unit "B" and the minimum path between the two units is greater than "1", then unit "A" contains the unit "B" indirectly. Examples of unit inclusions is presented in figure 3-2 (a and b) below.
- **Parent Units and Child Units**: If unit "A" directly contains "B", we say unit "A" is a child of unit "B", and unit "B" is the parent of unit "A".

- **Genesis Unit**: The genesis unit has no parent and is the first unit in TrustNote DAG indicated with the letter “G” in figure 3-2.
- **Childless Unit**: Units without child units are called childless units, which are also non-validated units. Unit “A” in figure 3-2 is an example of a childless unit.

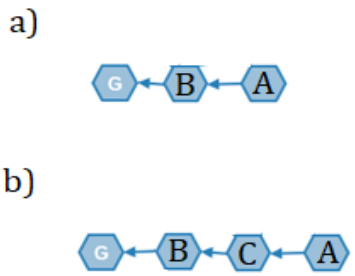

Figure 0-2 Examples of inclusion relationship between units: a) Unit "A" verifies and contains unit "B" directly Unit "B" is the parent of unit “A”. b) Unit “A” contains unit “B” indirectly and unit “A” is called a childless unit.

- **Sequential Units**: Any two desired units are defined as sequential units if they would have a directed path connecting those units, otherwise they are called non-sequential units. All the units sent by an Attestors from a certain address must be sequential units, otherwise they will eliminated from the DAG.
- **Unit Level**: The unit level is defined as the longest path length from that unit to Genesis unit.
- **Attestation Level**: To determine the Attestation Level for any unit, first it will be assumed, that unit as the starting point, and follow a path along the best parent chain, until finding more than half of the whole Attestors’ Attestation unit along the path. Then calculating the unit level of the last unit on this path, this value is the Attestation Level of the starting unit. The genesis unit contains twelve Attestors address so it’s the best parent naturally. The other units Attestation level (only at the beginning) will be zero as the Genesis unit level is equal to zero. This process will continue until enough Attestation units (rather than Genesis unit) satisfy our condition and will be set as last unit.

3.2 Main Chain Determination

Each unit of a DAG (except Genesis unit) has one or more parent units. According to certain rules, we can select the best parent units among the available parent units set. The best parent unit selection algorithm can be explained as below:

1. Starting from the childless units.
2. Selecting the parent units with the highest Attestation level as the best parent unit.
3. If there are multiple candidate units, the unit with the lowest unit level will be selected as the best parent unit;
4. If there are still multiple candidate units, the unit with the smallest unit hash is the best parent.

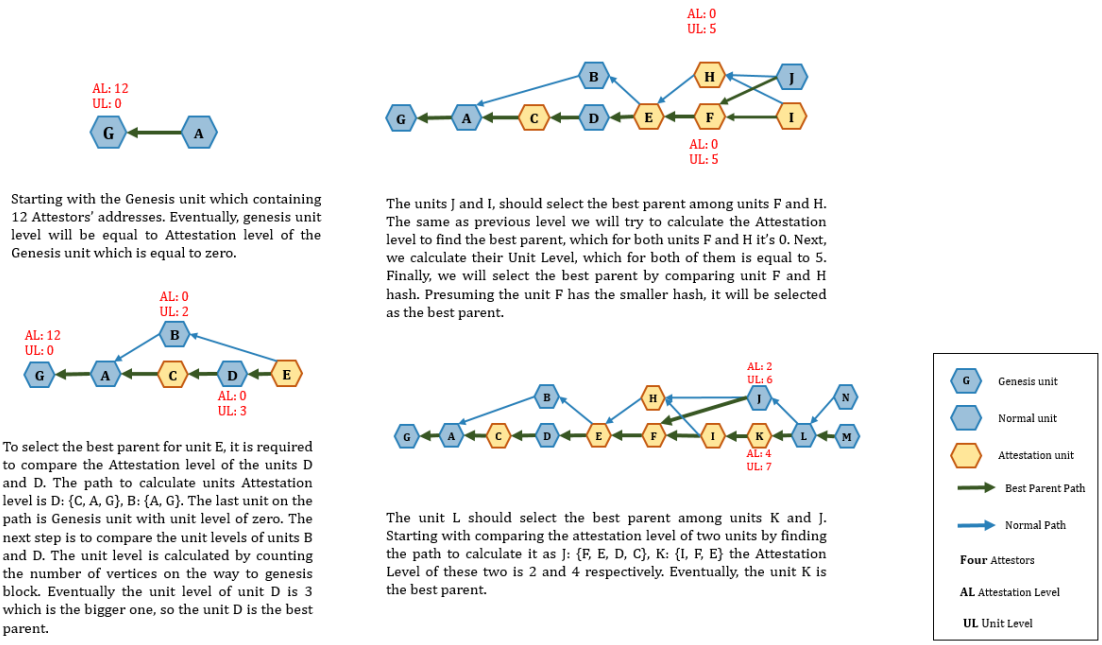

Figure 3-3 the best parent selection explained in four slides.

In this way, you can get a path starting from any Childless Unit, connecting the selected best parent units and finally reaching the Genesis Unit. This path is called the "Main Chain". As many childless units Exist, there might be many "Main Chains" but the one which starts from the Childless Unit with the highest Attestation Level finally will be selected as Main Chain.

3.3 Main Chain Stabilization

According to the Best Parent Unit selection algorithm, all candidate must be at a merging point with the last stable unit of the main chain. The parts from the merging point to the Genesis Units completely coincide. If the addition of new child unit does not change completely the overlapping part, then the overlapping part is considered to be stable, and the main chain from the merging point to the Genesis Unit is called the stable main chain. The last unit to stabilize the main chain is called the Last Stable Unit. All units that are contained directly or indirectly by the stable main chain element can strictly discharge the total order according to the stable main chain, and then use the total order to prevent double spending. This process is called "unit final confirmation". With the addition of new units, DAG continues to grow and the stable main chain will continue to advance. The new units in the DAG will be finalized in turn.

3.4 Main Chain Stabilization Algorithm

The main chain stabilization algorithm uses the Current Main Chain as a reference to determine whether the final stable unit can be advanced to the next unit of the main chain or not. According to the number of child units of the last stable unit, this algorithm can be divided into two cases:

Case 1: The last stable unit has only one child unit

Assuming that the unit level of the only child unit of the last stable unit is UL. Now, tracing back along the Current Main Chain from the child unit with the highest Attestation level, until more than half of the current round of notarized notarization units are encountered, find the minimum Attestation level of these Attestation units, denoted as min_{AL}. If $min_{AL} > UL$, the last stable unit is advanced to this unique child unit, otherwise it is not advanced.

Case 2: The last stable unit has multiple child unit

If these child units that are not on the current main chain are not the optimal parent for the other units, they are processed as in Case 1. Otherwise, in accordance with the reverse of the best parent node in reverse order to get a complete alt-branch, and then their maximum unit level, called max_{UL}. If $max_{UL} < min_{AL}$, we can push the stable node along the current main chain, otherwise it will not advance. Even if less than half of the other Attestors all support a candidate unit for main chain, the Attestation level of any unit on the main chain will not exceed max_{UL}. This candidate unit for main chain has absolutely no chance of becoming a stable unit on main chain.

3.5 Main Chain Index (MCI)

- The MCI of the Genesis unit is 0, which increases sequentially along the main chain to obtain the MCI of the main chain units.
- The MCI of a unit that is not on the main chain is equal to the minimum MCI of the main chain unit that contains that unit.

TrustME-PoW Consensus

TrustNote adopts a two-tier consensus mechanism comprising “base consensus” and “attested consensus”. The base consensus, also known as “DAG consensus”, requires new transaction units to be sent out by nodes to verify and reference previous transaction units. The attested consensus, or “TrustME Consensus”, requires that the sequences of Non-Attested units be rigorously determined by TrustME units generated from the Attestor Nodes. Such two-tier consensus mechanisms can improve transaction throughput and reduce transaction confirmation delay, thus effectively solving the problem of Excessive Bifurcation and double-spending.

4.1 Motivation

TrustME-PoW designed in two different type during the conceptual design phase:

- Fine-grained

Coarse-grained

Both methods studied carefully, but in fine-grained PoW schemes, there is not any bound between Super Nodes necessarily. In the case of network-partitioning, as long as there are Super nodes in each partition, MC growth and stability will still be promoted, which will lead to successful confirmation of a double spent unit.

To resolve this issue some attributes suggested for strengthening the bounds between the Super nodes. Eventually, it observed that nodes' bounding that achieves a fine-grained consensus is equivalent to transforming the fine-grained consensus into a coarse-grained consensus. Therefore, in final design Coarse-grained PoW selected.

In addition, when the network is divided, networks with more than $\left(\frac{n}{2}\right)$ Attestors will continue to expand, and the main chain will continue to grow and stabilize over time. If in the partitioned network, the attestor number is less than or equal to $\left(\frac{n}{2}\right)$, all the units in the network will not become stable and a new round of TrustME-PoW consensus cannot be completed until the network get more than $\left(\frac{n}{2}\right)$ Attestors.

4.2 How to Select Attestors

TrustME-PoW conducted in separate rounds. Each round lasts **five** minutes. **Eighteen** Attestors will be selected in **each** round and **two** Attestors will be added to the consensus round from **TrustNote** which results in **twenty** Attestors in each consensus round.

4.3 PoW Unit

The Super node thread runs the Equihash algorithm based on the seed and difficulty; it generates a PoW unit containing Equihash solution and publishes it to the DAG-ledger. The first eighteen addresses

owning a stable PoW unit on MC will receive Attestation power. A difficulty factor "**d**" will be designed and adjusted for each consensus round based on desired time of **five** minutes to find the solution of Equihash. It means this parameter is a threshold and the result of solving Equihash must have a certain number of leading zeroes to be accepted. This parameter will be adjusted with respect to desired time of generating a PoW unit and it will be applied to the system smoothly. If there are multiple Super nodes submitting the PoW unit at the same time, the priority is determined according to the MCI of the PoW units. If the MCI is the same, the smaller the unit hash is, the higher the priority it has. In each round of consensus process, when there are **n** stable PoW units on the Main Chain, the Super nodes corresponding to the n stable PoW units will be given an Attestation Power.

PoW Message

```
messages: [{
        app: 'PoW-Equihash',
        payload_location: 'inline',
        payload_hash: 'hash of payload',
        payload: {
                round: 'round number',
                seed: 'string of seed',
                difficulty: '',
                solution: '',
                attestor_address: 'Wallet Address of Node'
        }
}]
```

4.4 Equihash Difficulty Calculation

Difficulty Calculation
Super Node thread • INPUTS: - Target round time: T = 5 min - New consensus round number: i • OUTPUT: Difficulty in the next round D_i (a) Calculate the number of previous rounds to be considered in the calculation of difficulty: $PB = floor(f(T))$ (b) Calculate the Sum of Average PoW round time: - $SumAvgPoW = \sum_{i-N+1}^{i} \min[(T \times C_1), (AvgPoW[j] - AvgPoW[j-1])]$

(c) Calculate the Gross Average PoW Time: $GAPW = f(N, T, SumAvgPoW)$
(d) Calculating the difficulty for next round: $D_i = f(T, GAPW, SumAvgPoW)$

4.5 TrustME unit

An Attestor can send multiple TrustME units within its consensus round. The TrustME unit will be selected as the MC unit with a large probability. Each round of consensus will select multiple Attestors. Each Attestor sends TrustME units according to the priority order. To reduce the delay in confirmation of the transactions, if any of the Attestors don't send the TrustME message during a certain time the next two Attestors can send the TrustME unit.

TrustME unit Sending Mechanism

Attestation threads

At any Consensus round:

(a) Top twenty Attestors, aware of their priority joining the consensus round.
(b) First Attestor should sends its first TrustME unit within 5 seconds, otherwise it will be replaced with the next **eighteen** Attestors on the list.
(c) First Attestor on the list send the first TrustME unit.
(d) The first TrustME unit contains the Coinbase list of rewards for previous consensus rounds which will be sent by the first Attestor.
(e) All the Attestors should send their units with respect to priority list.
(f) The consensus round finishes and (a) will be performed again.

- If TrustNote Attestors or any of other eighteen Attestors in the consensus round wouldn't be available for any reason the next priority Attestor in current consensus round will immediately start sending TrustME units.

TrustME Message

```
messages: [{
        app: 'TrustME',
        payload_location: 'inline',
        payload_hash: 'hash of payload',
        payload: {
                round: 'round number',
                PoW_solution: 'the Equihash unit address',
                priority: 'priority of notary',
        }
}]
```

Coinbase Message

```
messages: [{
        app: 'Coinbase',
        payload_location: 'inline',
        payload_hash: 'hash of payload',
        payload: {
                output：  [
                        {address: '...', amount:21 MN},
                        {address: '...', amount:19 MN},
                        ...
                ]
        }
}]
```

4.6 Attestation Reward

If the TrustME unit sent by the Attestor of a consensus round is a stable unit on MC, the Attestor will receive Attestation reward and the larger the proportion of the TrustME units it sent, the more Attestation reward it would receive. In the absence of stability in the main chain, it is impossible to determine which TrustME units are on the main chain and it is not fair to calculate the Attestation bonus. Therefore, this consensus mechanism does not provide Coinbase directly (like Bitcoin) in the unit. Only after all the TrustME units belonging to the same round on the main chain are stable, TrustNote Attestors calculate how many Attestation bonuses can be obtained for each Attestor in that specific round; the reason for this is to eliminate the cost of generation of this unit for Attestors. After all the TrustME units of a consensus round are stable, the number of TrustME units at each address in the statistical chain shall be calculated according to the proportion of the Attestation bonuses. Even more, if any Attestors has doubt in calculated share of Attestation reward, they can generate the Attestation share list of that specific round by themselves and reference the TrustNote unit to make sure.

Coinbase calculation
full ledger thread • INPUTS: - List of Attestors: $\{A_1, \dots, A_{20}\}$ - Round number: i - Reward of consensus round i: R_T^i • OUTPUT: list of the share of each Attestor from the Attestation rewards round i $\{P_{A_1}^i, \dots, P_{A_{20}}^i\}$ (a) Calculate total number of Stable units on MC in round i: P_T^i (b) Calculate the number of stable TrustME units on MC generated by each Attestor $\{A_1, \dots, A_{20}\}$: $\{MCU_{A_1}^i, \dots, MCU_{A_{20}}^i\}$ (c) Calculate the share of each Attestor: $P_{A_j}^i = \frac{MCU_{A_j}^i}{P_T^i} \times R_T^i$ $(j = 1, \dots, 20)$

Switch from Witnesses to TrustME-PoW

5.1 Overview

At a certain time, the i[th] round's Attestors are on duty.

Round Number	**Attestor**	**TrustME unit**
$i-1$	invalid	being stable
i	sending TrustME units until all units in round $(i-1)$ has been stable and there are enough Attestors for round $(i+1)$	not stable
$i+1$	waiting until the first TrustME unit of round (i) to become stable, and making consensus with the seed of round (i) and the Coinbase of round $(i-1)$	N/A

⚠ At the junction, there might be cases where the i^{th} TrustME unit and the $(i+1)^{th}$ round of TrustME unit are juxtaposed. At this time, the TrustME unit of the $(i+1)^{th}$ round Attestor calculates

the optimal parent node.

- The first $(i + 1)$ rounds and all rounds of Equihash are sent again after the MC becomes stable. This is equivalent to recognition of this consensus round.
- The first round of TrustME unit will not refer to the second round of TrustME unit at the junction because it violates the rules for the removal of Attestation power.

5.2 Procedure

This section gives out the list of steps about how to convert to the TrustME-PoW consensus mechanism.

Step	Description
1	Wait for a MC cell with a specific MCI value (e.g. 1,000,000) to become stable.
2	The R_1 consensus round[1] is initiated, R_1 public seed is a constant number and it will be selected randomly, and then the respective seeds of the R_1 round super node are calculated according to the R_1 seed and the super node's own address, with respective seeds and initials. The difficulty factor is implemented as an input priority to Equihash and its initial value is 1.
3	The super node proceeding to take part in the R_1 consensus round should issue one PoW unit.
4	Wait for Witnesses to make first eighteen PoW units to become stable and determine the top twenty Attestors for R_1 round.
5	Witnesses consensus mechanism is deactivated, and the R_1 Attestors start sending the TrustME units.
6	Wait for the first TrustME unit of R_1 to become stable on MC.

[1] First consensus round after switching to TrustME-PoW.

Step	Description
7	The super nodes proceeding to attend the R_2 consensus round, calculates the R_2 public seed based on the public seed of the R_1. The super node uses the R_2 public seed to calculate the node specific seed. The super node using difficulty and node specific seed to calculate the Equihash solution.
8	The super node proceeding to take part in the R_2 consensus round should issue one PoW unit.
9	Wait for Attestor of the R_1 to make first twenty PoW units to become stable and determine the top twenty Attestors for R_2.
10	The R_1's Attestors end the mission, and the R_2 Attestors start sending the TrustME units. The first TrustME unit includes the Coinbase result for the Attestors of R_1.
11	Wait for the first R_2's TrustME units to become stable on MC.
12	...

Eventually, all the consensus rounds after R_2 following the same procedure as R_2.