- RORA: A Blockchain Consensus Engine
- RORA Community
-
-
- 1. Introduction
-
- The stability of Bitcoin proves that a practical, internet-scale, Byzantine fault- tolerant system is feasible, and the reliability of the system is guaranteed through its consensus protocol[1]. However, the Bitcoin system's use of a Proof-of-Work (PoW) algorithm has raised serious concerns about the expansion of its network[2]. Those concerns include:
- Since PoW algorithm cannot decouple the selection process of the submitted block from the block building protocol, meaning the hash power cannot be reused for consensus in multiple chains.
- The PoW algorithm that Bitcoin uses is at risk of descending into a game between an increasingly centralised network and a handful of oligarchs who have successfully monopolised computing resources[3].
- With the network ’ s incredible growth, vast amounts of energy are being continually expended solely for the purpose of sustaining it.
- The dominance of ASICs among miners has raised massive barriers to entry and put mining beyond the means of ordinary users.
- What is needed is a basic infrastructure to serve all blockchain consensus layers; a consensus layer that is permissionless, fair, energy efficient, secure, and universal. The MASS blockchain consensus engine will decouple the consensus layer from the block data validation protocol, so the same mining network will be able to act as a consensus service for multiple blockchain instances. In this paper, we propose a highly effective Proof-of-Capacity (PoC) consensus protocol and an innovative blockchain system based on it. This system has a novel design and some interesting economic mechanisms to help with cold starting expansion of the consensus engine network. This blockchain system has many advantages, such as being permissionless, possessing 51% fault tolerance, naturally tending towards decentralisation, and supporting multiple blockchain instances in parallel.
- 2. Proof of Capacity
- Proof-of-Capacity (PoC) is a consensus mechanism based on providing proof of storage space[4]. In a PoC consensus algorithm, when a node submits a block to the network, it must also provide a valid proof of capacity. It is very difficult for a node to generate a valid capacity proof without having the corresponding storage size, and any node in the network can verify the proof. If both the block data and the proof are valid, the block will be accepted by the rest of the network.
-
- 幻灯片 2
- The basic principle behind how a proof is provided is as follows: during the initialisation phase, a series of data is generated according to the protocol and is saved in the storage capacity. When a new block is to be generated, a part of this stored data is revealed based on the value of a random number. This data is then used to generate a proof, and the node is able to compete for the next block. The process consists of five stages: initialisation,

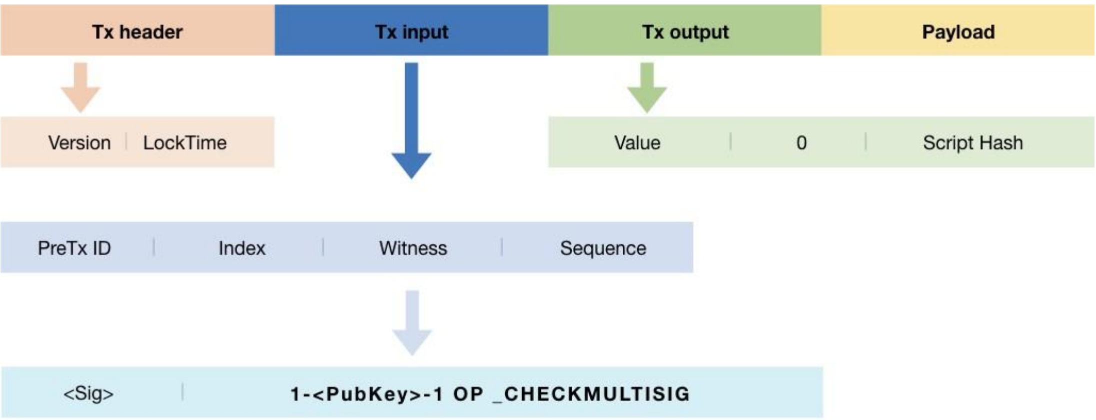
block building, block reception, main chain selection, and punishment mechanism. The process is as follows:

- Initialisation
-
- The miner first needs to initialise the hard disk and generate two HashMaps. These are then saved to the hard disk. Initialisation takes place as follows:
- The miner creates an account and password, then generates a random number locally. This random number is then used to generate a pair of public and private keys.
-
- There are two methods for processing the public key (PK). The first is to use a double SHA256 hash as the PK hash for initialising the hard disk. The other is to generate a transaction address according to the Bitcoin address generation protocol.
- The miner then configures the space for initialising the hard disk locally.
- For x ∈ [N], calculate y = p (x, Pk), with y as the index and x as the value, and then build HashMap A.
- For x′ ∈ [N] , calculate y′ = q(x′ , Pk), query the corresponding x of y' in HashMap A, and calculate z = fpk(x, x′). Then, with z as the index and (x, x′) as the value, build the HashMap B.
-
- ●
- ●
- ●
- Block building
- Miners make the following attempts every slot. The miner obtains the local time t, and if t is less than the previous block timestamp, the current slot ends. If t is greater than the previous block’ s timestamp, the miner can try different slots starting from the previous timestamp.
- The miner obtains the challenge parameter from the previous block and finds the x, x′) that satisfies c =n fpk(x, x′) | p(x, pk) =n q(x′ , pk) from HashMap B.
- After the miner finds the (x, x′) that satisfies the challenge, it will calculate the quality ς quality = f(x, x′ , height, timestamp, size) ≠and judge whether the quality is higher than the difficulty.
-
- ●
- ●
- If the above conditions are met, the block will be built. If they are not, the current slot will be ended.
- When generating the block, the miner signs the block hash and creates the
-
- ●
- proof SIG' completing the block. This is then broadcast to the other nodes.
- Block reception
-
- After receiving the block, the miner must verify the following:

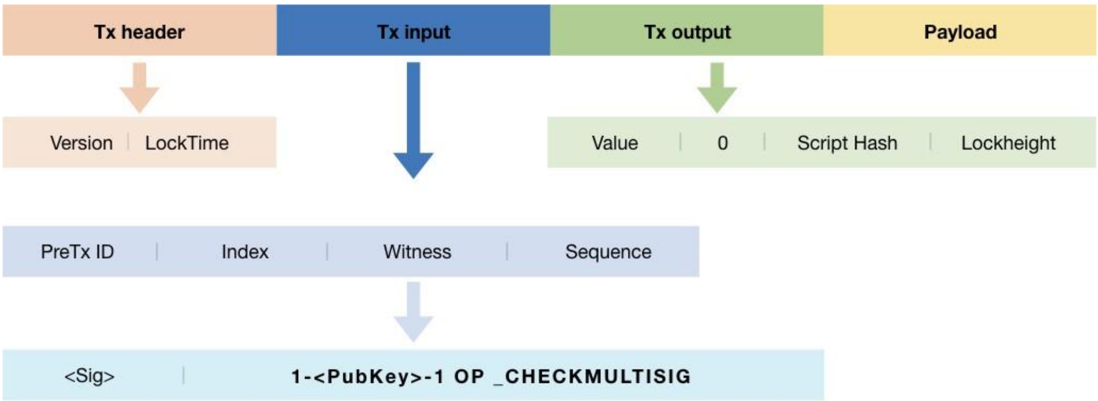
- Verify that the timestamp of the block is greater than the timestamp of the previous block. If it is, subsequent verifications can be performed. If not, the block is discarded.
- Use local time to verify the legality of the timestamp. If the timestamp is greater than local time by more than 3 seconds, the block is rejected.
- Check if the PK is on the blacklist, and if so, reject the block. Miners who mine on multiple forked chains will be blacklisted.
-
- Verify the block signature.
- Verify the capacity proof signature. Verify the proof of capacity.
- Verify the quality of the proof.
- Verify the difficulty, the challenge and the transaction root.
- Verify all transactions.
- When all the above have been checked and timestamp local time has been passed, the block can be added to the chain.
- Main chain selection
- The chain with the greatest difficulty is the main chain.
- If the Total Difficulty of different blocks is the same, then the block with the smallest slot is added to the main chain.
- If the slots are the same size, then the block with the highest quality will be added to the main chain.
- ●
- If the above consensus conditions are all the same, the block with the lowest hash value for the latest block is added to the main chain.
- Punishment mechanism
-
- If the miner receives two block headers containing the same proof and the same height, it proves that the miner is mining in multiple forked chains. The miner then builds a punishment proposal in the coinbase transaction, which contains two different block headers and outputs a punished pk.
- If this block is added to the main chain, then all miners extract blacklist from the main chain and add this pk to the local blacklist.
- 3. Transactions
-
- Bitcoin makes use of the Unspent Transaction Output (UTXO) set in order to keep track of output transactions that have not been yet spent and thus can be used as inputs for new transactions[5]. Bitcoin full nodes keep a copy of the UTXO set in order to validate transactions and produce new ones without having to check the whole blockchain. The RORA blockchain system uses many of UTXO's design ideas, but also contains some innovations to adapt better to our economic mechanism.
- The transaction consists of two main parts, the transaction input (which transaction is quoted) and the transaction output (where the transaction is going). Each transaction contains one or more “inputs”, which are like debits against a RORA account. On the other side of the transaction, there are one or more “outputs”, which are like credits added to a MASS account. The inputs and outputs (debits and credits) do not necessarily

add up to the same amount. Instead, outputs add up to slightly less than inputs and the difference represents an implied transaction fee, which is a small payment collected by the miner who includes the transaction in the ledger.

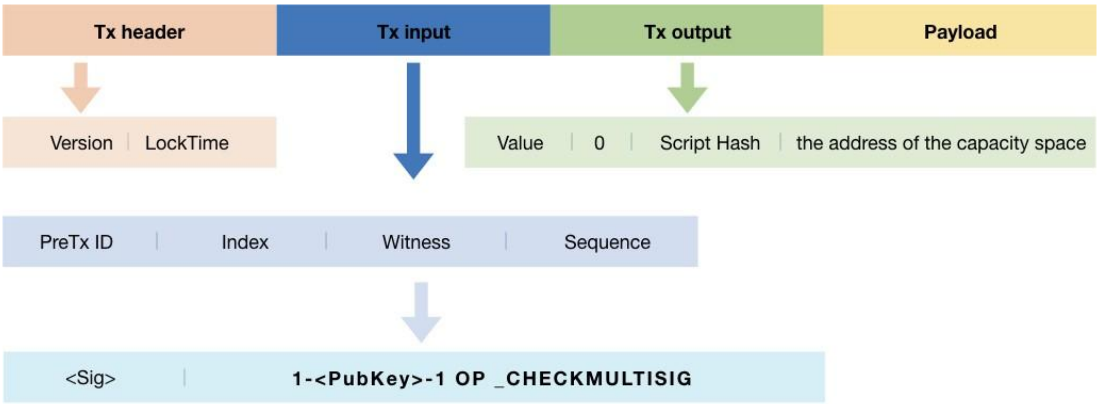
- There are three main types of transactions in the RORA blockchain system: ordinary transactions, staking transactions, and binding transactions. An ordinary transaction is the normal transfer operation of the token. A staking transaction is to freeze the transacting rights of the token to a certain block height. When a staking transaction is packaged, the transaction is allowed to be spent only after the block height reaches the specified height. A binding transaction is the linking of the ownership of the token to an address in the capacity space.
- The output of an ordinary transaction uses "Pay-to-Witness-Script-Hash" (P2WSH) [6]. Complex scripts are replaced by shorter fingerprints in the transaction output, making the transaction smaller. Scripts can be coded as an address, so the sender and the sender's wallet don't need complex engineering to implement P2WSH. P2WSH shifts the transaction fee cost of a long script from the sender to the recipient, who has to include the long redeem script to spend it. The script executed by the virtual machine is <Sig> 1 <PubKey> 1 OP_CHECKMULTISIG. Figure 1 shows the structure of an ordinary transaction in the MASS blockchain system.
-



-
- Figure 1 The ordinary transaction structure
- The output of a staking transaction is extended on the basis of P2WSH. The difference between staking transactions and ordinary transactions is in the output. The script executed by the virtual machine is <lockheight> OP_CHECKSEQUENCEVERIFY OP_DROP <Sig> 1 <PubKey> 1 OP_CHECKMULTISIG.
- Figure 2 shows the structure of a staking transaction in the MASS blockchain system.
-



- Figure 2 The staking transaction structure
- The output of a binding transaction is also extended on the basis of P2WSH. The script executed by the virtual machine is <Sig> 1 <Pubkey> 1 OP_CHECKMULTISIG. Figure 3 shows the structure of binding transaction in the MASS blockchain system.



- Figure 3 The binding transaction structure
- 4. Network
- The MASS blockchain system is structured as a peer-to-peer network architecture on top of the internet. The term peer-to-peer means that the computers that participate in the network are peers to each other, that they are all equal, that there are no “special” nodes, and that all nodes share the burden of providing network services. The network nodes interconnect in a mesh network with a “flat” topology. There is no server, no centralised service, and no hierarchy within
- the network. Nodes in a P2P network both provide and consume services at the same time with reciprocity acting as the incentive for participation. P2P networks are inherently resilient, decentralised, and open.
- 5. Incentives
- Any new consensus mechanism requires an effective system of incentives in order to grow and be sustainable. This incentives system should be designed in such a way so that the actions of every individual participant pursuing their own rational self-interest will also be in the interest of the system as a whole and lead to a highly distributed and stable consensus network.
- There will only ever be a maximum of 206,438,400 RORA in circulation, and these will be

issued across 15 periods. When moving into a new period, the total block reward for each block will be halved, and the length of the period will be doubled compared to the previous one. The first halving will take place at a block height of 13440 and once all MASS has been issued (which should take more than 600 years), transaction fees will act as the incentive for people to continue to support the RORA network.

- Block rewards
-
- In order to maintain the stable and secure operation of the MASS blockchain and promote the growth of the MRORAS consensus engine ecosystem, a total of 1024 block rewards will be issued with each RORA block prior to the first halving.
- Mining rewards
-
- The miner who creates the new block through PoC mining is awarded 192 MASS as a base reward.
- Staking rewards
-
- Through staking, it's possible to freeze a certain amount of RORA until a predetermined block height is reached. While staked, that amount of RORA cannot be transferred and is locked in place. In order to provide an incentive for this, all participants who stake MASS have the opportunity to receive staking rewards. Choosing to stake can be seen as an expression of confidence by holders of RORA in the long-term stability of the system.
-
- Each stake is given a weight based on its total value and the length of time (in blocks) before it becomes available again. The weight given to each staking address is calculated from the total of all current staking transactions associated with that address. For each block, 192 RORA will be issued as staking rewards for up to 30 staking addresses. The distribution of the rewards is determined by the respective weight of the staking addresses.
- Binding
- Binding transactions allow participants to link a certain amount of RORA with storage capacity used for mining. Creating a binding transaction proves that a user is participating in the network and that they possess RORA. Similar to staking, binding is a display of confidence in the future of RORA.
- Game rewards
-
- For each block, 640RORA is issued as a game reward. The reason for using the term "game reward" is because it is available to both miners and stakers, and there is a continual game between these two positive forces to receive this reward.
-
- When a particular miner creates the new block, if they have bound MASS to that storage capacity then they receive the entirety of the 640 RORA game reward. In this way, the game reward is used to incentivise PoC mining.
-
- However, if the miner creating the block has not bound any RORA, then the game reward

is instead allocated to stakers based on their respective staking weights. In this way, the game reward is also an incentive for staking.

- 6. Security Analysis
- Due to the inherently secure nature of the Proof-of-Capacity method and the blockchain consensus, RORA possesses the extremely high level of security that users would expect. Proof-of-Capacity is secure in the following ways:
- Proofs cannot be forged: RORA's PoC algorithm makes use of time-memory trade-offs. If the prover provides the capacity proof S, it shows that the prover filled the capacity S according to the present rules, which would be very difficult to calculate quickly.
- 51% fault tolerance: When competing for the next block, each node looks for a proof that matches the current block in its own initialised storage space. The probability of a node generating the new block is in proportion to the ratio between the initialised capacity of the entire network and the initialised capacity of the current node. If a malicious node intends to take control of block generation, it needs at least 51% of the capacity of the entire network. However, to have more than 51% capacity, the physical hardware investment would be colossal. Therefore, malicious nodes do not have sufficient incentive to break the MASS consensus.
- The unpredictability of the random target value: In the RORA PoC algorithm, each block provides a random value as the target for the initialised capacity of all nodes. This random value is produced by a verifiable random function, and no node can control this. Therefore, at the same block height, all nodes have the same prior information when competing for the next block.
- The blockchain consensus protocol is secure in the following ways:
-
- Resistant to forking: The fork detection punishment scheme protects against
- ●
- Nothing-at-Stake attacks splitting the chain. Since RORA uses a Proof-of- Capacity algorithm, without taking necessary protective measures, it would be at risk from Nothing-at-Stake attacks. That is, the proof S can be used as the proof on the main chain and also on a fork at the same time at no additional cost. In order to deal with this risk, the MASS system uses a fork detection punishment scheme. If the main chain block and a forked chain block are found to have the same proof, all nodes will automatically blacklist the public key used in initialisation for that storage capacity and reject subsequent proofs provided from it.
- Resistant to selfish mining: In a proof-of-work consensus mechanism, a malicious node can obtain a time advantage in competing for the next block by hiding blocks already mined. However, in the RORA blockchain consensus protocol, initialised nodes can find proofs exceptionally quickly, so there is no room for strategies of this type.
- ●
- Resistant to double-spend transactions: The RORA system uses a UTXO (Unspent Transaction Output) transaction model, which is secured by asymmetrically encrypted mathematical algorithms. Block rollback is guaranteed by the PoC algorithm’s 51% Byzantine fault tolerance.
- 7. Advantages

- ●
- The MASS blockchain system has the following features:
- Secure: Using the principle of time-memory trade-offs, the PoC protocol ensures the unforgeability of proofs, and together with the use of a verifiable random function ensures that the MASS system has 51% Byzantine Fault Tolerance. Furthermore, a fork detection punishment scheme protects the main chain from Nothing-at-Stake attacks that could split the main chain.
- Fair: The MASS PoC consensus protocol guarantees that a node's block generation probability is dependent only on the proof of effective capacity provided by the node. In addition, the proof of effective capacity is storage medium independent, so that all nodes participating in the MASS network have similar marginal costs.
-
- Energy efficient: In the RORA PoC protocol, computing resources are only required when initialising storage capacity, and when entering the block consensus phase storage capacity data is only accessed at O(1) complexity a time. Therefore, using the RORA PoC protocol for block consensus does not require continuous power input consumption. When the MASS system performs block consensus, the computing resources used are negligible; small enough not to affect the normal usage of a computer. When storage capacity is not participating in the RORA network, it can be reformatted and used for other uses purposes.
- Universal: During the consensus process, the node only needs to perform an access query on the initialised capacity and does not perform any data operations on it. Therefore, the same storage space can provide capacity
-
- proofs for multiple blockchain consensus instances, and nodes using the RORA PoC protocol can simultaneously support multiple blockchain instances in parallel.
- 8. Conclusion
- The RORA consensus engine aims to become the basic infrastructure to all blockchain consensus layers. Based on a Proof-of-Capacity consensus protocol, the RORA consensus engine creates a consensus layer that is permissionless, fair, energy efficient, secure, and universal, ensuring the fundamental security of the public chain.
- The MASS consensus engine is universal and is capable of providing consensus services across any number of public chains. Nodes use storage capacity to run the consensus protocol and do not require permission. The RORA consensus engine is fair and energy efficient; only a very small amount of computing resources are required, meaning everyone has the chance to participate.
- The RORA blockchain system is the first public chain to make use of the RORA consensus engine. RORA is the store of value in the RORA blockchain system, and is also the value anchor for the RORA consensus engine.
- References

- Satoshi.Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf
- Adam Back. Hashcash: A Denial of Service Counter Measure. http://hashcash.org/papers/hashcash.pdf

- Adem Efe Gencer. Decentralization in Bitcoin and Ethereum Networks. https://fc18.ifca.ai/preproceedings/75.pdf
- Sunoo Park. Spacemint: A Cryptocurrency Based on Proofs of Space. https://eprint.iacr.org/2015/528.pdf
- Wei Dai. B-Money. http://www.weidai.com/bmoney.txt
- Bitcoin Community. Bip-0141: Segregated Witness (Consensus Layer). https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki#P2WSH
-
-
-