An Efficient Proof-of-Work Mechanism for Computational Feasibility in Blockchain

Madhu Babu Janjanam, Siva Prasad Pinnamaneni, Prashant Atmakuri

Abstract— Recently, Blockchain technology created a buzz in technological world and gained importance as a solution that offers the realization of multiple authoritative domains, where economic transactions are guaranteed. It is very important for such technology to validate the transactions performed by different users flawlessly. To maintain such a trust, Blockchain uses different consensus protocols, where different miners follow them to validate transactions and mine a new block. One such consensus protocol is proof-of-work, where a miner has to solve a puzzle with his computational capabilities. The existing proof-of work mechanisms discriminates a normal user to participate in the mining procedure, as he needs to maintain a specialized hardware which is expensive. In this paper, we propose a new algorithm on basis of crypto-puzzle Integer Prime Factorization, for proof-of-work consensus, which makes a user with minimal hardware capabilities to participate in the mining procedure.

Keywords— Blockchain, Mining, Consensus, Proof-of-Work, Integer prime factorization, Computational feasibility.

I. INTRODUCTION

A Blockchain is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way [2]. The ledger itself can also be programmed to trigger transactions automatically. The Blockchain works in two models, traditional permission less model and permissioned model. In the traditional permission less model, any person, thing or entity can interact with other members or parties by creating an address on the network. Permissioned Blockchain act as closed ecosystems, where users are not freely able to join the network, see the recorded history, or issue transactions on their own. It is preferred by organizations, which leverage the power of the network for their own, internal business operations. Bitcoin is an example of the former model whereas Ripple Labs for the later.

The job of a Miner is to gather the transactions performed by various users and pack them into a block. In Blockchain, a Miner has to go through a mining procedure for adding a block to the blockchain. Typically, there will be multiple Miners working on adding blocks simultaneously to the blockchain. This mining procedure is governed by various consensus protocols. The consensus protocols decide which block will be added next to the blockchain. One of the widely adapted consensus protocols is "proof-of-work" (PoW). The PoW consensus protocol throws a challenge to a Miner, to solve which, the Miner has to spend a definite amount of processing time. The Miner who solves the challenge first, gets to add his block to the blockchain.

However, PoW consensus need not be applied in Permissioned Blockchains as they have known identity of network participants along with defined roles. Also the PoW mining demands high costs for computing hashes with a specified condition (C). Instead more emphasis can be put on transaction throughput, fault tolerance, overall efficiency and restriction of access to the Blockchain data and smart contracts [3].

The rest of the paper is organized as follows. Section II will describe the existing system. Proposed system is presented in Section III. Section IV shows the computational efficiency of the proposed system. Section V concludes the paper.

II. EXISTING SYSTEM

The Working Principle of Proof-of-Work

The PoW uses the puzzle friendliness property of Cryptographic Hash function. Given X and Y, find out N such that Y = HASH (X||N) where X is input, Y is Hash and N is a random number. It is difficult but not infeasible to find such random number N. With a known N, the Hash can be easily verified. The PoW consensus uses such a Cryptographic Hash puzzle proposed by Adam Back in HashCash [10] to throw challenges to the Miners.

Every block that is added to the Blockchain must have a Block Hash [4]. The Block Hash (BH) is computed from a Hash function with three parameters. First, the previous block hash (PH), second the Merkel Root (MR) and third, a random nonce N.

BH = Hash (PH, MR, N)

BH must follow condition C that there must be particular number of zeroes at the beginning of the hash. This condition is derived by Blockchain to achieve a certain degree of Difficulty. Difficulty is computed as follows [5].

Difficulty D = previous_difficulty * (2 weeks in milliseconds)/ (milliseconds to mine last 2016 blocks)

The Miner must find a nonce N to compute BH that satisfies condition C. The Miner creates the block, appends the BH to it and sends it to other miners for verification. The other Miners sign the block, if the transactions in the block are valid and the BH satisfies condition C.

Revised Manuscript Received on September 10, 2019.

Madhu Babu Janjanam, Computer Science and Engineering Vasireddy Venkatadri Institute of Technology Guntur, AP, India.
(E-mail: madhubabujanjanam@gmail.com)
Siva Prasad Pinnamaneni, Computer Science and Engineering Vasireddy Venkatadri Institute of Technology Guntur, AP, India.
(E-mail: sivaprasad.neni@gmail.com)
Prashant Atmakuri, Computer Science and Engineering Vasireddy Venkatadri Institute of Technology Guntur, AP, India.
(E-mail: prashant.atmakuri@gmail.com)

The expected number of hashes h [10] needs to be calculated to find a block with Difficulty D will be

$h = (D * 2^{256})/ (0xffff * 2^{208})$

Performance of Proof-of-Work with SHA-256

The Proof-of-Work mechanism in permission less Blockchain uses SHA-256 to compute BH. The time complexity of SHA-256 is O(n) [6]. For a Miner to compute BH, he/she has to check 2^{256} hashes out of which only 2^k hashes are valid, where k is the number of zeroes specified by condition C. For a user to become a Miner, he/she has to compute hashes at a rate of 17.2 Th/sec as of today to achieve break even. Such mining hardware like Innosilicon T2 Terminator [7] would cost approximately USD 1500. These requirements of high computation power, power hungry and expensive hardware are preventing a user to become a Miner. Even if a user affords the above hardware, it is a matter of time before the mining difficulty increases and the hardware becomes obsolete.

III. EXISTING SYSTEM & RESULTS

The Working Principle of Proposed System

Integer Prime Factorization is a process of splitting an integer into a couple of unique primes which, when multiplied together, form the original integer. The best published asymptotic running time is for the general number field sieve (GNFS) algorithm, which, for a b-bit number I, is

$$O(\exp \sqrt[3]{\frac{64}{9} b(\log b)^2}) \text{ [8]}$$

In the proposed system, instead of specifying a condition C to the Miner, an integer (I) is given, for which the Miner has to find prime factors p and q. After finding p and q, the Miner prepares a block that comprises of Integer I, prime factors p and q, previous PH, MR, a timestamp, BH and proposed list of transactions. The block is sent to other Miners for approval. The other Miners then follow the below steps to validate the block.

1. Validate p and q against I
2. Validate BH against I
3. Validate the proposed transactions

If all the three steps are successfully validated, the other Miners digitally sign the block. The Miner receives a copy of the digital signatures before the block gets added to the Blockchain.

Performance of Proof-of-Work with Integer Prime Factorization

As per the study of Madhu Babu J et.al [1] it is experimentally determined that the average latency for calculating prime factors in worst case will be 4.92 seconds when computed with the following hardware configuration. Intel Core2Duo Processor with 3.0 GHz, 2.9 GHz and 2 GB RAM.

IV. PERFORMANCE COMPARISON

In the existing proof-of-work mechanism, to compute h number of hashes with the hardware specified in Section II, it would take h/ $(17.2 * 10^9)$ seconds. With the proposed proof-of-work mechanism, the computation time is brought down to 4.92 seconds.

With such low figures, Integer Prime Factorization enables a user to mine Blockchain without the need of expensive hardware. At present, it takes 10 to 20 minutes to mine a block.

V. CONCLUSION

In this paper we proposed a new technique for proof-of-work consensus protocol. The proposed technique has less complexity compared to the existing SHA – 256 hashing algorithm. The proposed technique enables a user to participate in mining procedure even with minimalistic hardware configuration specified above in section III.

VI. REFERENCES

1. Saraiah Gujjunoori, Taqi Ali Syed, Madhu Babu J, Avinash D, Radhesh Mohandas, and Alwyn R.Pais, “Throttling DDoS Attacks,” Proceedings of SECRYPT 2009, Milan, Italy, 7-10 July 2009, pp. 121-126.
2. Iansiti, Marco; Lakhani, Karim R. (January 2017). "The Truth About Blockchain". Harvard Business Review. Harvard University. Archived from the original on 18 January 2017. Retrieved 17 January 2017.
3. Lukas Kolisko, 'Do we need mining in private and permissioned blockchains?', A Medium Corporation, 2018. [Online]. Available: https://medium.com/@lkolisko/do-we-need-mining-in-private-and-permissioned-blockchains-1a69b4c2c7a1.
4. Jayamine Alupotha, 'How to calculate the hash of a block in bitcoin?', A Medium Corporation, 2018. [online]. Available: https://medium.com/hackergirl/how-to-calculate-the-hash-of-a-block-in-bitcoin-8f6aebb0dc6d.
5. 'Difficulty', Wikipedia, en.bitcoin.it, [online], Available: https://en.bitcoin.it/wiki/Difficulty
6. D Rachmawati1, J T Tarigan1 and A B C Ginting, 'A comparative study of Message Digest 5(MD5) and SHA256 algorithm', 2nd International Conference on Computing and Applied Informatics 2017, Journal of Physics: Conf. Series 978 (2018) 012116.
7. Innosilicon T2 Terminator, [online], Available: https://www.asicminervalue.com/miners/innosilicon/t2-terminator.
8. Integer Factorization, Wikipedia, [online], Available: https://en.wikipedia.org/wiki/Integer_factorization
9. 'Difficulty in Mining', Wikipedia https://en.bitcoinwiki.org/wiki/Difficulty_in_Mining#What_network_hash_rate_results_in_a_given_difficulty.3F
10. Hashcash, Wikipedia, [online], Avaliable:https://en.wikipedia.org/wiki/Hashcash