A Proposal for a Quantum Safe Proof of Work Mechanism for Block Chains

Introduction

Bitcoin and all like digital currencies have two key vulnerabilities which render such technology unsuitable in its current form for the future of quantum computing. Most notable is the compromise of the ECDSA curve [1], however slightly less well known is the rendering of "proof of work" - the generation of a hash to "mine" tokens and process transactions - a trivial process. Proof of work is crucial to the processing of transactions, and its triviality would lead to hyperinflation as it would be a simple process to generate millions of tokens before the system would be able to compensate, if ever.

To begin to protect the future of this technology, this document proposes a new proof of work algorithm based on a quantum safe encryption, specifically the McEliece cryptosystem.[2] Such a solution not only future-proofs digital currency ecosystems, but enables entirely new industries as such computations would outdate ***all*** existing mining systems. Existing block chains, such as Bitcoin, could be adapted to this technology and protect existing investments. Those block chains that did not adapt this technology would seem inferior and a greater long term risk in comparison.

Proof of Work

It is the proof of work mechanism that determines several facets of how a block chain operates and should any single point become compromised, the continual operation of the network becomes threatened.

The Current Method

The first of the determinations created by proof of work is the regulation of timing between blocks. A block is a data set which contains a collection of transactions and a Merkle tree hash value that cryptographically links it to the parent block which preserves block order. A value known as the "difficulty" is representative of the scale of the amount of computations required for attaining an acceptable SHA-256D hash value that "solves" a block. The difficulty value is computed periodically as a function of the average of the solution times since the last difficulty calculation was made. Increasing difficulty is achieved by appending an increasing series of "0" characters to the beginning of a SHA-256D hash, making a suitable solution respectively less likely. The winning target hash must simply be of an ordinal value greater than a minimum value generated by the difficulty setting.

How Work Is Proved

As a new block may be submitted by anyone, the network rejects blocks that do not meet certain criteria which prove that the block was generated in a specific manner to which the entire network agrees (consensus). If these points are met, a block is considered valid and the transactions contained within as suitable for processing. After a certain number of additional blocks, the block is considered "uncontested" and "trusted".

The Quantum Safe Method

A quantum safe method of proof of work must satisfy all existing requirements in addition to invulnerability to the attacks of quantum processing. The solution is to create a network for which the proof of work is to "attack" the employed cryptosystem, itself. In this manner the system retains its designed operational parameters regardless of ongoing technological advancement.

> *NOTE: Throughout this explanation there are certain parameters which must be explored and proved to provide a suitable network operation.*

One of the facets of the McEliece cryptosystem is its known vulnerabilities at low entropy. [3]

Utilizing an application of a difficulty function similar to that which is currently employed, a quantum safe mathematical problem can be created utilizing the McEliece cryptosystem and the Sterns attack algorithm for which a relatively predictable period of time could be found for its solution.

How Work Is Proved

When a block is solved, a Merkle tree hash is generated from that result which becomes part of a solution hash for the next block. This solution hash represents the "problem" that miners attempt to "solve" to create the next block. The generation of this string will be based upon research into the amount of time Sterns algorithm requires to derive a valid private key from a string of values presented as a McEliece cryptosystem public key. ***The miner which "wins" a block will be the first to produce a valid signature that matches the solution hash as the public key, proving their execution of Sterns attack "solved" the "problem".*** As the solution hash verifiably has no corresponding private key when it is generated, a miner may not attempt to circumvent the proof of work process. The obvious attack vector is that a bad actor would solve a block and then "stuff" a value into the next solution hash for which they already know the computed private key, however to generate a public key which conforms to the difficulty constraints is an equally non-trivial task resulting in no net gain over the computational methods which have been outlined. Furthermore, as each solution is partially derived from a hash of the block immediately previous, no prior computation of values could be performed even if it were computationally expedient.

Conclusion

This document has attempted to provide a possible solution to the problem of a quantum safe proof of work mechanism which may provide a scalable resistance to increases in future computational power.

References

[1] https://bitcoinmagazine.com/articles/bitcoin-is-not-quantum-safe-and-how-we-can-fix-1375242150

[2] https://www.rocq.inria.fr/secret/C2/Transparents/Biswas.pdf

[3] https://eprint.iacr.org/2010/271.pdf

[4] http://cr.yp.to/codes/mceliece-20080722.pdf