```
BIP: amkn-posthyb
Title: Consensus (Hard Fork) PoST Datastore for Advanced Cryptography and Higher Efficiency Mining
Author: Andrew M. K. Nassief <loneroassociation@gmail.com>
Comments-Summary: General interests and concerns, community has been looking into more details
Comments-URI: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2021-March/018571.html
Status: Active
Type: Draft
Created: 2021-03-04
License: MIT | zlib | OPNL-2 | CC BY 4.0
Replaces: N/A
```

Table of Contents

Abstract

This BIP aims to provide a more complex cryptography alternative to Bitcoin's traditional encryption scheme. Secondly, it also specifies a hybrid algorithm for allowing for both traditional PoW and memory hard mining. The aim is for more optimal cryptographic efficiency and sustainable security in the future.

Copyright

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. End license text.

BIP Editor and Responsibilities

The current BIP editor/champion of this proposal **(the actual author)** will be responsible for the following:

1. Updating the BIP over time
2. Advocating for needed updates and changes
3. Communicating and commenting on community concerns
4. Leading the implementation and development of the proposed BIP

Specification

1. I do want to implement a NP hardness encryption scheme for complexity. This is not to be confused w/ cryptography for the polynomial reconstructive problem **[1]**
2. I am looking towards integrating memory hard compatibility w/ the mining algorithm. Memory hard computation allows for time and space complexity for data storage functionality, and there is a way this can likely be implemented without disenfranchising current miners or their hardware if done right.
3. In regards to mining, what I am doing isn't the same as the currently existing Hybrid Mining process out there, but it is similar.
4. The idea involves the implementation of specific memory-hard functions along with PoW to expand usage to the Hard Drive Mining space. Most likely, this simplest way to do this is a hybrid mining algorithm that computes similarly to PoW, but with some sort of specialized proof. I want to derive a PoW-like algorithm based off of proof of computation, or Verifiable computing.
5. The computation would likely integrate some sort of probabilistic check-proof scheme, and would need to be adaptable with traditional PoW as well as Proof of Storage. This means at the very least ASIC, CPU, and disk space compatibility.
6. I am looking towards more complex ways to tackle invalid blocks or sustainability problems. Most likely this may integrate a lot of custom code, but may be an implementation in regards to this specification. Also see **[2]**
7. The traditional PoW style mining and an asymmetric key encryption scheme will still be implemented in this new hard fork.
8. I found it interesting that there is already an open-source C++ implementation of Classic McEliece. Although, most likely we wouldn't be utilizing the Classic McEliece cryptographic proof in this hard fork implementation, this does show that a post-quantum implementation for *asymmetric key exchange schemes* is feasible.
9. More then likely, the cryptographic implementation created would be a multipath layer randomization cyphertext encryption scheme. It depends on algorithmic complexity, but this encryption scheme is based off of randomization and recomputation in similar ways to the McEliece proof. That scheme also will likley have forms of error encoding.
10. Looking into the bip-39 diceware sheme, and having seen discussions over high levels of entropy for diceware, I think the easiest cyphertext randomization method may be a high entropy NP-Complete and qauntum-resistant paraphrase generation method.

Hybrid Mining

This will likely be done through:

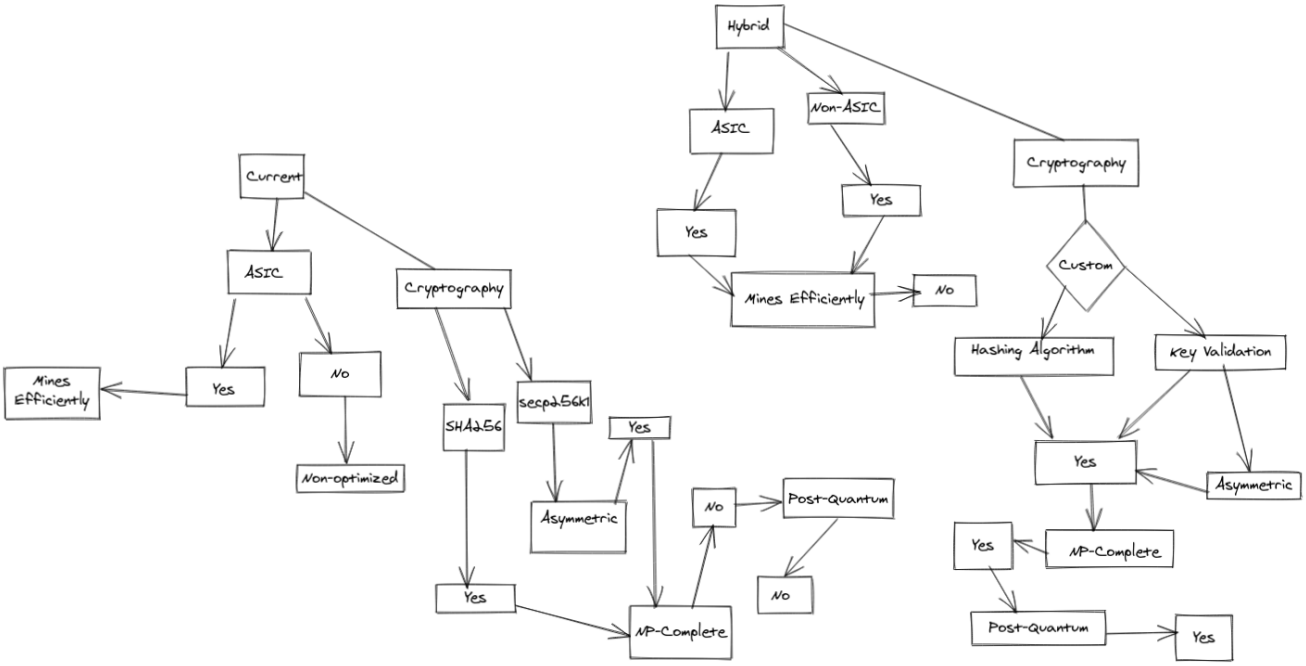
- A PoW/PoST Hybrid
- Not disenfranching current ASIC miners
- Allowing for specific MHFs (Memory-hard functions)
- Integrating a checkable proof centered around verifiable computing

Regarding Cryptography

My preferred method for Cryptography:

- NP-Complete
- Quantum-resistant
- High entropy ciphertext
- Randomization and recomputation

- Similar to diceware in paraphrasing



In regards to technological feasibility, this proposal does have a level of technological soundness in regards to implementation and how it is being developed. As noted, the biggest regard is in relation to cryptography and mining efficiency. Since you are making dramatic changes that may take a while to reach necessary consensus, likely this will need to be implemented through a hard fork. Many implications are considered in regards to the development of this proposal. For starters, a good reference point is in regards to the algorithmic implementations.

The new mining algorithm is a form of PoW in nature. That form is an example of proof of computation, and how to derive hashing power efficiently from both specialized hardware and the non-specialized. As noted, a core advantage is adaptability.

When you look at the combination of a more efficient mining algorithm and potentially higher levels of security, this hard fork can be easily upgradeable around certain technologies in the distant future such as mainstream Quantum advantage and supremacy. This is especially true considering where RSA and other hashing algorithms might stand in the future.

Access to more hardware makes it easier to fend off attacks, and an upgraded mining algorithm (which is seperate from the security upgrade), will provide the hashing power needed for optimized speed as well.

Motivation

The purpose of this proposal is primarily related to Bitcoin's mining algorithm, consensus and cryptography. The idea is centered around an improvement on mining efficiency and better security. The aim is for mining to be compatible the way that it currently is, as well as memory hard compatibility as well. Besides memory hard or *proof of space and time* compatibility, there are other changes that I'm looking towards implementation. This includes a cryptographic proof for NP completeness as how to best avoid a significant complexity cracking in the future or a cryptographic attack vector.

Nonetheless, Bitcoin's level of cryptography needs to significantly change in order for Bitcoin to be decentralized and relevant in the future. Significant worries such as Post-Classical Computing i.e Quantum Computing, the solving of the P vs. NP or other threats are posed for Bitcoin's sustainable future. Tackling many of these threats can better be instigated with the cryptographic proofs and systems I believe could be of value.

Mining Algorithm

In regards to mining, the new PoW hybrid mining algorithm should allow for mining accessibility across a variety of different hardware. This also won't disenfranchise current miners given that it would still have ASIC compatibility. Currently efficient hash rates are virtually incompatible outside of ASIC devices. This limits the network in terms of speed, and seems to incentivise mining farms instead of everybody else. If a sort of auxiliary PoW algorithm could exist for different hardware architectures, I don't see a reason why somebody with $2m of regular hardware can't mine the same amount of BTC as somebody with $2m worth of ASICs.

If mining pools could still exhibit similar levels or profitability with an upgraded consensus, it makes no sense why there needs to be a limit to the network growth. In the future, optimization algorithms might also be built into mining pools for block validation and energy efficiency across different infrastructure or for validation techniques. A PoW hybrid opens doors to multiple efficiency routes, as well as energy efficiency routes outside of renewables. This proposal might be a decent first step forward.

Regular vs. Hybrid Mining

Regular	Hybrid
ASIC	Multiple
Non-Optimized	Optimized
Slower Network	Faster Network
Lower Security	Higher Security

Further Information

Rationale

In regards to cryptographic key validation, an advancement beyond Satoshi's chosen secp256k1 algorithm needs to happen for sustainable security for the near future. This is in part due to the fact that the elliptic curve encryption chosen is very susceptible to break. I believe an asymmetric cryptography algorithm can be better optimized for polynomial time complexity such as an NP hardness encryption scheme possibly based off of Homomorphic encryption, or some sort of regressional model that can be more prominent in the future.

As stated prior, post-classical computing and potential unsolved mathematical theorems becoming solved, pose a huge threat to breaking Bitcoin's current encryption scheme on many levels. Outside of key validation, is the cryptographic complexity of the hashing algorithm. I believe SHA 256 can be replaced with something such as a NP-Complete Encryption scheme **[3]** and potential solutions similar to the McEliece cryptosystem **[4]**. This is something I aim to evidently tackle as well. However, I am looking into tackling mining first.

I also believe the PoW algorithm can be modified to allow mining the way it is currently done, as well as be more compatible for memory hard mining functionality. An implementation of a hybrid PoST *(Proof of Space and Time)* algorithm and traditional PoW *(Proof of Work)* is more suited for modern day cryptocurrency mining. While memory-hard hardware might be comparable to the costs of ASIC specific hardware in regards to mining efficiency, this new implementation is more related to expanding the infrastructure available for mining Bitcoin.

Outside of hardware availability, is also a need for an algorithm to be optimized towards less invalidated blocks. This will be feasible with complex data store functionalities.

Core Considerations

As noted, in regards to this hardfork and BIP proposal, there are many things to consider. These things include potential weaknesses in regards to Bitcoin and how it can be improved, energy consumption, the new mining algorithm, hashing and cryptography. By weakness, we are taking into account code and architecture as well as certain protocols, but not the overall system design. We are trying to find ways these things can be improved.

As the author of this BIP proposal, it is important to reiterate that the core is adaptability in regards to scalability and security. Secondary to this core, energy efficiency is a plus. As previously stated, "the aim is for more optimal cryptographic efficiency and sustainable security in the future". Henceforth, this is why this project is being developed.

Bitcoin's Weakness

Many cybersecurity experts believe that the way Bitcoin is currently developed, its encryption can easily be broken in a Post-Quantum world or through solving unsolved mathematical proofs that may be theoretically solveable. I also want to note that some researchers have proposed a Post-Quantum Blockchain or a Lattice-Based Proof-of-Work Consensus. However, these systems are different then what is being proposed in this BIP. In regards to cybersecurity, I have a preferred method in regards to tackling this. This has been iterated, as well as potential solutions to other problems Bitcoin may have.

One of the bigger problems, is energy consumption as well as the fact that Bitcoin may be subject to centralized mining. Given the sudden over-reliance on expensive specialized hardware, and rising problems in network speed and gas fees, I think simply solving the root would be the easiest method of tackling this.

It is also worth noting that some hardforks to an extent have successfully upgraded or changed the mining algorithm. An example of this to an extent may even be something like Litecoin. Infact, many people would say Litecoin is one of the better examples out there. Litecoin also has the problem of disenfranchising ASIC miners and similar future security risks.

Energy Consumption

There are many stats worth iterating when taking into account Bitcoin's energy consumption. Also, even if lots of BTC is mined with renewables in the future, an energy consumption problem that can be avoided should lead to a more optimal blockchain regardless. Bitcoin is an energy intensive network while having limitations to scale. It can handle an estimated 5 to 7 transactions per second, according to various sources such as Blockchain.com. While the full reliability of these sources could come into question, various sources reiterate the same thing.

As of March 2021 and now even April, the energy required for a single BTC transaction is 1065.95 kWh. That is the equivalent to the power consumption of the average US household in a period of 36.54 days, according to the Digiconomist. According to the same source, Ethereum's network also has growing energy consumption concerns. Single Ethereum transactions have the carbon footprint of 79,655 Visa stransactions, while single Bitcoin transactions by that metric have the carbon footprint of 1,122,196 Visa transactions.

According to various sources such as CryptoVantage and Princeton, China has an overwhelming majority stake in Bitcoin's hash rate. Although, the rate has steadily been declining over the years from nearly 80 to 60%. Still, this raises concerns on the potential ability to centralize mining.

When taking into consideration the sustainability problem and problems related to power consumption, renewables are too simple of an answer. The fact that it is still increasingly energy intensive, means increasing costs with renewables. It also means problems in regards to scalability and speed. The transaction times of the network are also quite slow considering the computing power it relies on.

Proof of Computation

For PoCP in regards to this hardfork, a hybrid algorithm is likely the way to go. An example of a network integrating a hybrid algorithm would be something like Lynx, however it disenfranchises ASIC miners on purpose. Instead of focusing on ease of mining, it is better to focus on ease of compatibility. This is why it would make more sense to focus on the hyrbid consensus that bridges the gab for PoST compatibility along w/ traditional PoW. Although PoST is a form of PoW, I meant traditional PoW in regards to tailoring towards traditionally targeted or utilized hardware.

A simple hybrid implementation could be between the Cuck(at)oo Cycle PoW and PoST or one can do a hybrid implementation w/ traditional HashCash & SHA256 and PoST. In regards to hybrids in the past, most of them seem to be hybrids of PoW and PoS (Proof of Stake) with exceptions being things like Deterministic Proof of Work (DPoW) and DSBFT.

In regards to our hybrid consensus proof, lots of the core variables would probably look alot like this:

```
Function PoCP
   Inputs:
      transaction (t):  // unconfirmed transaction
```

```
    epoch_time(e):  // current epoch time
    algoType(a):  // mining algorithm
    block_n (n):  // block number
    key (k):  // validation key
    hashType (h):  // hash type
Output:
    block_nx:  // next block
```

We would introduce `algoType` as a method for reading the algorithms used in the hybrid consensus. As mentioned earlier, we are looking for a PoW and PoST hybrid consensus, which may be applicable by recognizing both algorithms within the platform in regards to validation methods.

This shouldn't disenfranchise current ASIC miners, and can be utilized in a way that accepts specific memory-hard functions as well. We are basically creating an entire method where the validation process is a form of verifiable computing. This is in regards to the algorithms working on the hashing, confirming transactions and validating blocks.

Cryptographic Generation

There are many things to consider in regards to the cryptography, especially considering Bitcoin's chosen key verification method and its reliance on SHA-256. To reinstate a prior point, the preferred method in this BIP is NP-Complete and quantum-resistant. Since the simplest method should be reliant on key verification, perhaps some form of high entrophy ciphertext can do the trick. It may be similar to diceware in paraphasing, with a heavy reliance on randomization and recomputation.

For sake of simplicity, the best method in regards to paraphrasing may be something similar to the McEliece cryptosystem or a variation of hashing centered around quantum random numbers. A quantum random number hashing iteration would be high entropy and less vulnerable, as opposed to algorithms centered towards classical computing that have a more deterministic element to it. QRNG or quantum random number generators, can be utilized in the crypo space for potential post-quantum encryption candidates.

Quantum Random Numbers

Since many QRN generators are hardware-based or require an API, it is likely better to focus on the key mechanisms and paraphrasing elements of quantum-safe cryptography, as it is optimal in the very beginning to run a variation of quantum-safe cryptography through a quantum-safe key encapsulation mechanism and hash functions directly in the code. Liboqs is a decent place to reference.

The three methods preferred by the author of this BIP are potentially deciding on Classic McEliece, the Quantum Diceware method (which is quite similar), or a variation of hashing likely centered around Quantum recursiveness. What all these have in common are key and paraphasing mechanisms. The cryptographic implementation is to be determined based off of benchmarking which results will be optimal for speed. There also needs to be ease of simplicity.

In regards to the style or nature of the implementation, this has already been determined as likely a quantum-safe key encapsulation mechanism.

Cipher Mechanisms

Since one will do a key and paraphrasing mechanism such as quantum-safe key encapsulation, the focus would likely be primarily on randomization. The cipher mechanism would require the following:

- Quantum paraphrasing for the keys
- Generated seed based on cipher mechanism
- Randomization for the parphasing generation

It is worth noting that Quantum-resistant keys or cryptographic variations when integrated, shouldn't have that much of an effect in regards to network speed. It is more in relation to a security upgrade as opposed to mining which is tied to speed and efficiency. However, it would be optimal for it to be as robust as possible so that speed can be a key advantage along with security.

Overall Hardfork and Blockchain

Regarding the specifications, hopefully this has been as explanatory as possible in regards to the expected parameters or technological implementation this harkfork would need. There has been general discussion in regards to whether or not this BIP should be broken down into multiple BIPs. However, I think since this is in relation to the same hardfork and the target is still optimization, speed and security, a single BIP makes more sense.

There are also other things I may do in regards to this hardfork such as change the total supply from 21M to 21B, but technological aspects in regards to supply, aesthetics, or things outside of protocol or consensus upgrades aren't meaningful to include in this BIP. The core is to stay focused.

In regards to discussions related to whether or not this should be a soft or hardfork, since there are dramatic changes taking place in regards to the mining algorithm, hashing and signature, it makes more sense to introduce this as a hardfork. Afterall, the hope in regards to successful hardforks is community adaption.

Backwards compatibility

This should be backwards compatible with stratum and BIP 40 and 41. Things such as blind merged mining in BIP 301, will likely not work. BIP 340 is implemented specifially for secp256k1. In regards to BIP 151 compatibility, BIP 151 was already withdrawn.

This is a hard fork given the obvious changes in the consensus. Keep in mind that also given the new cryptography method, most likely a new address type would need to be created **[5]**.

Also given the changes to the mining algorithm, it would likely be more optimal for current miners to mine on this new network to reach majority. The same rule of thumb generally applies to most hard forks.

Reference implementation

I want to emphasize that I believe greater mining compatibility means a higher level of decentralization. Also, in regards to encryption keys and validation, I believe a higher level of security on that front also means a higher level of decentralization. On the issue of mining, I believe that a hybrid mining protocol done right wouldn't disenfranchise current miners (as stated prior), and would also still allow people with other hardware to mine BTC.

I believe a key validation system if done right, could eventually reach NP-completeness. Many promising algorithms have had forms of NP-Completeness in regards to encryption schemes, and coincidentally some of them were also known to be forms of Post-Quantum encryption **[6]**. Building an NP-Complete solution can be centered around, yes, it can be solved utilizing deterministic Turing machines, but make the luck chance near impossible forming some form of information-theoretic security in theory.

Once my proposal starts getting some general concensus, I aim to publish some preprints further detailing the encryption scheme and math. The same applies to starting the coding and software development, as I have the workflow planned out.

Definitions

1. **Consensus:** The agreement protocol among nodes/peers in the network
2. **Post-Classical Computing:** Primarily refers to computing w/ a non-classical form of computation. Primarily it often refers to Quantum Computing, but very occasionally is used in reference to forms of DNA computing or other less common areas.
3. **secp256k1:** The elliptic curve parameters for Bitcoin's public-key cryptography. **[7]**
4. **Hashcash:** A PoW algorithm made by Adam Beck that is widely used for Bitcoin's mining functionality. **[8]**
5. **XII:** Former SHA-3 candidate that was previously ASIC-resistant **[9]**
6: **Proof of Capacity:** In layman terms, Proof of Work done through data storage
7: **Memory-hard:** Functions that require significant amounts of memory to evaluate. **[10]** However, this can go beyond CPU-cycles.
8: **NP-Complete:** NP Complete is different then NP-hardness and hardness of an NP-Complete problem. NP Complete problems are the hardest in the NP complexity class. In regards to hardness, one can optimize building a solution that is near impossible to solve over deterministic brute force or some sort of information-theoretic security scheme such as MPC. **[11**]
9: **Asymmetric Keys:** An encryption validation scheme that utilizes private/public keys.
10: **ASIC Mining:** ASIC stands for application-specific integrated circuit, and is often synonymous with specialized hardware.
11: **Homomorphic encryption:** An encryption scheme that lets one perform calculations on the data prior to decryption. **[12]**

See Also

[1] Evaluation of Polynomial Reconstruction Problem using Lagrange Interpolation Method
[2] Some Miners Generating Invalid Blocks
[3] Why hasn't there been an encryption algorithm that is based on the known NP-Hard problems?
[4] McEliece cryptosystem
[5] Quantum computing and Bitcoin
[6] NIST Post-Quantum Cryptography Competition
[7] Secp256k1
[8] Hashcash
[9] X11
[10] Memory-hard function
[11] MPC
[12] Homomorphic encryption

Acknowledgements

Partial inspiration comes from the Chia Network, however this isn't the same thing. The mining capabilities in this proposal isn't meant to be a replacement to PoW, rather we are giving PoW data storage memory-hard compatibility w/ the currently existing mining compatibilities it has now. In regards to overall cryptography, we were somewhat inspired by the halting problem and Quantum recursion theory. There is a huge belief behind this proposal that Bitcoin will radically need to change its cryptography to be relevant in the not so distant future. Regardless of supposed controversy, this is a huge step evidentially needing to be done and even allows for Bitcoin to be further decentralized than what it currently is. I am looking to help create some of these custom cryptographic proofs that may help with that mission.

Further Reading: Better Cryptography Means a Better Bitcoin