BiblePay – White Paper

A Crypto-Currency using the Proof-Of-Bible Hash (PoBh) algorithm as a Proof-Of-Work function

Robert Andrew (rob@biblepay.org)

September 4th, 2017

Abstract

A decentralized crypto-currency using a new proof of work algorithm based on the King James Bible, called Proof-Of-Bible-Hash. One of the goals employed is the prevention of hashing in ASIC devices or GPUs.

Proof of Bible Hash uses 31,101 verses from the text of the bible, making it impractical to port to an ASIC or GPU, due to multiple considerations including but not limited to: Chained verse logic, memory consumption, encryption, decryption, transaction ID lookups, blockindex lookups, Receiving address Lookups, Full Node business Logic, floating time parameters, and late block logic. In addition, the hashing algorithm requires AES512 encryption and decryption to restrict GPU execution. The algorithm also uses a complicated ruleset to chain the verses together (more on the technical details below).

Additionally, the algorithm appends information from the blockchain as the miner is mining, with the goal of requiring a full node. In this way, the algorithm creates more overhead to run outside of a full node than inside a full node. The effect of having a network of full nodes available adds electrical efficiency and stability to the bible pay network.

Target Audience

Cryptologists, Scientists, Engineers, Financial Analysts, Accountants, Investors, Programmers, and crypto-coin enthusiasts.

Introduction

After Bitcoin was created by Satoshi Nakamoto in 2008, as it evolved and matured, waste heat increased exponentially to secure bitcoin blockchains and bitcoin clones. The mining operation generates heat by testing billions of combinations of SHA256 hashes. To replace this function with a proof-of-work that generates less heat and is less prone to degrading the network down to hashing devices (promoting a service based ecosystem in contrast to a device competition based ecosystem), Biblepay uses the PoBh algorithm that discourages execution outside of the full node environment.

Sanctuaries

BiblePay shares DashPays MasterNode technology to provide the ability to use Private Send, Instant Send, and Voting. This was driven by the desire to decentralize all of the charitable functions of Bible Pay.

For example, in a classic decentralized currency, in order for the community to vote on an item, a forum poll would be held and the limited results used for the effort (to manually code) the results in the wallet to execute the decision.

With Sanctuaries, we create a decentralized in-wallet poll, allowing masternodes to cast votes. The vote results will drive the disbursement of funds to the appropriate expense accounts and respective allocated amounts. The wallet will automatically release the funds from escrow, using superblock payments to our charity addresses while adhering to the established coin parameters. In this way, the coin may function over time in a completely decentralized manner, with governance at the Sanctuary level. The lead dev will not have any say other than his/her sanctuary votes once that phase is in production.

Technical Details for Proof Of Bible Hash

The wallet uses the X11 blockhash to maintain the reference pointers for the blockindex map. However, it uses the BibleHash to regulate difficulty and prove that a full node generated the biblehash (by requiring a txindex lookup, a block hash, or a receiving address to be present in the suffix of the hash).

Detailed Technical Information for the Bible Hash:

1. The BibleHash function is fed an input X11 hash of the current block template at a point in time. This starts as a uint256. It is also fed the reference to the last block index (and previous height and previous block time).
2. The BibleHash function encrypts the x11 hash uint256 using AES512 into a ciphertext vector. This ciphertext vector is then converted to base64. (These functions were chosen to raise the bar to reduce the likelihood of porting the hasher to a GPU as AES512 requires the OpenSSL library).
3. The resulting base64 is then md5 hashed.
4. The md5 hash is 32 bytes long. The BibleHash function breaks the md5 hash into 8 octets of 4 bytes each.

For each 4 byte octet, the Hex is multiplied * the IVerseFactor (.4745708). This IVerse factor points to the corresponding KJV bible verse between 0-31101. This resulting verse is chained to the output and this process repeats for octets #2-8, while appending the verses to the chained verse output.

5. When the BibleHash function reaches verse #8, it breaks up the source four byte octet into four elements: a Hex 2 byte source resulting in a lookback block offset from 0-255, a hex one byte source resulting in a transaction offset of 0-15, a hex one byte source resulting in a transaction output offset of 0-15, and a one byte source resulting in a datatype pointer of 1-3 (by multiplying 0-15*.1875) (used to determine if this bible verse will need to reference a blockhash, transaction ID or a receiving address). Then the BibleHash calls out to the full node for the resulting DataType from the chain by reading the disk, retrieving the result, and appending the result to the final chained bible verse (verse #8).
6. Then the resulting chained verses text contents are MD5 hashed to provide a concise input to the X11 hasher.
7. The MD5 hash is X11 hashed.

8. The X11 hash is sent through a business logic filter requiring the full node business logic of the latest Mandatory version of biblepay (IE some business logic from the wallet adjusts the resulting hash depending on block number).
9. Next, if the block is older than the late block threshold, the X11 hash is modified to be easier to solve.
10. If the block is a TITHE_MODULUS block, the block is easier to solve.
11. The resulting X11 hash is sent out of the function as the hash result.

Charitable Functions of BiblePay

One block per 10 is designated as a "TITHE BLOCK". This allows 10% of the blockchain emissions to be spent on charitable contributions to charities of over 75% efficiency. In our startup phase, we partner with compassion.com exclusively.

This is a temporary situation, set until Sanctuaries go live. Once the sanctuaries are live, the governance committee may vote on how the charitable wallet funds are spent, designated, and facilitating the charity funds to be transferred from the allocated escrow address to the charity. In this way, the wallet will become a decentralized autonomous charity.

Reward Breakdown

The block subsidy is 20,000 BBP per block with 7 minute block targets.

The subsidy reward decreases by 10% per year. The reward is also decreased per block if network difficulty spikes.

In this way, since no interest is paid, and fewer emissions are generated per year, the coin is a deflationary asset.

References

Nakamoto S. (2008): Bitcoin: A peer-to-peer electronic cash system. (http://www.bitcoin.org/bitcoin.pdf)

Waste heat and power dissipation: Inefficient Heat Generation: (http://en.wikipedia.org/wiki/CPU_power_dissipation)