

Bitcoin Hush (**BTCH**) Cryptocoin Specification

Pre-Release Version

Duke Leto, jl777

January 27, 2018

Abstract.

Bitcoin Hush (**BTCH**) is a new research and development cryptocoin which has many unique features compared to existing options. It avoids all transaction history and simply imports Unspent Transaction Output (UTXO) values for four different blockchains onto a fifth, brand-new chain. We use the Komodo Asset Chain feature to build a coin with delayed-Proof-of-Work (dPOW), which enjoys the full security of Bitcoin level security via notarization.

Additionally, the need for the latest two way replay protection (2WRP) algorithms are completely avoided, since no transaction hashes are leaked onto the new chain. This also completely avoids the problem that many Bitcoin forks have where they inherit a very large existing chain and must sync gigabytes of data.

We hope these techniques are utilized in all future Bitcoin and related forks to avoid large inefficiencies as well as potential replay attacks.

The recently released **HushList** protocol is compatible with **BTCH**, **KMD** and all **KMD** asset chains, which all contain *zk-SNARK* technology. Additionally, **HushList** is known to be compatible with **HUSH**, **ZEC**, **ZCL**, **ZEN**, **ZER** and the upcoming Zgold **ZAU** by radix42.

This specification defines how the **BTCH** cryptocoin works and how it builds on the foundation of **Komodo**, **Zcash** and **Bitcoin**.

Keywords: privacy coin, cryptocurrency, UTXOs, anonymity, freedom of speech, cryptographic protocols, electronic commerce and payment, financial privacy, proof of work, zero knowledge, zkSNARKs.

Contents	1
1 Introduction	3
2 Hush UTXOs	3
3 Bitcoin UTXOs	3
4 SUPERNET UTXOs	3
5 DEX UTXOs	3
6 Transporting Money to the BTCH Chain	4
7 Special Thanks	4

Introduction

Bitcoin Hush is a "mergedrop", i.e. it is an airdrop of value from four different chains, merged together, on a new chain.

Hush UTXOs

NOTE: This pre-release refers to stats before the actual snapshot, these numbers will change slightly with the official snapshot.

This data corresponds to Hush Block Height 245496 at XXX UTC and was extracted directly from the internal LevelDB database using both [bitcointools] and [utxodump].

Since Hush is a fork of the Zcash codebase, and Zcash forked from Bitcoin 0.11.2, Hush and all Zcash code forks, to our knowledge, use the older v0.8-v0.14 LevelDB format.

At this height, 607134 UTXOs exist, in 95933 transactions and 3177032.96851848 HUSH. This data is extracted from the response of the *gettxoutsetinfo* RPC command while the full node is paused at the correct block. Pausing the Hush daemon to allow RPC calls is not a standard function, the **pause** branch of MyHush/hush.git includes a way to achieve this. This functionality will most likely be made a proper command-line argument to make it easier to analyze UTXOs in the future.

Bitcoin UTXOs

NOTE: This pre-release refers to stats before the actual snapshot, these numbers will change slightly with the official snapshot.

This data corresponds to Block Height 505157 at 1-20-2018 10:22:15 UTC and was extracted directly from the internal v0.15 LevelDB database with the [utxodump] tool. It took approximately 22 minutes to dump all 65 million UTXOs at this height.

There is 16814298.58608387BTC in circulation stored across 63646112 UTXOs in 27834789 unique transparent addresses (taddrs). Somewhat surprisingly, there are 5605 UTXOs with exactly 0BTC in them and another 872314 UTXOs of a single satoshi. 7002416 transparent addresses are above the dust level of 0.01BTC, and these addresses and values will be transported to the BTCH chain.

Note that UTXOs of any size are taken into account, but the final sum of all UTXOs in one taddr must be at least 0.01BTC to be part of the airdrop. This level was also used by the Bitcore project (BTX) and it seems like a good standard to follow.

SUPERNET UTXOs

...

DEX UTXOs

...

Transporting Money to the BTCH Chain

The **z.sendmany** RPC is used to efficiently send money to all the appropriate addresses, with the appropriate amount, and the new BTCH chain.

Once the final snapshot balances are known, the taddrs from other networks are transformed into a Komodo-compatible address. This is done by taking the RMD160 and then changing the prefix to the KMD pubtype or p2sh type (given a 1 or 3) and then base58_check encoding.

Then, many many **z.sendmany** transactions are performed, each with many recipients (such as 100 or 128). Once the **z.sendmany** transactions for a particular chain are sent, that completes the airdrop process. Now users can claim their airdrop via their private key.

Special Thanks

Thanks to madbuda for providing servers and massive amounts of bandwidth for this fantastical project.

References