

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** [HushList] protocol is compatible with **BTCH**, **KMD** and all **KMD** asset chains, which all contain *zk-SNARK* [BCTV2015] 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 What is a UTXO, really?	3
3 Hush UTXOs	3
4 Bitcoin UTXOs	4
5 SUPERNET UTXOs	4
6 DEX UTXOs	4
7 Transporting Money to the BTCH Chain	4

8	How To Access Your BTCH Airdrop	4
9	BTCH Chain Size	5
10	Future Directions	5
11	Special Thanks	5
12	References	5

Introduction

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

It's become common to fork the Bitcoin or Zcash network while inheriting all transaction history, which lead to a cryptocurrency that has very little hashpower to protect massive amounts of data. We refer to this as a "worst of all worlds" solution and the ideas in this paper provide working examples of avoiding this situation.

For example, Bitcoin Gold inherited about 120GB of Bitcoin history and every Bitcoin Gold full node must download that locally, before ever getting to any Bitcoin Gold history. The later in time a Bitcoin fork occurs and uses the method, the larger the dataset is (currently 150GB and growing faster) and the more worse the "worst of all worlds" becomes. Just about every Bitcoin fork seen today uses this method.

We hope that various organizations and companies realize that the Komodo Platform provides "easy onramps" to making a new blockchain which improves on older "techniques" which amount to forcing full node operators to use excessive amount of bandwidth and disk space for no good reason.

Additionally, the new **HushList** protocol is compatible with Bitcoin Hush and provides the world yet another place to have secure and private communication.

What is a UTXO, really?

Addresses spending money are called "inputs" and those receiving money in a transaction are called transaction outputs, TXOs. Transaction outputs that are not spent are called Unspent Transaction Output, or UTXOs. The concept of "spentness" is a function of block height, i.e. the fact of whether a particular TXO is spent or not changes over time. This is why a snapshot block is needed, to define a point in time to find unspent outputs.

A UTXO is just a blob of binary data, and has many different formats that have evolved since Bitcoin began in 2009. The data usually contains some kind of public key or a hash of one, and other metadata. Most UTXO are around 20 bytes but multisig UTXOs can be hundreds of bytes.

For example, one of the most common forms of UTXO looks like this

76a914d3730f005e16bbf741ccbf60a5f66b7be930cb7a88ac

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 Jan 24, 2018 7:03:34 AM 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 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 Jan 24th, 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 **63.6** million UTXOs at this height.

There is **16814298.58608387 BTC** in circulation stored across **63646112** UTXOs in **27834789** unique transparent addresses (taddrs). Somewhat surprisingly, there are **5605** UTXOs with exactly **0 BTC** in them and another **872314** UTXOs of a single satoshi. The latter are mostly likely from transaction spam attacks from other networks.

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).

SUPERNET UTXOs

To Be Determined

DEX UTXOs

To Be Determined

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 transparent address 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 type, then base58_check encoding to produce a transparent address for the BTCH network.

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.

How To Access Your BTCH Airdrop

The very high level idea is that your private key to a transparent address gives you access to new funds on new network. Extreme paranoia about private keys is not unreasonable, and a very easy and powerful defense you can take to any bug or malware or anything stealing your original **HUSH** is this: After the snapshot is defined, move your **HUSH** to a new address, which can be a taddr or zaddr.

If you move your **HUSH** post-snapshot, the privkey in question is no longer valid for your **HUSH**, and you can worry a bit less. This also protects people that might have DNS or BGP attacks redirect them to illegitimate downloads.

For simplicity, let us assume t_A is a taddr with private key k_A . The **dumpprivkey** RPC method exists in all Bitcoin forks as well as Zcash forks and dumps the private key in "Wallet Import Format" (WIF). This format is accepted by the Agama Wallet. Once imported to the Agama Wallet, **BTCH** can be used on BarterDEX [BarterDEX] as well.

BTCH Chain Size

One of the huge features of using a KMD asset chain with an airdrop is avoiding all transaction history, i.e. all transaction data is ignored, only final balances in a transparent address are transported to the new chain. When airdropping a Bitcoin fork, this currently will save close to 150GB in chain size!

The exact numbers for the new BTCH chain will not be known until the snapshot is complete, but current estimates are that the initial BTCH chain will be under 0.5GB in size!

Future Directions

The [HushNG] GUI is actively being worked on and will integrate Bitcoin Hush as the first additional coin and will provide decentralized censorship-resistant communication, powered by *zk-SNARKs*.

Special Thanks

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

References

- [BarterDEX] j|777. *barterDEX - Atomic Swap Decentralized Exchange of Native Coins*. URL: <https://github.com/SuperNETorg/komodo/wiki/barterDEX-Whitepaper-v2> (visited on 2017-12-28) (↑ p5).
- [BCTV2015] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. *Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture*. Cryptology ePrint Archive: Report 2013/879. Last revised May 19, 2015. URL: <https://eprint.iacr.org/2013/879> (visited on 2016-08-21) (↑ p1).
- [Bitcoin] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. October 31, 2008. URL: <https://bitcoin.org/en/bitcoin-paper> (visited on 2016-08-14) (↑ p1).
- [dPoW] j|777. *Delayed Proof of Work (dPoW)*. URL: <https://supernet.org/en/technology/whitepapers/delayed-proof-of-work-dpow> (visited on 2017-12-27) (↑ p1).
- [HushList] Duke Leto David Mercer. *HushList Protocol Specification*. URL: <https://github.com/leto/hushlist> (visited on 2017-01-25) (↑ p1).
- [HushNG] Matthew Reichardt. *Hush New Generation GUI*. URL: <https://github.com/MyHush/hush-ng/> (visited on 2017-01-25) (↑ p5).
- [Komodo] superNET. *Komodo Platform*. URL: <https://komodoplatfrom.com> (visited on 2017-12-28) (↑ p1).
- [utxodump] Duke Leto. *utxo_dump Github repo*. URL: https://github.com/leto/utxo_dump (visited on 2017-01-25) (↑ p3, 4).
- [Zcash] Daira Hopwood. *Zcash Protocol Specification*. URL: <https://github.com/zcash/zips/blob/master/protocol/protocol.pdf> (visited on 2017-12-28) (↑ p1).