Bitcoin Hush (**BTCH**) Cryptocoin Specification
Pre-Release Version

Duke Leto, jl777

February 10, 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 full Bitcoin hashpower 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 unused data.

We hope these techniques are utilized in 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**, Komodo **KMD** and all **KMD** asset chains, which all contain *zk-SNARK* [BCTV2015] technology. Additionaly, **HushList** is known to be compatible with **HUSH**, Zcash **ZEC**, VoteCoin **VOT**, Zen **ZEN**, Zerocoin **ZER**, Zclassic **ZCL** and the upcoming Zgold **ZAU** by radix42.

This specification defines how the **BTCH** cryptocoin works and how 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

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 cryptocoin 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

The first 4 bytes **76a9** and the last 4 **88ac** correspond to Bitcoin Script opcodes and the rest of the data in the middle is the hash of a public key, which is why this kind of UTXO is called a Push-To-PubKey-Hash (P2PKH) UTXO.

Hush UTXOs

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.

This data corresponds to Hush Block Height **249985** at Feb 1st 2018 11:58:34 AM UTC and was extracted directly from the internal LevelDB database using both [**BitcoinTools**] and [utxodump].

At this height, **653804** UTXOs exist, in **103408** transactions and **3221897.42163446 HUSH** was in circulation. At this point in time, there were 17617 unique transparent addresses with at least 1 UTXO.

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.

The Hush airdrop included all addresses, no matter how small the amount in the address.

Hush UTXO Statistics

One would expect zero, but actually 4 addresses used Push-To-PubKey P2PK UTXOs, a format that has not been actively used since before Hush came into existence. By explorer analsysis and the fact that these addresses almost exclusively receive 12.5 HUSH block rewards, we know they must be solo miners who modified existing P2PK software for the Bitcoin network to work for HUSH. That is a great example of how Zcash "embraced and extended" Bitcoin itself, and why it is very reasonable to port features, and entire applications between coins.

Bitcoin UTXOs

This data corresponds to Block Height **507089** at Feb 1st 2018 11:55:44 UTC and was extracted directly from the internal v0.15 LevelDB database with the [utxodump] tool. The initial process of dumping took approximately 30 minutes, and then there are various other stages of filtering, parsing and verifying the data.

There was **16838448.58598230 BTC** in circulation stored across **61559974** UTXOs in **28983417** transactions and **xxx** unique transparent addresses (taddrs).

Bitcoin Spam Attacks

Somewhat surprisingly, there are **5568** UTXOs with exactly **0 BTC** in them, in 816 unique taddrs. These are overwhelmingly P2PKH UTXOs (99.77%) and no multisig or native segwit zero satoshi UTXOs were detected.

Much more common are 1 satoshi UTXO, with 871177 UTXOs in 608188 unique taddrs. All UTXO types except segwit native are present in this set.

These are "unprofitable" UTXOs in the sense that someone would lose more money by claiming their value, due to the fact that current network fees are greater than those UTXO values. Most likely they are remnants of transaction spam attacks from miners, who can often earn their money back via fees from the block that the UTXOs occured in.

Bitcoin UTXO Statistics

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, on the new BTCH chain.

Once the final snapshot balances are known, the transparent addresses from other networks are transformed into Komodo-compatible addresses. This is done by taking the RMD160 and then changing the prefix to the KMD type, then base58_check encoding to produce new transparent addresses 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 the private key to your transparent address gives you access to new funds on a new network. Extreme paranoia about private keys is not unreasonable. Fear of software bugs, malware, or anything stealing your original **HUSH** can simply be countered by moving your **HUSH** to a new address, which can be a taddr or zaddr, after the snapshot.

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 illegitamate 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 airdroping 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!

Running a BTCH Full Node

Running a full **BTCH** node requires downloading and syncing the **KMD** chain, which **BTCH** uses for dPoW. This will use gigabytes of bandwidth at first, and could take a day or more to fully sync depending on download speeds.

Running a full **BTCH** node is not required to protect the security of the network, nor to generate new money, but full nodes can earn the verification reward of 0.0001 BTCH per block plus any transaction fees in that block.

Full instructions can be found on the main **BTCH** Github Repo [BitcoinHush].

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 Tamsin Thorn for extensive help editing this document and to all the people that help maintain all the Free and Open Source software that our community relies on.

References

[BarterDEX] jl777. *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).

[BitcoinHush] Duke Leto jl777. *Bitcoin Hush Github repo*. URL: `https://github.com/bitcoinhush/bitcoinhush/` (visited on 2017-01-25) (↑ p5).

[dPoW] jl777. *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] Hush Developers. *Hush New Generation GUI*. URL: `https://github.com/MyHush/hush-ng/` (visited on 2017-01-25) (↑ p5).

[Komodo] superNET. *Komodo Platform*. URL: `https://komodoplatform.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).