Wormhole: A Smart Contract Solution for Bitcoin Cash

Abstract

Born at block height 478,558, Bitcoin Cash (BCH) has been dedicated to bringing a reliable electronic cash to the world and fulfilling Satoshi's original vision as a "peer-to-peer electronic cash". It enjoys global seamless circulation, permissionless innovation among other features. The problem of issuing tokens on BCH has bothered BCH community for long. Heavy research has been conducted regarding to the problem and solutions like colored-coins and OP_GROUP opcode have been proposed. The OP_GROUP solution was proposed by Andrew Stone to enable representative tokens on BCH with a new added opcode OP_GROUP. To enable function similar to that of ERC20 protocol which enjoys great popularity on Ethereum network with the OP_GROUP solution, the consensus rule of BCH has to be altered.

Any token issuance proposal that requires certain consensus upgrade will inevitably cause problems, including technical risks, harsh conflicts and huge controversy among community developers. Such controversial proposals often end in failure. The debate could be beneficial to the community and can be regarded as an insurance mechanism to prevent "radical" initiatives being depolyed and to ensure stability and security of the network. Yet it might as well hinder innovation at the protocol level. The prolonged debate around block size which lead to the birth of BCH is an even more unavoidable evidence of social psychology.

Rapid innovation demands a PERMISSIONLESS community. We have been exploring ways to realize smart contract on BCH without modifying BCH's consensus rule. After extensive research, the Omni layer protocol, a token issuance solution via OP_RETURN opcode, drew our attention. The Omni layer protocol runs on Bitcoin network and is also the foundation for the daily distribution and circulation of USDT. Fortunately, the Omni layer project adopted the MIT open source license which allowed us to fork the project, transplant to BCH network with minor modification and meet the needs of token issurance over BCH. To distinguish the protocol from the original Omni protocol over Bitcoin, we refer to the protocol as Wormhole protocol and refer to the corresponding native token as Wormhole Cash (WHC).

Terms

OP_RETURN: an opcode in BCH. Transaction output with OP_RETURN is unspendable and can be safely removed from UTXO collection. After the protocol upgrade in May 2018, BCH increased its default data-carrier-size of OP_RETURN to 220 bytes.

Wormhole Protocol: the protocol to realize smart contract on BCH, based on Omni layer protocol.

Wormhole Cash: native token of Wormhole protocol, or "WHC" in short.

Principle

Wormhole protocol is implemented upon BCH blockchain. It adds new features like token generation, circulation and burning etc. to BCH network without any modification to the consensus rule of BCH.

OP_RETURN carries the metadata of a token transaction. Token issuance, circulation and burning within Wormhole protocol can only be completed via BCH transaction, e.g. by interpretating the metadata in OP_RETURN.

Wormhole protocol reuses BCH's transaction system. Wormhole nodes need to identify transactions and addresses of BCH network and parse possible Wormhole transaction carried by OP_RETURN. Wormhole protocol can be regarded as a superset of BCH's consensus rule which does not care and has no need to parse the data carried by OP_RETURN.

Implementation

Wormhole protocol is implemented by extending Bitcoind and there is no need to change the consensus rule of BCH. Bitcoind client with integration of Wormhole protocol is called Wormhole client. Nodes running Wormhole client are able to recognize the Wormhole protocol data in OP_RETURN.

Two-Layer Security

The first layer is the security of BCH transaction. BCH adopts POW mechanism as a decentralized timestamp server. The mechanism has been running stably for nearly 10 years. Here are some of the benefits of the UTXO model:

- UTXO requires no balance maintenance
- UTXO serves as an independent data logger that can speed up transaction verification.
- UTXO model only cares locking script and unlocking script.
- UTXO has high performance when processing transactions.

Wormhole protocol reuses the entire UTXO security model of BCH and its decentralized timestamp server model.

The second layer of security is that nodes running Wormhole protocol do not process data that failed to follow Wormhole protocol specification. Each node has the ability to reanalyze transaction data and recalculate the most "recent legal final state".

Wormhole Cash (WHC)

Wormhole cash (WHC) is the native token for Wormhole protocol. The introduction of WHC is necessary because Wormhole protocol layer has no control of BCH layer when executing smart contract and in this case there is no way to realize transaction. Besides, we need WHC to serve as gas when executing smart contract to prevent abuse of BCH network.

WHC Generation

WHC is generated by the **Proof-of-Burn (PoB) mechanism**. BCH users can send one or more BCH to address qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu08dsyxz98whc to generate WHC after official launch of Wormhole protocol. If less than one BCH is sent to the address, then no WHC will be generated. This "**burn to generate**" process is subject to rollback risk of the underlying blockchain. Therefore, WHC can only be generated after 1000 confirmations of the corresponding transaction. 1 BCH can get 100 WHC.

Based on current cryptographic knowledge and engineering experience, nobody owns private key of the burning address: qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu08dsyxz98whc. According to BCH ledger, this is still a fresh new address when we launch the Wormhole project.

If there are market demands around WHC and liquidity is high, users who need WHC can also purchase WHC from the market.

How about the two-way anchoring of WHC and BCH? Unfortunately, currently there is no feasible two-way anchoring method that is safe, decentralized, and can effectively deal with the inevitable rollback risk of blockchain. When discussing space travel, Elon Musk said that he was not coming back after immigrating to Mars. The way of WHC generation via PoB mechanism works the same way as the one-way ticket to Mars. Each Satoshi (BCH) will never be back and there is no deadline to the burning process. Wormhole protocol implements smart contract and adopts different programming language, and there is a rapid development plan ahead.

WHC Use Cases

Fees are often used to prevent network abuse. Smart contract execution is realized via BCH transactions. The fee requiements of the Bitcoin Cash transaction can effectively cope with DoS attacks. Therefore, in early phase of Wormhole protocol, no WHC fees are demanded for transfer.

Yet there are several cases where one need to pay transaction fee with WHC:

1. Creating new token demands transaction fee of 1 WHC. Transaction fees will be burned directly and thus the total supply of WHC will be reduced. WHC fees are designed to prevent malicious attacks to Wormhole nodes as token creation consumes certain computing resources.
2. One-to-many transfer. For example, to send token to all addresses that possess a certain token. Such kind of operation needs to iterate over many addresses and consumes considerable resources.
3. Gas for smart contract.
4. Other transactional operations or activities that are suspected of being DoS attacks.

Token Issuance

Anybody is free to create tokens on the system after paying reasonable BCH transaction fee and the specified WHC fee. Currently, WHC protocol supports three types of token creation:

1. Fixed token
    a. After creation, the creator automatically owns all tokens immediately
    b. Cannot be increased, cannot be burned
    c. Cannot trigger crowdfunding
2. Token supports crowdfunding
    a. Automatically trigger crowdfunding after creation
    b. The creator does not own all tokens after creation
    c. Tokens left after crowdfunding goes automatically to creator’s address.
    d. Cannot be increased, cannot be burned
3. Manageable token
    a. Initial number of tokens is 0 after creation
    b. Cannot be used for crowdfunding
    c. Can be increased, can be burned

Token Transfer

Tokens issued within Wormhole protocol and the native token WHC can be transferred. 1-to-1 transfer requires only BCH transaction fee, no additional fee is required. BCH transaction fee is up to BCH network.

In addition to BCH transaction fee, one-to-many transfer also requires certain WHC fees, which is denominated and charged in WHC. The one-to-many transfer is mainly used for token airdrops. WHC fees will be burned directly.

Token Burning

Manually managed token supports direct burning. The total number of the token (after burning) will be updated after burning.

Wormhole Roadmap

The development of Wormhole protocol is divided into four phases: Earth, Tropos, Ionize, and Exophere.

Earh (TTC: August 2018)

Forked from Omni layer protocol, Wormhole protocol aims to be the smart contract solution for BCH. The first step will focus on realizing decentralized token issuance function.

To ensure the security and to launch the project ASAP, we delay the support of decentralized trading on Omni Layer protocol to next phase.

Tasks to be completed:

- Release Wormhole protocol white paper
- Implement token issuance upon Bitcoin ABC v0.17.2 and update in accordance with Bitcoin ABC's update

Tropos (TTC: November 2018)

Tasks to be completed:

- Implement Wormhole protocol-based decentralized exchange protocol
- Wormhole's Android wallet reference implementation
- Wormhole's iOS wallet reference implementation
- Wormhole‘s PC wallet reference implementation

Ionize (TTC: January 2019)

Tasks to be completed:

- Implement ERC721 protocol in Wormhole protocol
- Provide multi-language SDK to facilitate application developers of Wormhole
- Cold wallet for Wormhole

Exophere (TTC: June 2019)

Tasks to be completed:

Permissionless smart contract: Omni layer is not a mechanism for permissionless innovation. Only by hardcoding into the protocol implementation can new contracts be deployed. An unlicensed smart contract platform will be implemented in Exophere phase. That is, any developer can deploy smart contract in the network as long as the smart contract complies with certain security rules.

Implement Plasma protocol for scaling: after heavy internal research, we may have discovered an effective approach to realize Plasma. Meanwhile, Vitalik also announced on Twitter that they have found a way to implement Plasma. We will choose either way to implement Plasma protocol as we see fit.

A new generation of smart contract virtual machine: as a contract-oriented language that can be applied to a variety of different blockchain platforms, Solidity has borne scrutiny from computer experts in recent years. Lots of new ideas have been proposed recently. A new type virtual machine will be realized to enable building DApps with efficient and developer-friendly programming language.

Thanks and Summary

We'd like to give credit to Omni layer protocol. Its extensive use on USDT gave us confidence that more things can be done based on BCH. Omni protocol takes full advantage of the features of the UTXO model and enables token management without any modification to the consensus rule of the underlying network. Omni team helped us a lot during the development of Wormhole project. Meanwhile, Omni layer protocol sticks to the spirit of open source and adopts MIT open source license, which makes permissionless innovation possible.

UTXO model based public chains have been struggling to realize smart contract. Wormhole protocol can enable smart contract on BCH and open new possibilities to BCH.

Document History

- Version 0.1 Wormhole Cash Completion of Phase 1 (May 23, 2018)
- Version 0.2 Wormhole Cash Roadmap (June 20, 2018)
- Version 0.3 Wormhole Cash alpha version (July 15, 2018)

References

[1]. Satoshi Nakamoto. Bitcoin: A Peer-to-peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.

[2]. OP_RETURN https://en.bitcoin.it/wiki/OP_RETURN

[3]. OmniLayer https://github.com/OmniLayer/spec

[4]. ERC20 Token Standard https://theethereum.wiki/w/index.php/ERC20_Token_Standard

[5]. The Colored Coins Protocol https://github.com/Colored-Coins/Colored-Coins-Protocol-Specification/

[6]. Andrew Stone : Enable representative tokens via OP_GROUP on Bitcoin Cash
https://github.com/BitcoinUnlimited/BUIP/blob/master/077.mediawiki

[7]. ERC-721 http://erc721.org/