Chain Interoperation in E-Government System of Shanghai North Bond

MAP Studio

October 25, 2020

Introduction

Background

Since Chairman Xi emphasized Blockchain technology as an important breakthrough in independent innovation of core technology in 2019, blockchain technology is widely used in multiple application fields. E-government is one example which takes advantage of the unforgeability and transparency of blockchain.

What problem we want to solve?

Issues of blockchain application in E-government

Most applications of E-government are distributed in different blockchain. And these blockchains can not exchange data and information directly, which means each blockchain are information isolated island, as well as such applications. Thus, achieving chain interoperability among these blockchains is a critical issue for integrating blockchain technology into E-government.

What is the current solution?

Chain interoperation through Trusted Third Party(TTP)

Most chain interoperation solutions require TTP to provide information of another blockchain for crosschain validation. TTP could be a single node or a relayer group. These solutions need additional trust upon TTP.

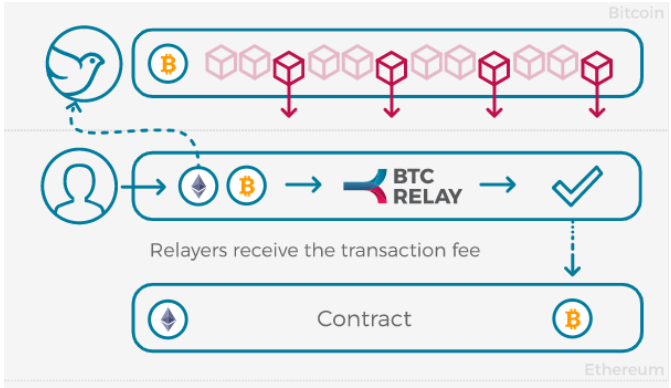

Figure: BTC-Relay need relayers as TTP for crosschain validation

What is our solution?

Chain interoperation through Ultra-Light Verification Protocol(ULVP)

We suggest to use ULVP based on flyclient technology proposed on 2019. Through ULVP, node can verify any transaction of one blockchain without synchronizing all the block header. Thus crosschain validation could be accomplished without a relayer group.

Application scenario of our project

Application scenario

Suppose there are two blockchain: one is called ID chain which manage ID information of resident maintained by Police Station of Shanghai North Bond, and another is called Passport chain which manage the passport information maintained by Exit-Entry Administration. User who register his information in ID chain want to apply a passport.

Design of our project

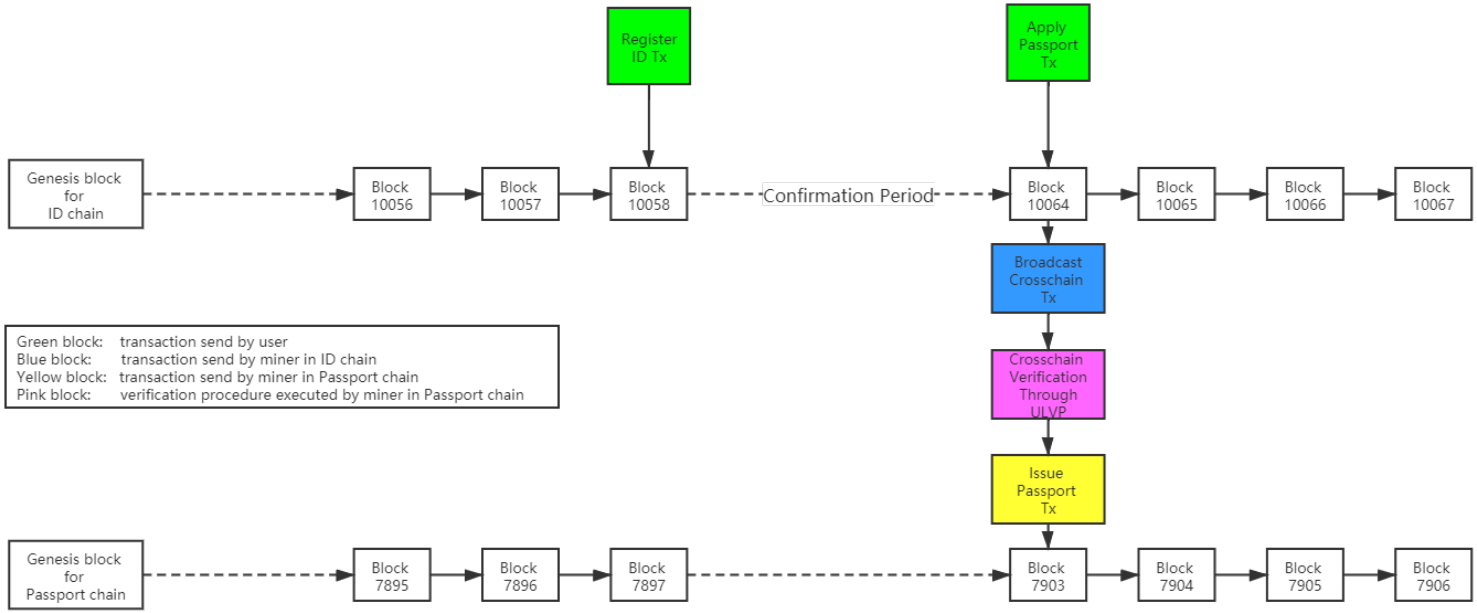

Figure: Our Design

Demonstration of our project

Let's jump to our demonstration now!

Details of ULVP

Simplified payment verification(SPV)

User hold a full copy of all block headers in a blockchain. And if he need verify the existence of a transaction in one block, he would request some randomly selected full node send the corresponding merkle branch proof of this transaction. He could then check its validity use the merkle root in his own block header.

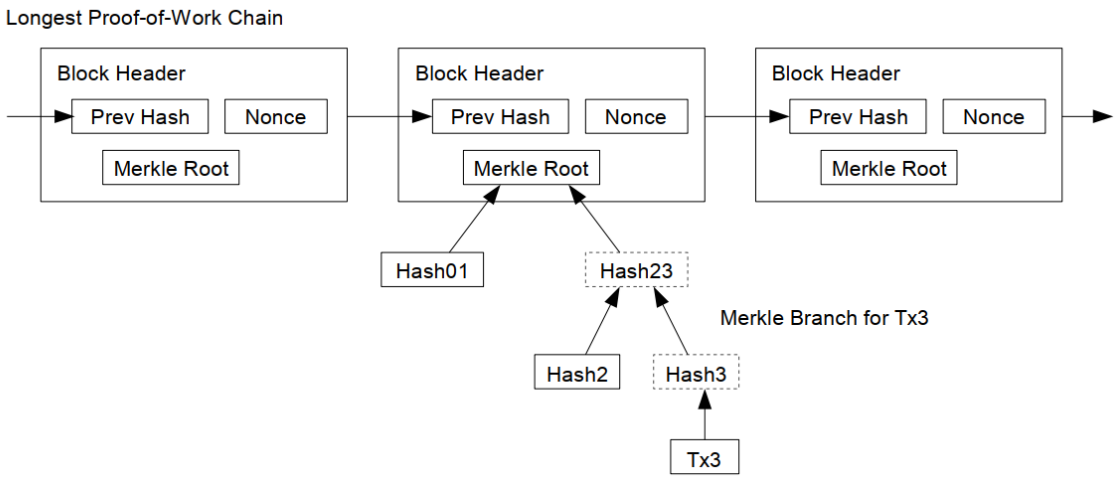

Figure: Simplified payment verification

Details of ULVP

Merkle mountain range(MMR)

MMR is a variation of Merkle tree. The advantage of MMR over the Merkle tree is that when new data arrives, the value of the intermediate node does not change, thus the data is always appended. ULVP would request each block header contain a MMR root of all previous headers. If node hold the latest header, he could retrieve any previous header using a MMR branch proof.

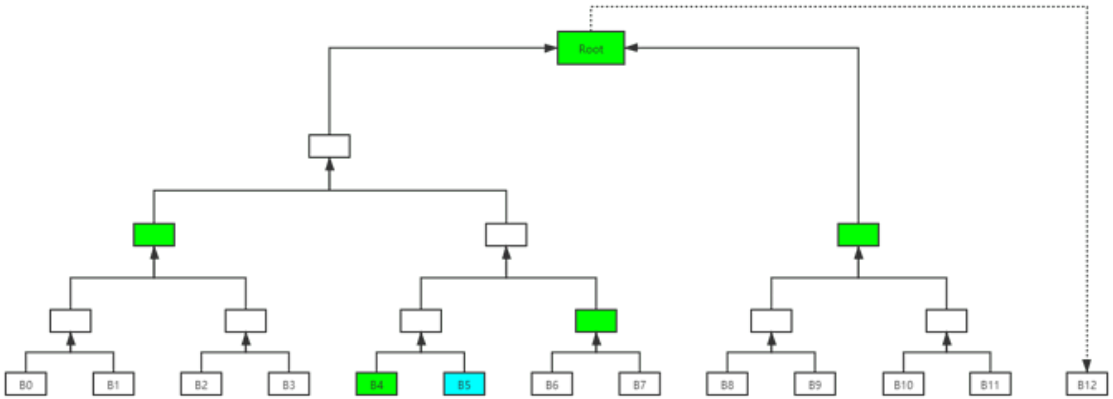

Figure: Merkle mountain range

Details of ULVP

How ULVP works(I)

ULVP request the full node provide a few block headers and its MMR branch proof for block header validation. Under honest majority assumption, adversary could not generate more block than honest party. In SPV setting, he could not cheat since verifier need the whole chain of block header. In ULVP setting, adversary could fill some fake block to increase his chance to cheat.

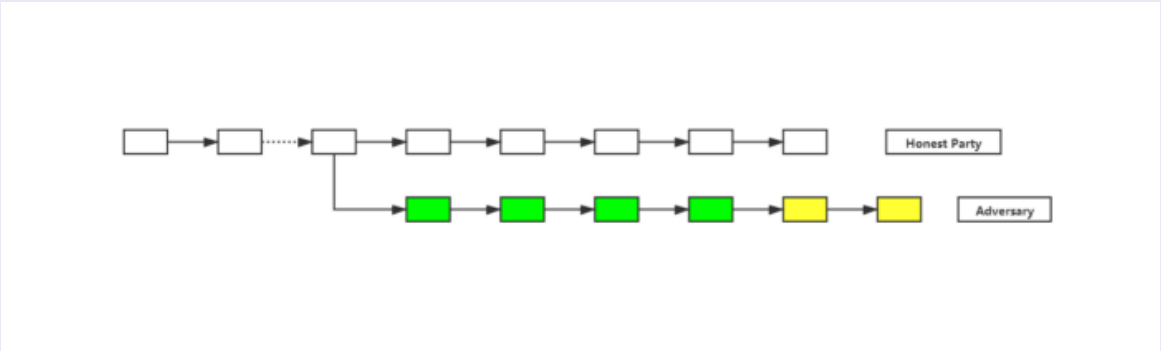

Figure: Cheating strategy of Adversary

How ULVP works(II)

In order to catch the fake block filled by adversary, we will use a random sampling strategy. In flyclient paper it proves that if we sample certain size of headers using the optimal density function, the adversary could succeed cheat verifier with negligible probability.

Corollary 2. *Under the (c, L)-adversary assumption for any constant L and using a collision-resistant hash function the FlyClient proof size is $\Theta(\lambda \log(n) \log_{\frac{1}{c}}(n) + L)$*

Figure: Sampling size

Theorem 2 (Optimal Sampling Distribution). *Given that the last $\delta = c^k, c \in (0, 1], k \in \mathbb{N}$ fraction of the chain contains only valid blocks and the adversary can at most create a c fraction of valid blocks after the fork point a, the sampling distribution defined by the PDF $g(x) = \frac{1}{(x-1)\ln(\delta)}$ maximizes the probability of catching any adversary that optimizes the placement of invalid blocks.*

Figure: Optimal density function

Details of ULVP

How ULVP works(III)

- First, verifier connect to a randomly selected prover. The prover would sample certain block headers and generate corresponding MMR branch proof. Then prove will send these to verifier.
- Verifier would check the validity of these proof. If it can not pass the check, disconnect the prover and request another prover sending proof.
- Once verifier get the valid proof. He could further request transaction inclusion proof similar to SPV.

Summary

Summary of Our Project

- We developed two E-government blockchains, one can manage ID information and another can manage passport information.
- We realized ULVP which enable the chain interoperability of these two E-government blockchains.
- We implemented the frontend of each blockchain system for ID registration and Passport application.

References


Nakamoto S. “Bitcoin: A Peer-to-Peer Electronic Cash System.” Bitcoin Project. (2008)


Benedikt Bünz, Lucianna Kiffer, Loi Luu, and Mahdi Zamani. FlyClient: Super-Light Clients for Cryptocurrencies. (2019)


Our Demo: https://github.com/wanxiang-blockchain/hackathon2020-T16-mapstudio


Some Technical paper: www.hashwei.cn/c/interchain/8

Thank You!