Bitcoin & Ethereum Cross-chain Atomic Swap

A Trustless Method of Exchanging Bitcoin For Ether Between Two Peers

Bachelor's Thesis submitted to the

Computer Science Engineering, Software and Multimedia developement orientation of the *Haute Ecole Arc Ingénierie (HES-SO), Switzerland*

in partial fulfillment of the requirements for the degree of

Bachelor of Applied Science in Computer Science

presented by

Luca Srdjenovic

under the supervision of

Prof. Ninoslav Marina

co-supervised by

Thomas Shababi

July 2019

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Luca Srdjenovic
Neuchâtel, 26 July 2019

Abstract

Atomic swaps are pratical for exchanging different cryptocurrencies in avoiding any trusted third-parties. This project show a swap between Bitcoin and Ethereum blockchain using payment channels tools like hashlock or timelock. When the protocol is followed by the both participants, it guarantees the swap without any risk. In the opposite, there is no scenario where someone can control both coins.

Keywords: Bitcoin, Ethereum, Atomic Swap

Acknowledgements

Thank you

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 2

Bitcoin, a peer-to-peer network

Chapter 3

Ethereum, A Decentralised Computing Platform

3.1 Test

Chapter 4

Atomic Swap, A Method of Exchanging Different Cryptocurrencies

Definition : `Atomic Swap is the process of peer-to-peer exchange of two cryptocurrencies between two parties, without using any third-party service like crypto exchange.`

In few explication, an atomic cross-chain swap is a smart contract distributed where two parties or more exchange two cryptocurrencies across different blockchains. It is called cross-chain because you are no longer dependant on the blockchain. An atomic swap protocol guarantees if both parties follow the protocol, then all swaps take place. But if one of the two parties deviates from the protocol, then no conforming party and the no coalition produce automatically the cancel of the swap. At any moment, no one can control both coins, hence no coalition has an incentive to deviate from the protocol.

4.1 Atomicity

`Atom` comes from Greek and means 'a' -not/un, 'tom' -cut, in other word, no divisible or cuttable. It means that an atomic transactions cannot be splittable into parts. We use the familiar expression `all or nothing` in atomic where it is the same applied concept in bitcoin. For example, Alice pays Bob in one transaction, they all know that either Bob will be paid or either bob won't. There is only two ways, the transaction is confirmed or not but there is no way for having an half-confirmation. That's the reason why the atomicity is fundamental in atomic swap, to protect both parties, there must be no scenario in which one part can control both coins at the same time.

An other example, no atomic transaction for illustrating is when Alice wants buy something in a web store. First, she needs to transfer the money to the site and then waits for the store send her the object back. Here there always is a chance that Alice doesn't get her purchase.

4.2 Difference with Payment Channels

In Bitcoin, Payment Channel is class of techniques designed to allow users to make multiple Bitcoin transactions without committing all of the transactions to the Bitcoin block chain. In a typical payment channel, only two transactions are added to the block chain but an unlimited or nearly unlimited number of payments can be made between the participants.[1] It is faster, cheaper transactions between parties because each transaction doesn't need to be written to the blockchain. Therefore there is only the net result of multiple transactions.

Atomic swap is not a payment channel but uses tools of it like Hashed Timelock Contracts (HTLC), a technique that can allow payments to be securely routed across multiple payment channels that we describe below (see chapter 4.3). It is a concept from the Bitcoin community that is used in the Lightning Network.

4.3 Security by Hashed Timelock Contracts

Atomic swaps uses HTLC (Hashed Timelocke Contracts), which are part of the scripting language used by most major cryptocurrencies in existence right now. Both parties involved in a cross-chain transaction submit their individual transactions to the appropriate blockchain.

HTLC is a kind of smart contracts that allows to eliminate counterparty risk using tools like hashlock and timelock. It enables time-bound transactions between the two parties. A time-bound means when a recipient at the other end of the transaction is required to acknowledge the transaction, the person needs to provide a cryptographic proof. The person also needs to provide that cryptographic proof within a time-frame. In case of failing so will automatically make the transaction null and void. In practical terms, this means that recipients of a transaction have to acknowledge payment by generating cryptographic proof within a certain timestamp. Otherwise, the transaction isn't valid. The cryptographic proof of payment that the receiver generates can then be used to trigger other actions in other payments, making HTLC a powerful technique for producing conditional payments in Bitcoin.

There are many benefits for HTLC :

1. It prevents the person who is making the payment from having to wait indefinitely to find out whether or not his or her payment goes through.
2. The person who makes the payment will not have to waste his or her money if the payment is not accepted. It will simply be returned.
3. The recipient actually helps to validate the payment on the blockchain because cryptographic proof of payment is required for the recipient to accept the payment.
4. The hashes that are created for the HTLC can be easily added to blockchains.
5. The structure of the method allows the people sending and receiving the payments do not have to trust each other or even know each other to make sure that the contract will be executed properly. In other words, each party is protected from counterparty risk.

[1] Micropayment channel: Bitcoin.org Developer Guide

To work, A Hashed Timelock Contract implements several elements from existing cryptocurrency transactions. The concept of signatures, HTLC uses multiple signatures that consists of using a private key and public key to verify and validate transactions. The main elements that make HTLC a powerful method are the concept of `hashlock` and `timelock`.

Hashlock

A hashlock is a type of encumbrance that restricts the spending of an output until a specified piece of data is publicly revealed. Hashlocks have the useful property that once any hashlock is opened publicly, any other hashlock secured using the same key can also be opened. The hashlock is a scrambled version of a cryptographic key generated by the originator of a transaction. By encumbering transaction outputs with a hashlock and timelock, the channel counterparty will be unable to outright steal funds and coins can be exchanged without outright counterparty theft.

CheckLockTimeVerify (CLTV)

In addition to hashlock, it also uses timelock. Hashed Timelock Contracts use two different types of timelocks during an Atomic Swap. The second important element of HTLC is a timelock. CheckLockTimeVerify (CLTV) uses a time base to lock and release bitcoins. This means that time constraints are hard coded and coins are released only at a specific time and date or a specific height of block size.

```
1 IF
2     <provider pubkey> CHECKSIGVERIFY
3 ELSE
4     <expiry time> CHECKLOCKTIMEVERIFY DROP
5 ENDIF
6 <client pubkey> CHECKSIG
```

Listing 4.1. Example of locking script with CheckLockTimeVerify.

CheckSequenceVerify (CSV)

The second one is CheckSequenceVerify (CSV). It is not dependent on time. Instead, it uses the number of blocks generated as a measure to keep track of when to finalize a transaction.

```
1 IF
2     <provider pubkey> CHECKSIGVERIFY
3 ELSE
4     <expiry time> CHECKSEQUENCEVERIFY DROP
5 ENDIF
6 <client pubkey> CHECKSIG
```

Listing 4.2. Example of locking script with CheckSequenceVerify.

Chapter 5

Protocol

We describe a protocol for an on-chain atomic swap between Bitcoin and Ethereum, but the protocol can be generalized for Ethereum and any other cryptocurrencies that fulfill the same requirements as Bitcoin (e.g. LiteCoin), see **??**. This protocol is heavily based on the BIP 199 (Bitcoin Improvement Proposal (BIP)) [Bowe and Hopwood, 2017] for the Bitcoin part. For Ethereum the concept is roughly the same but with less prerequisites than Bitcoin. For sending funds, each participant must generate a specific address to lock fund on each chain (cross-chain) where each other party can take control of the funds from the other chain (swap) only.

5.1 Limitations

The most important process of the protocol is the `liveness`. Liveness means that participant must be online for respecting the protocol (at least one participant is still online). In the worst scenario where someone doesn't follow the procotol, it can happen the coalition end up and loose the funds. This happen only if a party is not remained online during the swap or it has not claimed the funds in time.

5.2 Scenario

test

5.3 Prerequisite

5.4 Hashed Timelock Contract

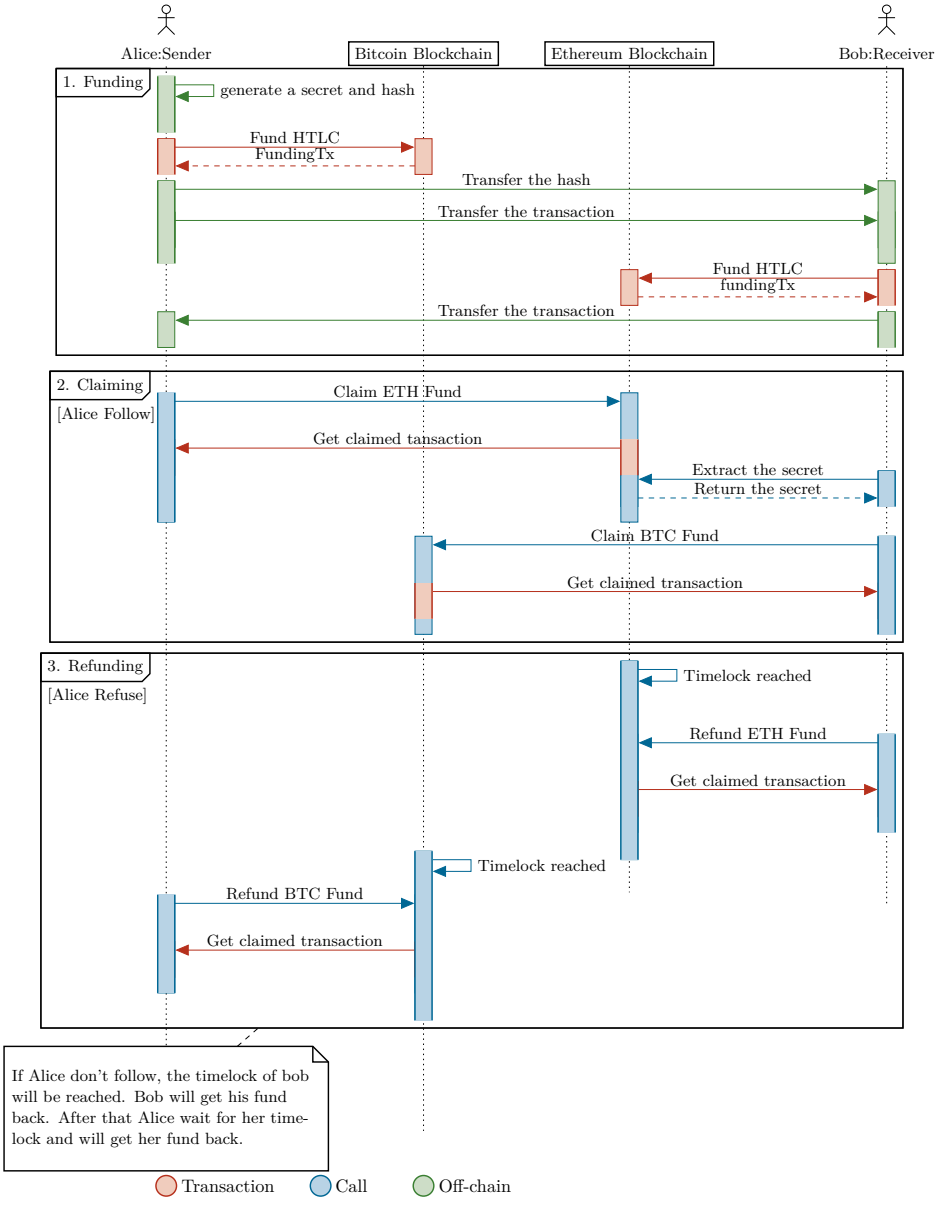

Figure 5.1. Sequence of atomic swap protocol.

Chapter 6

Implementation in Bitcoin

saksaksakas

6.1 test

Chapter 7

Implementation in Ethereum

Chapter 8

Observation

Test Web3 for testing.
Test [Buterin, 2013]. # Conclusion
Conclusion
saasjsajsajasjsajjjjjjjjjjjjjjjjjjjjjjjjjjsajsajsasajsajsajjasjsajsajasasjasjjasaj

List Of Abbreviations

BIP Bitcoin Improvement Proposal. 11

HTLC Hashed Timelock Contracts. 8

Glossary

Web3 Web3 often refers to `web3js`, the Javascript implementation of the Ethereum JSON-RPC. It may also refer to other implementation in different languages. Overall it is the technology aiming to build the next and more decentralised version of the web 2.0 we know today. 17

Bibliography

Sean Bowe and Daira Hopwood. Hashed time-locked contract transactions, Mars 2017. URL `https://github.com/bitcoin/bips/blob/master/bip-0199.mediawiki`.

Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform, 2013. URL `https://github.com/ethereum/wiki/wiki/White-Paper`.