Jelly Swap Ecosystem

Cross-chain atomic swaps from wallet to wallet

Tito Titov , Krasimir Raykov
admin@jelly.market

(v.01, updated on 4.10.2019)

CONTENTS

1. INTRODUCTION

The very aim of digital blockchain assets and cryptocurrencies is to achieve a truly open and free financial system accessible for everyone. Such system will fulfill the idea of having programmable money to help people exchange assets digitally over the blockchain.

The first major attempt was made by the centralized exchanges. The problem here is that they use the blockchain technology but keep the well-known trusted third-party model where users must give up the custody of their own assets to third-party providers such as Binance or Kraken. Once you register there, you have to fully trust these systems not to get hacked, expose your assets to risk, or use some malicious tactics such as front-running.

Further steps in achieving an open financial system were made by the Decentralized Exchanges (DEXs). In contrast to the centralized platforms above, the DEXs operate in a fully decentralized way, meaning that you (as a user) keep the control of your own assets; you do not provide personal information or perform any KYC/AML procedures; the trading is happening peer-to-peer without any interference of third party. Although, theoretically DEXs sounds flawless, there are some problems here as well. Platforms such as Uniswap and KyberSwap can only do ETH Token-to-ETH Token Transactions (e.g. ERC20 token only transactions). But what happens if you want to exchange Bitcoin for Ether?

Here comes the latest innovation in Decentralized Finance - cross-chain transactions executed by Atomic Swaps without any intermediaries.

In this paper, we introduce a multichain swap service called Jelly, which aims to achieve decentralized cross-chain trading by eliminating the need for trusted third party (TTP) entity.

2. JELLY SWAP

Over time, the Jelly Swap Ecosystem strives to streamline the process of providing cross-chain liquidity to main blockchain networks and make the multichain swap simple and secure. Jelly suppliers provide liquidity to the protocol by contributing a diverse range of coins. Makers are comprised of a wide range of parties, from project teams looking to list their tokens, to professional market makers with customized trading strategies, VCs and institutional traders and even developers looking to build new jelly liquidity pools. (Table 1)

Jelly Liquidity Pools	Security	Requirements	Profit from spreads	Risks
Run your node	High	24/7 running machinea	Only for you	System Failure Market volatility Hack on owners machine
Work as organisation with common wallet	Medium	1 machine running 24/7 per organization	Shared between organization	System Failure Market Volatility Hack on organization machine Internal organizational hacks
Use cloud hosted service	Medium	Pay monthly fee for cloud support	Only for the owner	System Failure Market volatility Hack on cloud machine

TABLE 1

There are currently 3 ways that anyone can help contribute coins to the atomic protocol. Users with technical skills can choose to run their own node and aggregate. Alternatively, a non-technical user can take part in jelly liquidity pool organization and build a pool with people that they believe in. The team might maintain an official organization wallet, where everyone can join. Lastly, technical or non-technical users can decide to use a cloud based solution and to maintain a cloud liquidity pool in a few clicks. The key benefit jelly liquidity pools provide is *instant cross-chain liquidity*. Takers such as dApps, vendors and wallets that integrate the protocol are able to immediately convert one coin into another because of the liquidity made readily available by the liquidity pools. (Figure 1)

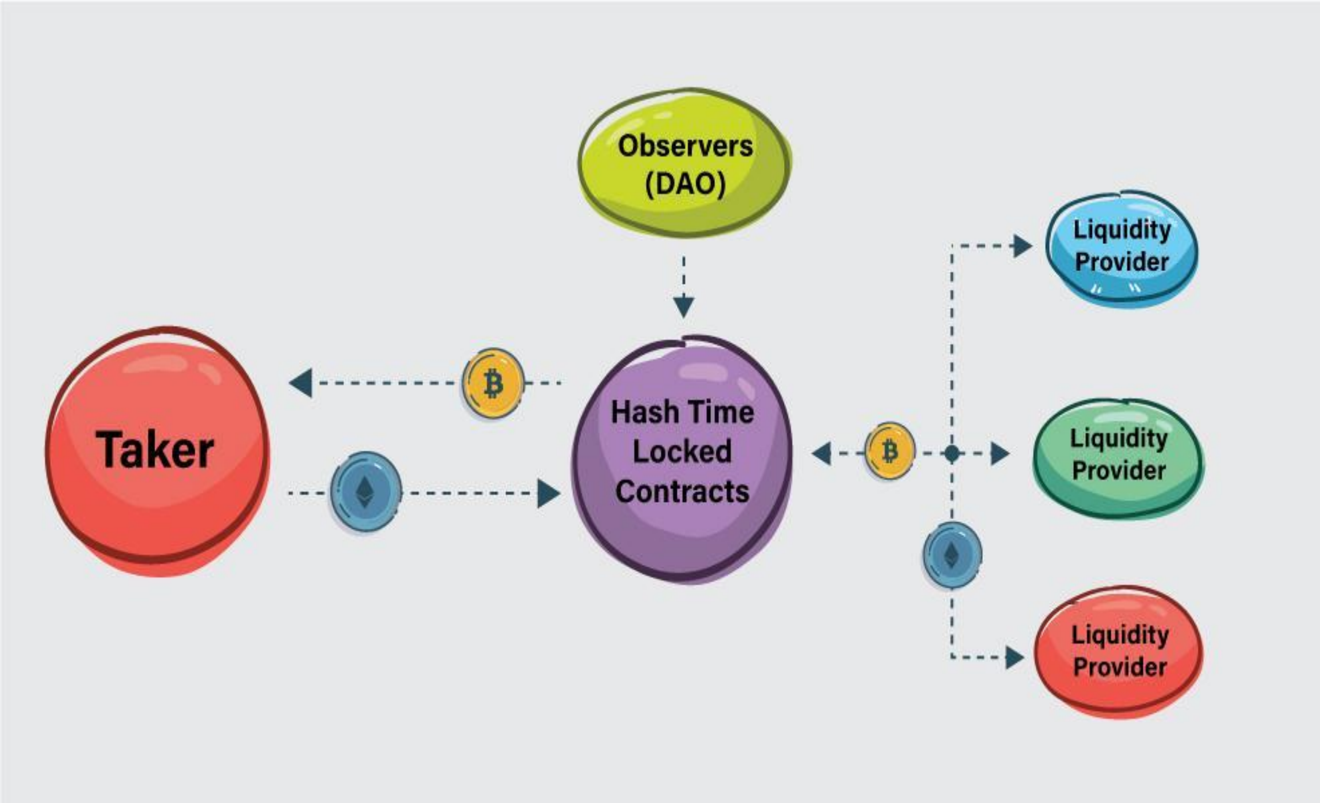

FIGURE 1

3. KEY TECHNOLOGIES AND ALGORITHMS?

The cross-chain atomic swap is possible, because of these key technologies:

1. Cryptographic Hash Function [7] - A **cryptographic hash function** (**CHF**) is a hash function[11] that is suitable for use in cryptography. It is a mathematical algorithm[13] that maps[15] data of arbitrary size (often called the "message") to a bit string[14] of a fixed size (the "hash value", "hash", or "message digest") and is a one-way function[16], that is, a function which is practically infeasible to invert. Ideally, the only way to find a message that produces a given hash is to attempt a brute-force search[17] of possible inputs to see if they produce a match, or use a rainbow table[18] of matched hashes. Cryptographic hash functions are a basic tool of modern cryptography

2. Time Locked Transactions - the receiver has a limited amount of time to confirm on the blockchain that he has received the funds. It guarantees that the blockchain will reverse the transaction[19] if the receiver never confirms it and the assets will be sent back to the sender.

3. Hash Locked Transaction - a special secret key (different from the private key) used to unlock the transaction and get the funds.
   ** the hash is derived from the secret key but it is a one-way process - there is no way to go back from the hash and reveal the secret key. While the hash is visible for the receiver, the secret key is the only piece of information which can unlock the transaction.*

4. Smart Contract[8] - A **smart contract** is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract[10]. Smart contracts allow the performance of credible transactions without third parties. These transactions are trackable and irreversible.

5. Bitcoin Script[9] - Bitcoin uses a scripting system for transactions. Forth-like, **Script** is simple, stack-based, and processed from left to right. It is intentionally not Turing-complete, with no loops.

A script is essentially a list of instructions recorded with each transaction that describe how the next person wanting to spend the Bitcoins being transferred can gain access to them. The script for a typical Bitcoin transfer to destination Bitcoin address D simply encumbers future spending of the bitcoins with two things: the spender must provide:

1) A public key that, when hashed, yields destination address D embedded in the script, and
2) A signature to prove ownership of the private key corresponding to the public key just provided.

Algorithm in example:

1. Alice has BTC, Bob has ETH
2. Alice creates a secret key consisting of random letters, words or numbers.
3. A hash is created from Alice's secret key
4. Alice submites a transaction on the Bitcoin blockchain which is sent to Bob's address. It is locked in both of terms of time (where Bob has a certain amount of time to confirm it) and in terms of security (with the hash).
5. Alice sends the hash to Bob
6. Now it is time for Bob to create his Ethereum transaction to Alice (again it is both time and hash locked, where the hash is the same as Alice's). As a result, we have a transaction on both the Ethereum and the Bitcoin blockchains which is locked with the same hash.
7. Alice unlocks Bob's Ethereum transaction (because she holds the secret key). By unlocking his transaction, Alice reveals the secret key and it is publicly recorded on the blockchain.
8. Bob can now see the secret key and unlock the Bitcoin transaction which was aimed at him.
9. They have exchanged assets without the need to trust or know each other.

4. FINANCIAL INCENTIVES

Jelly is a p2p permissionless cross-chain swap protocol for instant liquidity. Everyone can participate, provide liquidity, and execute swaps via HTLC contracts.
What is the incentive to be a taker or liquidity provider?
Should I take part in the Jelly protocol?

There are many reasons and incentives for the Jelly protocol participants. We can use as reference Kyber Network and Uniswap. They are liquidity aggregators, simple and efficient Ethereum based swap protocols. The whole DeFi ecosystem is built around these two products. Can you imagine multichain uniswap with much more liquidity, many different blockchains, many more trading opportunities, a brand new interchain trading financial system - trustless, secure and decentralized. All these things are possible thanks to hash time locked contracts technology, and the weiDex team have developed and packed the whole process in an elegant and user-friendly manner.

What is the incentive of takers to use Jelly Swap?

- Access to instant liquidity
- Innovative trustless cross-chain swaps e.g. ETH<>BTC, ETH<>TRX, ETH<>ADA
- Open/Close interchain long/short positions
- No profile registration or coin swap risks
- Fast and reliable p2p service
- Easy to embed in different wallets and DeFi products

What is the incentive of Jelly to provide liquidity?

- Gain profit from price spreads
- Maintain diversified portfolio and earn coins
- Run a secure, trustless and semi anonymous node with lowest possible risks
- Take part or get out of the protocol at any time
- No restrictions and instant earnings in liquidity provider’s wallet on every trade
- Support the whole blockchain ecosystem

5. JELLY LIQUIDITY POOLS TYPES

There are three main types of Jelly Liquidity Pools. The main difference between them is the governance, security level and technical maintenance.

5.1.1. Type One - Alpha Liquidity Pool:

Self-hosted Liquidity Pool node. This is the most secure and highly recommended way to join the jelly protocol. As self-hosted Liquidity Pool you can control your portfolio and wallet security. You can set different price and tolerance rates. You can provide liquidity for different pairs ETH<>BTC, ETH<>TRX, ETH<>ADA etc. The one and only downside is that you have to maintain the machine that runs the node and manage your portfolio by yourself.

5.1.2. Type Two - Tribe Liquidity Pool:

Crowd hosted Liquidity Pool node. This is relatively insecure choice. As crowd hosted Liquidity Pool you can vote for portfolio management and wallet security is in the hands of the tribe chief. You can’t set different price and tolerance rates if it is not discussed with the tribe. You can provide liquidity for different pairs if the tribe votes for it e.g. ETH<>BTC, ETH<>TRX, ETH<>ADA etc. The biggest benefit is that you do not have to maintain the machine that runs the node and to manage your portfolio by yourself. Last but not least - “Follow the wisdom of the crowd”.

5.1.3 Type Three - Dodger Liquidity Pool:

Cloud hosted Liquidity Pool node. This is the relatively secure and probably the easiest way to join the jelly protocol. As cloud hosted Liquidity Pool you can control your portfolio and wallet security. You can set different price and tolerance rates. You can provide liquidity for different pairs - ETH<>BTC, ETH<>TRX, ETH<>ADA etc. The one and only downside is that you have to trust your cloud provider and to manage your portfolio by yourself.

6. JELLY LIQUIDITY POOL REQUIREMENTS

All of the nodes have common interface. They are using the same source code, therefore they provide a few main methods:

- Current Conversion Rate
- Market Making Addresses
- Available Balance in Liquidity provision addresses
- Create HTLC
- Withdraw HTLC
- Refund HTLC

Jelly Liquidity Pools are part of the protocol as long as their node is running and they are acting according to the rules.

What can get wrong?

- Create, Withdraw or Refund service is not working for some reason
- Liquidity provider want to change a deal parameters, which are already approved
- Liquidity provider is running malicious software in order to deceive the taker
- Taker can try to act dishonestly and to ruin the deal

In all these cases both parties can be sure that their funds are save and the whole process is atomic (the deal happens or funds are securely stored on each participants wallet). This is valid as long as the liquidity providers node is up and running.

7. TRADING STRATEGIES

There are many trading strategies that can be used by the market makers, but we will list three simple strategies that can be used as a starting point.

7.1.1. Long and Short positions

Let’s analyze a theoretical situation where I believe that the price of ETH will go up and the price of TRX will go down. Simply said I am Long on ETH and Short on TRX. I will provide liquidity on ETH/TRX, therefore everyone will be able to exchange ETH for my TRX with a little price spread, which is an extra profit for the liquidity provider. With this strategy I will convert my TRX to ETH, without paying any taxes and even a small profit. If the price of ETH goes up, I will earn a good profit, because all of my funds will be in ETH.

7.1.2. Lazy Strategy

Let’s analyze a theoretical situation where I believe that the price of ETH and TRX will stay at the same rare for the next months. Simply said I am in a neutral position. I will provide liquidity on ETH/TRX and TRX/ETH, therefore everyone will be able to exchange ETH for my TRX and TRX for my ETH with a little price spread, which is an extra profit for the liquidity provider. With this strategy I will try to keep my balance and as long as the price of ETH and TRX is stable I will have some profit from the price spread, without doing any complex computation, just via running the liquidity node, supporting both of the markets.

7.1.3. Portfolio Rebalancing Strategy

Let’s analyze a theoretical situation where I believe that I want to keep the ratio of my ETH/TRX to 80/20. I want to keep this ratio, but I would like to earn some income from the price spread of each swap. I will provide liquidity on ETH/TRX and TRX/ETH, therefore everyone will be able to exchange ETH for my TRX and TRX for my ETH with a little price spread, which is an extra profit for the liquidity provider. With this strategy I will try to keep my balance ratio and earn some profit. If the ratio is changed e.g. ETH/TRX 70/30, I can stop my liquidity provision on ETH<>TRX and to provide liquidity only on TRX<>ETH, until my balance ratio is fine. The second option it to go to some other exchange and to rebalance my portfolio. At the end of the day, I should have some income based on the price spreads.

8. PLANNED UPDATES

1. Notification System
    a) Liquidity providers will be notified via email if their node is not operational
2. Rating System
    b) Liquidity providers and takers will have ratings based on the success rate of their swaps, their node availability and conversion rate
3. Hide my IP
    c) Liquidity providers will be able to configure their spread and tolerance parameters via our UI, and they will be able to hide their machine origin
4. Portfolio management
    d) Liquidity Providers and Takers will be able to check their swaps, profit/loss and rating

9. CONCLUSION

Although the crypto market is unpredictable and has risks, it is better to earn some income providing liquidity, rather than just hold your funds. Kyber Network, Uniswap and Compound have proved that on Ethereum level. Jelly accelerates this strategy on a multi-platform level, following the same philosophy and the highest security standards. The platform offers a real opportunity for crypto investors and traders to diversify their coins in faster, cheaper and secure way. The technology has serious potential to bring us to the next evolution of crypto trading by eliminating scalability and interoperability problems, the need for trusted third party entity and limitations of single blockchain transactions. Multi-chain trading is significant achievement which will help us accelerate the crypto adoption.

10. REFERENCES

1. Kyber Network documentation: https://developer.kyber.network/docs/Start
2. Uniswap documentation: https://docs.uniswap.io
3. Atomic Swap Overview: https://medium.com/@titotitov/cross-chain-atomic-swaps-4bc81ae399ef
4. Trading strategies Part I:https://medium.com/weidex/liquidity-strategies-for-decentralized-exchanges-part-i-f700461be5db
5. Trading strategies Part II: https://medium.com/weidex/liquidity-strategies-for
6. decentralized-exchanges-part-ii-1947d3e61566
7. Cryptographic hash function: https://en.wikipedia.org/wiki/Cryptographic_hash_function
8. Smart Contracts - https://en.wikipedia.org/wiki/Smart_contract
9. Bitcoin Script - https://en.bitcoin.it/wiki/Script
10. Contract - https://en.wikipedia.org/wiki/Contract
11. Hash Function - https://en.wikipedia.org/wiki/Hash_function
12. Cryptography - https://en.wikipedia.org/wiki/Cryptography
13. Algorithm - https://en.wikipedia.org/wiki/Algorithm
14. Bit array - https://en.wikipedia.org/wiki/Bit_array
15. Map - https://en.wikipedia.org/wiki/Map_(mathematics)
16. One-way function - https://en.wikipedia.org/wiki/One-way_function
17. Brute-force search - https://en.wikipedia.org/wiki/Brute-force_search
18. Rainbow table - https://en.wikipedia.org/wiki/Rainbow_table
19. Transaction - https://en.bitcoin.it/wiki/Transaction