Portal: DeFi on Bitcoin

Duggirala,Chandra Martindale, Eric

June 2020
v0.0.4

Contents

Contents **1**
Abstract . . . 1
Introduction . . . 2
Centralized middlemen to DeFi . . . 2
Fabric: A Decentralized Computation Market for Executing Financial Primitives . 3
A Coin Swap as a "fee for computation" . . . 4
Setup . . . 4
Alice finds a Peer . . . 5
Key Insight . . . 5
Alice deposits Bitcoin into the Network . . . 6
Bob deposits Altcoin . . . 7
Alice places an Order . . . 8
Bob places an Order . . . 8
Charlie places an Order . . . 9
Atomic Swap: The Fundamental Primitive of DeFi . . . 10
Layer 1 Atomic Swaps . . . 10
The fault in our swaps . . . 10
Layer 2 swaps . . . 12
Limitations: . . . 13
Properties of Portal . . . 13
A Trust-minimized, Atomic Swap Protocol . . . 14
Case 1: Setup . . . 15
Outcome branches . . . 18
Analysis . . . 19
Fairness in Optionality: . . . 19
Facilitation: . . . 19
Trust-Minimization: . . . 20
Analysis of Usability . . . 20
Drawbacks . . . 21

Case 2: Initiator does not have assets on counterparty's blockchain. 21
Construction (# 1): 22
Outcome branches 24
Analysis (Pass/Fail) 24
Fairness: Pass 24
Facilitation: Pass 24
Trust-Minimization: Fail 25
Analysis of Usability 25
Drawbacks 26
Alternate Construction (# 2): 26
Outcome branches 27
Analysis 28
Fairness: Fail 28
Facilitation: Pass 28
Trust-Minimization: Pass 28
Analysis of Usability 28
Key Insight 29
Censorship resistance 29
Composing More Complex Financial Contracts using Atomic Swaps 30
Interest Bearing Instruments: Lending & Borrowing 30
Derivatives: 31
Conclusion: 31
Code Resources 31
Appendix A 32
Flaws inherent to Decentralized Exchanges (DEXs) 32

Abstract

Here we unveil a multi-layered system purpose-built for allowing censorship-resistant financial (and non-financial) applications on top of the Bitcoin Network. This system allows for the secure establishment and execution of peer-to-peer agreements, including financial and non-financial contracts. Financial applications that can use this infrastructure include spot trading, lending, borrowing, investing, crowdfunding and cryptocurrency derivatives. Other decentralized and censorship-resistant applications such as communication networks, social media networks and others can be built as well on the same infrastructure. This system allows users to access all decentralized services from a non-custodial, user-controlled application or a wallet. Users can trade and contract with the speed, user experience, and liquidity of

centralized alternatives, but without relinquishing control of assets or data to a trusted party in the middle.

The first part of this white paper gives an overview of the fabric peer-to-peer network. The rest of the paper describes our unique construction of a peer-to-peer contract called an "atomic swap" and how it can be used for building peer-to-peer exchanges that have the speed, usability and liquidity of centralized alternatives. By solving longstanding problems with cross-chain atomic swaps, we describe how a trust minimized exchange can be built on top of fabric.

We also describe how our approach fixes the well-known problems associated with current Layer 1 and Layer 2 swaps. Moreover, it describes how liquidity in all the decentralized financial applications can be aggregated across a network of traders by homogenizing and matching orders across trades, using a zero-knowledge system with proofs for order book execution. In addition, we show how our construction introduces partial order execution and composability to cross-chain atomic swaps, how they can be composed into arbitrarily complex contracts for lending, borrowing, derivatives and other financial primitives.

Finally, we describe how the concepts herein can be generalized to build decentralized, censorship-resistant applications, thus allowing fully-featured, trust-minimized web scale applications built on Bitcoin.

Introduction

Decentralized finance is poised to grow rapidly, while at the same time is getting highly fragmented. Many Centralized providers of "pseudo-decentralized" financial services are at risk of getting hacked, shut down by legal and extralegal threats and expose sensitive, private user information to hackers and other malicious third parties. The path to DeFi adoption runs through a unified, easy to use and secure protocol that integrates with all the different DeFi services, but does so without incorporating any new security vulnerabilities or compromising the decentralization, censorship resistance or pseudonymity of the underlying chains and protocols.

Centralized middlemen to DeFi

Centralized financial services built on top of blockchains ("pseudo-decentralized Finance" represents the following threats:

1) **Security:** They act as single "economic nodes" and therefore are central points of failure and negate the decentralization of underlying blockchains.

2) **Censorship**: They can and do censor transactions and blacklist and whitelist addresses, nullifying censorship resistance provided by blockchains,

3) **Privacy**: They can lose sensitive client data as well as funds[1].

Despite these flaws, most DeFi protocols have seen a broad interest from the crypto community. Centralized exchanges that offer various services such as lending, borrowing and derivatives have a near complete market share of crypto (as of this writing) because of their advantages in speed, liquidity, and user experience over decentralized alternatives.

Portal fixes this.

[1] https://cointelegraph.com/news/crypto-exchange-hacks-in-review-proactive-steps-and-expert-advice

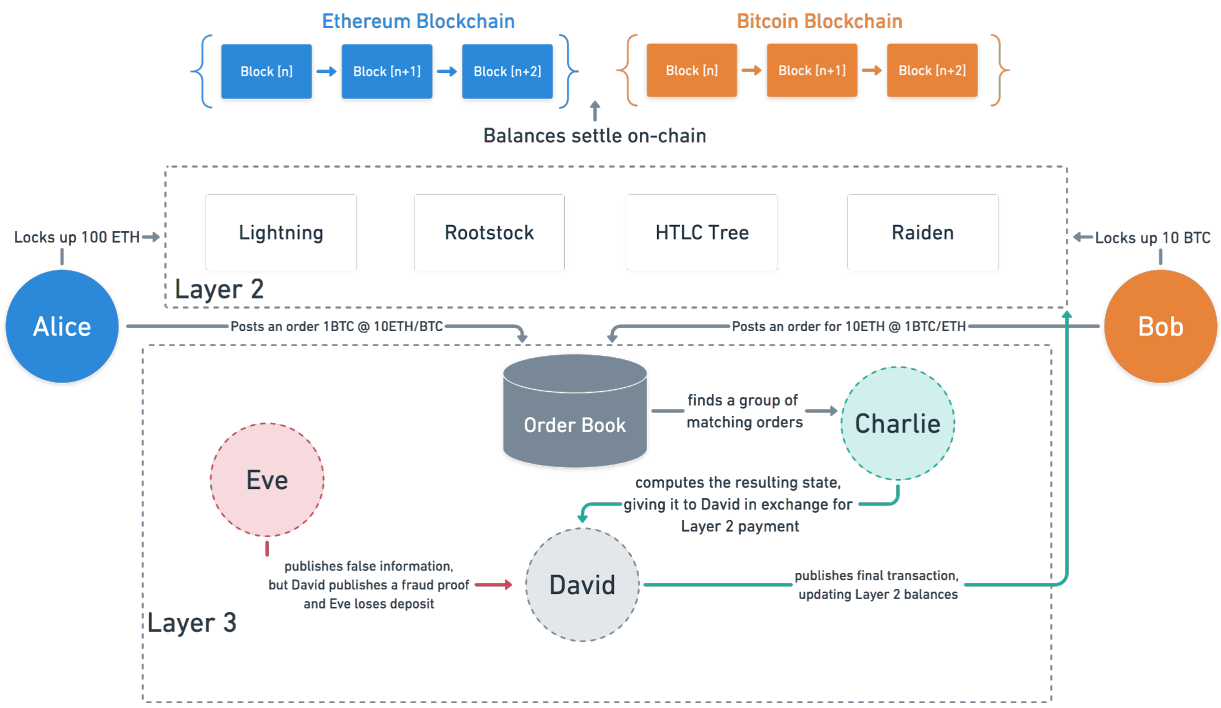

Figure 1: Schematic of Fabric/Portal components

Fabric: A Decentralized Computation Market for Executing Financial Primitives

Fabric is a Layer 3 peer-to-peer market for computation. The system implements ephemeral compute infrastructure using Smart Contracts on different blockchains (i.e. Bitcoin and Bitcoin-like chains like Zcash, Litecoin, etc, and Ethereum and other smart contract-based chains). A user can compose a secure multi-party circuit over which they can compute some program, typically written in a higher-order language or visually composed with an editor. This is based on encrypted information supplied to the program as part of a *homomorphically-encrypted cryptosystem.*

1) Alice generates some prover program P
2) Alice makes available some amount A of digital currency C

3) Alice publishes a proof that a class of solutions exists in which outcome X results from some input space I

4) Alice pushes some homomorphically-encrypted data D, signing over I

5) Peers compute solutions, unlocking funds from A

In the case of Portal, the Prover programs are swap requests built in the form of HTLCs. If Alice wants to exchange her Bitcoin for Ethereum, she broadcasts her offer to the market for correct execution of her designated circuit by creating and broadcasting the swap request. Spot trading, options, lending and borrowing, derivatives become composable as "programs for fee" . Contract execution, sorting of the order book, proving that the swaps are executed based on the rules set forth in the facilitation engine, real-time pricing of options, interest, and deposits, all become prover programs. The agents that provide proofs of correct execution can unlock funds from the HTLCs.

A Coin Swap as a "fee for computation"

As an example, we illustrate how a simple "coin swap" works in the peer-to-peer computation marketplace.

Setup

Alice bonds her Bitcoin into an HTLC hash, refundable to her after 90 days (144×90 blocks on the Bitcoin blockchain).

Alice finds a Peer

Fund recovery: At this point, Alice's risk is a total loss of funds, but this is a worst-case scenario and only if she loses her keys. In all other cases, Alice can fully recover her funds after 90 days.

```
flags:
  - transaction: 1 BTC from Alice, 1 BTC from ANYONECANPAY
  - inputs:
    - 1 BTC from Alice
    - 1 BTC from ???
  - output: 2 BTC
  - sighash: SINGLE|ANYONECANPAY
---
OP_DUP # duplicate element on stack (peer's signature)
12960 # 90 days in Bitcoin Time
OP_CHECKSEQUENCEVERIFY OP_IF
  <pubKeyAlice> OP_CHECKSIG # spendable by Alice
OP_ELSE
  <pubKeyAlice> 2 OP_CHECKMULTISIG # peer's signature already on stack
OP_ENDIF
```

Key Insight

Once Alice finds a peer, further contract executions require her signature to continue. By composing long chains of unsigned transactions, Alice can selectively unlock portions of her funds for solutions to "puzzles" in the larger transaction graph. Expanding on this technique, we can compute long-running states, even cycles, over some ephemeral (yet mutual) state.

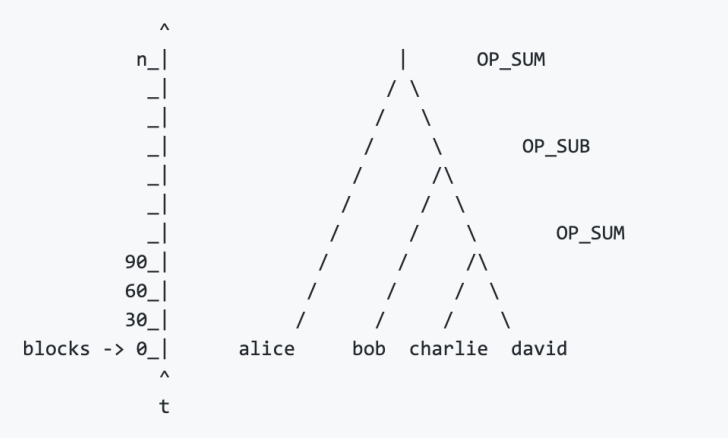

Figure 2: A simple machine

In the Portal system, we call these *threads*. Let's explore how such a thread is created.

Alice deposits Bitcoin into the Network

Alice now wants to unlock her Bitcoin for use in her network, so she generates some secret S and generates the sha256 sum (*preimage hash)*.

```
flags:
  - transaction: 1 BTC from Alice, spendable by her peer with secret _S_
---
OP_IF
  OP_SHA256
  <preimage hash>
  OP_EQUALVERIFY
  <pubKeyBob>
  OP_CHECKSIG
OP_ELSE
  12960 # 90 days in Bitcoin Time
  OP_CHECKSEQUENCEVERIFY
  OP_DROP
  <pubKeyAlice>
  OP_CHECKSIG
OP_ENDIF
```

Alice does not broadcast this transaction, but signs it and passes it off to Bob for his safekeeping. Bob, the partner in Alice's transaction, can only claim the funds in this channel when he also knows the secret, either by observing the network or by learning it from Alice directly. Alice maintains at least one of these channels per peer, by nature of construction. In our reference implementation, we designate 32 as the target network size, but in practice, networks will be larger.

Bob deposits Altcoin

Bob, also a member of the network, is able to compose a transaction which is spendable to Alice, but she'll have to reveal S to do so. Alice can hand over the secret directly to Bob to ensure her outstanding channels remain open, or simply broadcast the transaction on-chain to begin the settlement process. Now, given that the facilitator has a fee included for himself, the order shown below can continue to get filled, until the inputs equal the outputs, at which time, it can be broadcast and be settled on layer 1. Notice that the composability here comes from using special transaction flags Sighash_Single/Sighash_Anyone_can_pay or Sighash_All/Sighash_Anyone_can_pay.

```
flags:
  - transaction: 1 ALT from Bob, spendable by Alice with secret _S_
---
OP_IF
  OP_SHA256
  <preimage hash>
  OP_EQUALVERIFY
  <pubKeyAlice>
  OP_CHECKSIG
OP_ELSE
  12960 # 90 days in Bitcoin Time
  OP_CHECKSEQUENCEVERIFY
  OP_DROP
  <pubKeyAlice>
  OP_CHECKSIG
OP_ENDIF
```

Alice places an Order

```
flags:
  - SIGHASH_SINGLE
  - SIGHASH_ANYONECANPAY
---
OP_IF
  OP_SHA256
```

```
  <preimage hash>
  OP_EQUALVERIFY
  <pubKey of swap>
  OP_CHECKSIG
OP_ELSE
  <relative locktime>
  OP_CHECKSEQUENCEVERIFY
  OP_DROP
  <pubKey of refund>
  OP_CHECKSIG
OP_ENDIF
```

Bob places an Order

```
flags:
  - SIGHASH_SINGLE
  - SIGHASH_ANYONECANPAY
---
OP_IF
  OP_SHA256
  <preimage hash>
  OP_EQUALVERIFY
  <pubKey of swap>
  OP_CHECKSIG
OP_ELSE
  <relative locktime>
  OP_CHECKSEQUENCEVERIFY
  OP_DROP
  <pubKey of refund>
  OP_CHECKSIG
OP_ENDIF
```

Charlie places an Order

```
flags:
  - SIGHASH_SINGLE
  - SIGHASH_ANYONECANPAY
---
OP_IF
  OP_SHA256
  <preimage hash>
  OP_EQUALVERIFY
  <pubKey of swap>
  OP_CHECKSIG
OP_ELSE
  <relative locktime>
  OP_CHECKSEQUENCEVERIFY
  OP_DROP
  <pubKey of refund>
  OP_CHECKSIG
OP_ENDIF
```

This system makes private transactions possible for any blockchain. Given that the private key Alice uses to sign I is only held by Alice, the funds sent to the contract can only be controlled by either Alice or the party that "computes" the solution that unlocks funds from A, a decentralized marketplace for financial transactions exists, if the following conditions are met:

1) Facilitation (i.e, aggregation of supply and demand) is incentivized, and

2) Identity of the transactors is hidden from the facilitator

Both these properties can be guaranteed by fixing problems in layer 1 (Tier-Nolan) swap.

Atomic Swap: The Fundamental Primitive of DeFi

A swap contract is a two party trade between assets belonging to two different blockchains. and constitutes the fundamental unit of trade between two parties. Unfortunately, until now,

the problem of "swapping" coins belonging to two different blockchains between untrusted parties remained unsolved.

Layer 1 Atomic Swaps

"Atomic[2] swaps" , initially thought of as a solution to the problem of trust-minimized cross-chain exchange, have not advanced enough to provide a viable alternative to centralized exchanges.

The fault in our swaps

Atomic swaps were first proposed as a solution to the "exchange risk" in 2013[3]. The classic Tier Nolan Atomic swap uses Hash Time Locked Contracts (HTLCs) and is well understood to have the following problems:

1) **Facilitation**: The classic Tier Nolan atomic swap does not have incentives built into the protocol itself for the swaps to be facilitated by any third parties

2) **The "Inadvertent Call Option"** [4]: In a trade, the party holding the preimage gets an option but is not obligated to buy the counterparty's coins for a fixed exchange rate before time lock expiration. This is called an "*Inadvertent American call option*" . If the price moves against the trade, it can be aborted anytime.

3) **Liquidity Trolling/Lockup Griefing**: The party that acts second (i.e, the one without the preimage) can make the party with the preimage lock up liquidity for significant periods of time with no intention of following through.

[2]An atomic transaction is a series of operations that is indivisible and irreducible. It means that either the entire series of operations happen or none of them happen.

[3]https://bitcointalk.org/index.php?topic=193281.msg2224949# msg2224949

[4]https://blog.bitmex.com/atomic-swaps-and-distributed-exchanges-the-inadvertent-call-option/

4) **Speed**: On Bitcoin and other chains, lock times of HTLCs have to be long enough because the security comes from the time difference between the holder of the preimage and counterparty locking funds in HTLCs and the block production times, since confirmation cycles vary around the mean block time. This lag makes them unsuitable for spot exchange.

5) **Coordination Costs**: Swaps can fail after agreeing on exchange rate (wasted negotiation) because agreements prior to both parties' commitments are non-binding. This discourages parties from negotiating.

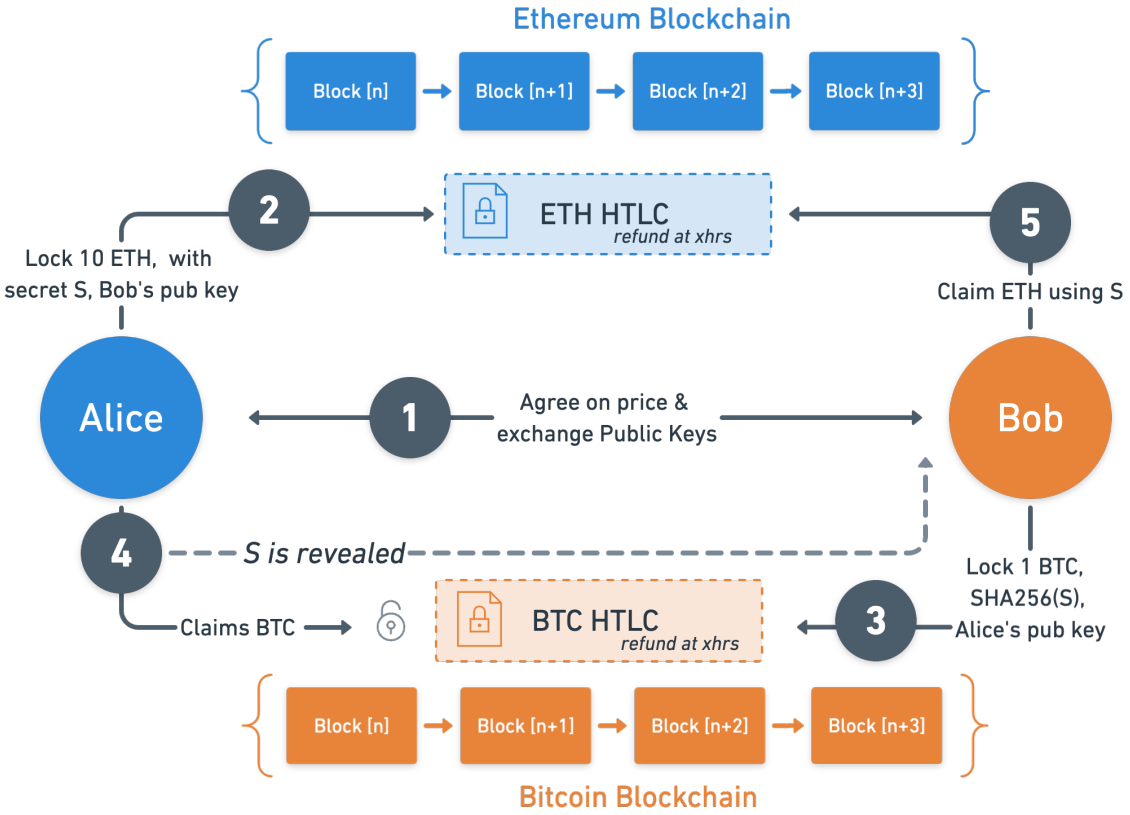

Figure 3: A simple Tier-Nolan atomic swap

Layer 2 swaps

As of this writing, only one Layer 2 swap protocol exists[5]. Arwen was designed for trading between a centralized exchange and its users. Note that in this case, the exchange does not play the role of an "exchange" but that of a trading desk or OTC desk, counterparty to its users. In Arwen, both the user and the exchange lock up their coins in on-chain escrow smart contracts, and "pair" their escrows by exchanging their respective Public Key Hash (PKH). Henceforth, all trades involve off-chain passing of signed (but not posted) transactions. These are between the two parties that update balances from the escrows between the centralized exchange and the user until the escrows are closed. Transactions are therefore fast and happen at near-zero cost.

Limitations:

1) **Dependence on identity:** Arwen protocol depends on the real-world reputation of centralized exchange companies to circumvent the "inadvertent call option" problem. They are expected to act in a manner that protects their reputation and not strategically cancel trades. But it is not a guarantee enforced by the protocol. The user needs to trust that the exchange cares about its reputation enough to voluntarily forego the profits from the free options it owns. The assumption that "reputational risk" is enough of a deterrent from strategic manipulations of orders and order books is not borne by evidence. Despite reputational costs of manipulating order books, many of the world's top exchanges continue the practice[6]. Arwen protocols have no rational economic incentives to mitigate this.

2) **Not Peerable:** As the authors admit in their whitepaper, Arwen is purpose-built for users to trade *against* centralized exchanges, not with other peers *at* centralized exchanges. An anonymous, peer-to-peer cross-chain exchange is not implementable using Arwen.

3) **Incompatibility with "exchange" business model:** Arwen protocols require centralized exchanges to become trading desks and market makers, revamping their existing business models.

[5]https://arwen.io/whitepaper.pdf
[6]https://www.sec.gov/comments/sr-nysearca-2019-01/srnysearca201901-5164833-183434.pdf

Properties of Portal

The Portal system incentivizes exchanges to facilitate swaps, while being anonymous to the user.

The technical goals of this swap protocol are to ensure the following properties:

1) **Fairness ("Priced Optionality")**: The party with the option pays a fair market price for that option (henceforth "reclaimable deposit").

2) **Facilitation**: Neutral, anonymous third parties ("Facilitators") are economically incentivized to coordinate users who don't know each other and facilitate swaps by providing resources.
(Resources include: user onboarding, order aggregation, fair pricing service for the option/reclaimable deposit, and services on-par with centralized exchanges like execution speed, liquidity & user experience)

3) **Trust-minimization**: Collusion between any of the trading parties (Party, Counterparty and Facilitator) does not allow theft of any party's coins.

We show that given the constraints of Bitcoin opcodes (as of the day of this writing), all three of the above can be achieved if the swap participants have:

1) A minimum amount of both the assets involved in the trade, and

2) Key pool files on blockchains of both the assets

In the absence of either of the above, we show that we can still achieve Facilitation + Fairness, or Facilitation + Trust-minimization, but not all three.

A Trust-minimized, Atomic Swap Protocol

These are the building blocks for our protocol are HTLCs on Bitcoin (and other UTXO chains) and smart contracts on Ethereum (and other smart contract blockchains).

Case 1: Setup

Alice and Bob want to trade. Alice wants to sell her BTC for LTC, and Bob wants to sell his LTC for BTC. Alice has both BTC and LTC in her Wallet. Bob has LTC in his wallet, but his Bitcoin wallet is empty. The facilitator relays messages between both users using a mailbox.

1) Alice gets the "price feed" from either the facilitator or a third party. She decides to swap 10 BTC for 100 LTC.

2) Alice initiates the swap by sending a request for swap to the facilitator. This communicates to the facilitator her order size/trading pair and requests the deposit price.

3) Facilitator responds with:

   a) a reclaimable deposit price,

   b) Its fee, and

   c) its Public Key Hashes on both BTC and LTC. The facilitator takes into account the size of order, the price of the asset and the volatility of the asset, using an options pricing method such as binomial options pricing[7] or other methods. The deposit is quoted in the counterparty currency. For the sake of this example, we assume that the deposit is "5 LTC" and the fee for a successful swap is "0.1 BTC + 1 LTC" .

4) IF Alice has $>$ 5 LTC in her wallet, she goes to step 5; ELSE she goes to Case 2.

[7] http://static.stevereads.com/papers_to_read/option_pricing_a_simplified_approach.pdf

5) Alice generates a random preimage ("secret"). Alice then constructs the following two partial transactions: one bitcoin transaction and one litecoin transaction (a & b below), then sends 3 pieces of data to the exchange:

   a) A partial HTLC on Bitcoin chain: Input is 10 BTC from Alice, left blank. Time lock is set to 2 days. Output is locked at 10 BTC + 0.1 BTC (left blank for Bob to fill).

      (i) HTLC conditions: Before 2 days, blank can sign if he reveals Alice's secret. In this case, a fee is paid to the facilitator's address upon a successful swap.

      (ii) Alice can refund herself her 10 BTC (minus transaction fee). Facilitator gets zero fee.

```
# Alice's Bitcoin HTLC, refund branch pays Alice
OP_IF
    OP_SHA256 <hash_of_secret>
    OP_EQUALVERIFY OP_DUP OP_HASH160 <scriptPubKey(Pay_To_Script_Hash)>
OP_ELSE
    timeout OP_CLTV OP_DROP OP_DUP OP_HASH160 Alice_pubkey
OP_ENDIF
OP_EQUALVERIFY
OP_CHECKSIG

#Successful Swap P2SH, pays to Bob, Facilitator
scriptPubKey: OP_HASH160 <redeemScriptHash> OP_EQUAL
scriptSig: <signatures> <public_Keys Bob, Facilitator> <redeemScript>
```

   b) A partial HTLC on Litecoin chain (the deposit transaction): This takes as input, 5 LTC from Alice, Output locked at 105 LTC. 100 LTC input left blank. HTLC conditions:

      (i) Before 1 day, Alice can claim LTC by revealing her secret. In this case, the facilitator gets 1 LTC (the fee) andAlice gets 104 LTC (this essentially reclaims Alice's deposit minus the facilitator fee).

      (ii) After 1 day, Bob gets 105 LTC. Facilitator gets nothing.

```
#Alice fails to complete swap, pays Bob (refund + deposit)
OP_IF
    OP_SHA256 <hash_of_secret>
    OP_EQUALVERIFY OP_DUP OP_HASH160 <Pay_To_Script_Hash>
```

```
OP_ELSE
    timeout OP_CLTV OP_DROP OP_DUP OP_HASH160 Bob_pubkey
OP_ENDIF
OP_EQUALVERIFY
OP_CHECKSIG

#Successful Swap P2SH, pays to Alice, Facilitator
scriptPubKey: OP_HASH160 <redeemScriptHash> OP_EQUAL
scriptSig: <signatures> <public_Keys Alice, Facilitator> <redeemScript>
```

c) Hash of her secret

Note: Notice that if Alice does not go through with the swap, Alice loses her deposit and Bob gets it. If she does, within the time lock, she can reclaim her deposit. This is what makes it a "reclaimable deposit" .

6) Facilitator checks the transactions, verifies PKHs were included in both the HTLCs in the success swap branches, and the amounts match the message previously sent in 3. Then Facilitator relays the swap request to all its users.

7) Bob sees Alice's request on the "order book" that is visible in his wallet/client, and accepts the price. To accept, Bob does the following:

   a) Verifies that the hash of secret is included in the Bitcoin HTLC.
   b) Adds his PKH to Alice's BTC HTLC, adds his input of 0.01 BTC (to pay facilitator fee), and signs.
   c) Adds his 100 LTC input to the Litecoin HTLC, locked under the hash of the same secret Alice sent, adds his PKH to the HTLC, signs, and sends it back to the facilitator.

8) Facilitator then checks the HTLCs again to make sure that its PKHs, amounts, etc are unchanged, and then relays both the partial HTLCs to Alice.

Note: Notice that until this time, all the messages passed between Alice, Facilitator and Bob are off-chain and are instantaneous.

9) Alice checks the partial HTLC, verifies that her PKH is included in the correct branch, makes sure that the hash of the secret Bob included in his Litecoin HTLC matches her Bitcoin HTLC. If it is verified, she adds her 10 BTC input, signs it and then broadcasts it to the Bitcoin chain.

   Note: This takes one on-chain confirmation cycle equivalent to current "transfer to centralized exchange" delays. But unlike CEXs, there are no further delays as they would arbitrarily enforce due to a) CEX fractional reserve business models, and b) transaction aggregation to minimize on-chain transaction fees.

10) Bob can now add his 100 LTC input to his partial Litecoin HTLC, which then makes it a valid transaction (the inputs equal or exceed the outputs). If he doesn't, he pays the facilitation fee to Alice instead of Facilitator. This is Bob's "deposit" that prevents him from lockup griefing Alice.

11) Alice waits for one Litecoin confirmation cycle and then claims her 4 LTC deposit plus Bob's 100 LTC (total of 104). 1 LTC goes to Facilitator.

12) Bob then claims Alice's BTC on the Bitcoin chain. 0.1 BTC goes to Facilitator.

The swap is successfully executed.

Note: Notice that the total cycle is 2 confirmations. This is, at most, equivalent to transferring an asset to a centralized facilitator and withdrawing after the trade.

Outcome branches

1) **Alice's Cancellation**: After requesting a swap, Alice can cancel anytime before Bob commits by adding his 100 LTC and broadcasts to the chain.

   a) If Alice cancels her order and spends her LTC, when Bob tries to "execute" , the Litecoin chain rejects one of the inputs to his HTLC.

b) Alice can then claim her BTC back after 2 days.

2) **Success**:

a) Alice claims her LTC, enabling Bob to claim her BTC. She can claim Bob's LTC + her premium of 5 LTC (minus facilitator fee) anytime before 1 day, by revealing her secret X on LTC chain. Facilitator gets paid a fee.

b) Bob can claim his BTC: Anytime after Alice claims her LTC, but before 2 days.

3) **Failure**: If Alice does not reveal secret X by T_2, Bob can get 105 LTC (his 100 LTC plus Alice's option premium). Alice gets her BTC + facilitator fee after T_1. Facilitator does not get a fee. Alice loses her premium.

Analysis

Fairness in Optionality:

There are two ways to mitigate the unfairness of an inadvertent call option inherent in atomic swaps:

1) Eliminate the unintended option, or

2) Internalize the cost of an option so that the option is "priced in" with the cost of the swap.

Eliminating the inadvertent option on Layer 1 is not possible, as separate blockchains do not communicate with each other. Instead, Portal internalizes the price of the option in the form a "reclaimable deposit/bond" .

Here, the cost of the option is included in Alice's Bitcoin HTLC. This internalizes the cost and eliminates inefficiency inherent to Tier Nolan atomic swaps.

Facilitation:

1) **Fairly pricing the** "**reclaimable deposit**" : The burden of accurately pricing the option is taken off Alice because the swap facilitator is incentivized to appropriately price the option aligned with the market. If the option price is too high, Alice won't buy, and the facilitator earns nothing. If it is too low, Bob won't sell, and the facilitator earns nothing. Only a successful swap pays the facilitator.

2) **Coordination**: Facilitator is incentivized to:
   a) connect Alice with Bob using a relay service,
   b) maintain and aggregate swap requests ("order book"),
   c) provide a good user experience, and
   d) make sure that all messages are relayed between them.

   Note: If the above protocol halts at any stage of the success branch, the facilitator does not get paid.

Trust-Minimization:

Neither Alice colluding with Bob nor Alice and/or Bob colluding with the facilitator risks losing the any party's coins, if the third party follows the protocol fully.

None of the elements of the swap require a trusted third party to control coins of any participant. Alice controls her coins until the swap request is accepted, Bob controls his coins until he accepts the swap, and Facilitator gets his coins successful execution. Alice broadcasts her own Bitcoin HTLC to Bitcoin chain, Bob broadcasts his Litecoin HTLC to Litecoin chain.

Analysis of Usability

1) Alice's "request" for a swap requires her to lock her Bitcoins in an HTLC on-chain. This requires no more time or transaction fees than transferring BTC to a centralized

exchange before trading.

2) Alice's request, Bob's PKH relay, and Facilitator's PKH relay are all off-chain communications at Layer 2 and therefore happen fast.

3) Facilitator maintains and displays all swap requests on its order book. All of them are partially-constructed HTLCs and have no need to be broadcast to their chains yet. This makes the experience as fast and usable as CEXs.

4) Cancellation option: Alice can cancel her order anytime before Bob accepts by simply sending her input into Litecoin HTLC elsewhere or back to a change address that she owns. The possibility of a race condition is resolved by the LTC blockchain. This requires no more steps than pressing a "cancel" button.

5) To make sure that no party abandons its activity in the middle of a swap (either by going offline or for other reasons) all steps except 1 & 2 can be automated on the client side (exchange app or directly within a noncustodial wallet app). This makes the user experience seamless and comparable – or superior – to centralized exchanges.

Drawbacks

1) Here, the initiating party needs to already have the asset being purchased and access to a true multi-currency wallet that supports both BTC and LTC.

2) Both parties need to be online during the swap, or have access to a client that follows the protocol without fail from beginning to end.

   Note: *Portal* is a multi-currency wallet that performs both of these functions.

Case 2: Initiator does not have assets on counterparty's blockchain.

In this case, we have three possibilities.

1) The swap initiator purchases the asset they need to deposit, using one of the following pathways Alternate construction # 2 or Alternate construction # 3.

2) Alternate construction # 2 minimizes transaction fees and wait time by compressing the initial transaction into one of the HTLCs, but sacrifices trust-minimization to gain usability with lower transaction fees.

Alternate construction # 3 sacrifices fairness in favor of trust-minimization. It makes both transactions trustless, but requires two transaction fees and two wait times. The first transaction in this construction is a Layer 1 Tier Nolan atomic swap, but problems are limited in scale only to the cost of the second transaction's reclaimable deposit.

Construction (# 1):

1) Alice wants to swap her 10 BTC for 100 LTC. She sends a message to the facilitator with this request.

2) She sets a price of 100 LTC / 10 BTC.

3) Facilitator prices the reclaimable deposit of 0.5 BTC based on the volatility, size, order timing, etc.

4) Alice constructs the following transaction and sends to the facilitator, the following:

    a) 10 BTC from Alice, signed
    b) 0.5 BTC from Alice, signed
    c) Locked under secret for time T_1
    d) Redeemable by Blank for 10 BTC
    e) 0.5 BTC to Facilitator (output locked)
    f) After time T_1 Alice gets 10 BTC

g) Facilitator gets 0.5 BTC

5) Alice also constructs the following LTC transaction:

   a) Input of 5 LTC blank (to be filled by the facilitator)
   b) 100 LTC input left blank (needs to be added by Bob and sign, if he accepts)
   c) Output 105 LTC
   d) Locked under the following conditions:
      (i) Before time T_2 Alice can spend funds using the secret X (105 LTC), 104 go to Alice and 1 LTC goes to Facilitator, or
      (ii) After T_2 Alice can claim all 105 LTC.

      Assumption: Facilitator will broadcast the partially constructed LTC order by adding a valid input to the transaction from its own wallet.

6) Facilitator adds its input of 5 LTC to Alice's LTC transaction.

7) Facilitator "relays" or posts this order on its order book, and also sends it back to Alice.

8) Bob, who wants to sell his LTC, "accepts" by relaying his PKH to Alice & the facilitator.

9) Facilitator then waits for confirmation of Alice's BTC transaction from # 4, filled with Bob's PKH, and posts on Bitcoin chain.

10) Facilitator then adds its 5 LTC to Alice's LTC transaction, and sends to Bob.

11) Bob then adds his input to Alice's Litecoin HTLC and posts it to Litecoin chain.

12) Alice can then either:

   a) claim her LTC, paying the facilitator fee, or

b) Bob will claim his LTC plus Alice's option premium, paying zero fee to the facilitator.

13) If Alice does not claim her LTC, in the refund path the facilitator receives .5 BTC to compensate for the LTC it sent for Alice's deposit.

Outcome branches

1) After requesting a swap, Alice cannot cancel the swap without facilitator's cooperation before Bob commits. Alice can send a "Cancel" request to the facilitator to have the facilitator send the input elsewhere.

2) **Cancellation**: If Alice/Facilitator cancels the order and spends the LTC, when Bob tries to "execute" , the Litecoin chain rejects one of the inputs to his HTLC. Alice can then claim back her BTC after T_1. The facilitator gets paid a fee.

3) **Success**:

   a) Alice claims Bob's LTC, enabling Bob to claim her BTC. She can claim Bob's LTC + her premium of 5 LTC anytime before T_2 by revealing the secret X on Litecoin chain. Facilitator gets paid a fee.

   b) Bob can claim Alice's BTC anytime after Alice claims Bob's LTC, but before T_1.

4) **Failure**: If Alice does not reveal X by T_2 Bob can get 105 LTC (his 100 LTC plus Alice's option premium). Alice gets her 10 BTC back, but has to pay the facilitator 0.5 BTC.

Analysis (Pass/Fail)

Fairness: Pass The swap is fair to both Alice and Bob. Neither one gets an inadvertent call option. This is similar to Construction # 1.

Facilitation: Pass

1) **Option pricing**: Facilitator is incentivized to price the option appropriately because pricing the option too high deters Alice from initiating the swap request, yet pricing it too low deters Bob from accepting the swap.

2) **Coordination**: Facilitator is incentivized to connect Alice with Bob and to make sure that all messages are relayed between them. If the above protocol halts at any stage of the success branch, the facilitator does not get paid. Only the success branch results in a fee for the facilitator.

Trust-Minimization: Fail Alice trusts that the facilitator will take her BTC and input its LTC into the partially-constructed LTC transaction, and relayed to Bob. Alice also trusts that the facilitator will cancel the order if Alice requests so before acceptance of the swap.

Analysis of Usability

1) Alice's "request" for a swap requires her to have her BTC (deposit BTC as well as swap BTC) locked in an HTLC on chain. This requires no more time or transaction fees than transferring BTC to a centralized exchange before trading.

2) Alice's request, Bob's PKH relay, and Facilitator's PKH relay are all off-chain communications and therefore happen fast.

3) Facilitator maintains and displays a cache of partially-constructed LTC transactions on the facilitator's server. Because all transactions are off-chain until a swap request is accepted, it is entirely off-chain, and is therefore as fast and as usable as CEXs.

4) Cancellation option: Alice can cancel her order anytime before Bob accepts by simply sending a request to Facilitator, which then spends the input elsewhere or back to a change address that she owns. The possibility of a race condition is resolved by the

Litecoin blockchain. This requires no more steps than pressing a "cancel" button.

5) To make sure that no party to the swap abandons its activity in the middle of a swap, either by going offline or for other reasons, many of the essential actions can be automated on the client side.

Drawbacks The drawbacks are obviously a sacrifice in trust-minimization.

Alternate Construction (# 2):

1) Alice wants to swap her 10 BTC for 100 LTC. She sends a message to the facilitator with this request.

2) She sets a price of 100 LTC / 10 BTC.

3) Facilitator prices the reclaimable deposit of 0.5 BTC (5 LTC) based on the volatility, size, order timing etc.

4) Alice performs a Tier Nolan swap with Facilitator for 0.5 BTC / 5 LTC.

5) Alice then constructs the following partial LTC transaction and sends to the facilitator:
   a) Input of 5 LTC from Alice, signed
   b) 100 LTC input left blank (to be added and signed by Bob if he accepts)
   c) Output of 105 LTC
   d) Locked under the following conditions:
      (i) Before time T_2 Alice can spend funds using the secret X (105 LTC), where 104 LTC goes to Alice and 1 LTC goes to Facilitator, or
      (ii) after T_2 Alice can spend all 105 LTC.

6) Facilitator "relays" or posts this order on its order book.

7) Bob, who wants to sell his LTC, "accepts" by relaying his PKH to Alice through the facilitator.

8) Alice locks 10 BTC in an HTLC on-chain, using secret X, with the following conditions:
   a) Before T_1 Bob can spend with X, or
   b) After T_1 Alice gets to spend.

9) Alice's transaction is confirmed.

10) Bob then adds his input to Alice's Litecoin HTLC and posts it to Litecoin chain.

11) Alice can then either:
   a) Claim her LTC, paying the facilitator fee, or
   b) Bob will claim his LTC plus Alice's option premium, paying zero fee to the facilitator.

   The branches of the above are the same as construction # 1.

Outcome branches

1) After requesting a swap, Alice cannot cancel the swap without facilitator's cooperation before Bob commits. Alice can send a "Cancel" request to the facilitator to have the facilitator spend the input elsewhere.

2) **Cancellation**: If Alice/Facilitator cancels her order and spends her LTC, when Bob tries to "execute" , the Litecoin chain will reject one of the inputs to his HTLC. Alice can then claim her BTC back after T_1. The facilitator gets paid.

3) **Success**:

Alice claims Bob's LTC, enabling Bob to claim her BTC. She can claim Bob's LTC + her premium of 5 LTC anytime before T_2, by revealing the secret X on Litecoin chain. Facilitator gets paid a fee.

Bob can claim Alice's BTC: Anytime after Alice claims Bob's LTC, but before T_1.

Failure: If Alice does not reveal X by T_2 Bob can get 105 LTC (his 100 LTC plus Alice's option premium). Alice gets her 10 BTC back, but has to pay the facilitator 0.5 BTC.

Analysis

Fairness: Fail The swap is not fair to Alice because she is required to give the facilitator a free option through a preliminary transaction, Swap_1, where she must buy LTC as a deposit. Here, we revert to a standard Tier-Nolan swap.

Facilitation: Pass

1) Option pricing: Facilitator is incentivized to price the option appropriately because pricing the option too high deters Alice from initiating the swap request, yet pricing it too low deters Bob from accepting the swap.
2) Coordination: Facilitator is incentivized to connect Alice with Bob and to make sure that all messages are relayed between them. If the above protocol halts at any stage of the success branch, the facilitator does not get paid. Only the success branch results in a fee for the facilitator.

Trust-Minimization: Pass None of the transactions require trust in the counterparty of that transaction. Trust-minimization is preserved.

Analysis of Usability

a) Alice must perform two transactions:

(i) Her preliminary *Swap₁* transaction (to acquire LTC as a deposit for *Swap₂*) takes time and incurs transaction fees, and

(ii) $Swap_2$ which proceeds according to the steps outlined in Construction # 1.

b) Alice's request, Bob's PKH relay, and Facilitator's PKH relay are all off-chain communications and therefore happen fast.

c) Facilitator maintains and displays a cache of partially-constructed LTC transactions on the facilitator server. Because all transactions are off-chain until a swap request is accepted, it is therefore as fast and as usable as CEXs.

d) Cancellation option:

(i) For *Swap₁* Alice cannot cancel once she initiates the swap. Facilitator can choose to participate or not after quoting a price to Alice.

(ii) In *Swap₂* conditions are the same as Construction # 1.

e) To ensure no party abandons its activity in the middle of a swap, either by going offline or for other reasons, many of the essential actions can be automated on the client side.

Key Insight

As we can see above, the limited but powerful functionality of Bitcoin scripts can be used to express complex conditions for spending and therefore construct powerful financial contracts that solve several important problems inherent in cross-chain atomic exchange, such as the free option problem, lockup griefing, fairness etc.

Censorship resistance

Solving these issues allows us to bring the security model of bitcoin mining to cross-chain exchange. A distributed mining model makes Bitcoin censorship-resistant because it preserves the market's ability to supply transaction confirmations for a fee,

without revealing the identity of the miner to the Bitcoin sender and vice versa[8]. With the protocol described above, anyone can, pseudonymously facilitate cross-chain atomic swaps if a facilitator fee is included. If the fee is competitive, swaps are expected to execute. This means that the service of "exchange" of Bitcoin and other cryptocurrencies would no longer be the exclusive domain of centralized exchanges rooted in real-world identity.

However, given both the lack of real identity and the system's peer-to-peer nature (both are desirable), the facilitator circumvents "reputation risk" and can exercise its incentive to manipulate markets, just as centralized exchanges are known to do today. To prevent this, we use powerful zero-knowledge constraints – within a Layer 2 & Layer 3 protocol called Fabric – that limit manipulation of the peer-to-peer atomic swap market by any facilitator.

Composing More Complex Financial Contracts using Atomic Swaps

The atomic swap contract described above is the fundamental primitive used to build arbitrarily complex financial contracts such as interest bearing instruments, derivatives, and others. The Portal protocol is flexible enough to support a wide range of financial derivatives of on-chain assets, a few examples are provided below for illustration.

Interest Bearing Instruments: Lending & Borrowing

Portal's swap contract can, with a slight modification of HTLC tree and time, becomes the primitive for lending and borrowing. For example, in the contract described in section 1, (Case 1, step 3), instead of including the swap fee, the initial contract includes the interest rate, which depends on the time and amount of BTC/ETH being exchanged. The amount of interest received once the loan is paid back is included in the initial partial HTLC, as a long chain of unsigned transactions, which get selectively unlocked, when the loan gets paid back, with the amount of interest received depending on the time of repayment. Because all the funds are "time locked" , repayment at a specific time carries a certain interest, depending on the agreed upon interest rate.

[8]https://github.com/libbitcoin/libbitcoin-system/wiki/Censorship-Resistance-Property

Notice that this has the exact same security model as that described in section 1 swap contract, with no additional dependencies.

This, therefore will provide a true P2P lending experience. It requires no escrow service or an exchange in the middle. The contracts used to lend and borrow themselves have incentives to encourage borrowers to pay back their loans on time, and for lenders to provide these loans knowing that they do not have to trust a third party.

Derivatives:

The "inadvertent call option" problem described in section 1 where in a cross chain atomic swap, the party exercising the option gets it for free, is the exact problem that makes Portal a viable primitive for cross chain options trading. Imagine the layer 1 transaction described in section 1, but with a long enough timelock of days, months or longer, and it suddenly becomes an american call option on the locked asset, at the exchange rate determined in step 3.

Conclusion:

The innovations described herein create access to a decentralized dark pool via a private, fast, trust-minimized network accessible right inside a crypto user's wallet. Any user can now manage multiple blockchain assets, liabilities and financial services from within his wallet.

Moreover, the security of our approach does not depend on off-chain or real-world identities of counterparties, nor of the entity facilitating these services.

With Portal, "decentralized finance" becomes a service that any anonymous entity can provide for a competitive fee within open, transparent markets, with a security model as robust as Bitcoin mining.

Code Resources

https://github.com/Tides-Network/Portal-MacOs https://github.com/Tides-Network/portal-bridge https://github.com/fabriclabs https://dev.fabric.pub/ https://github.com/FabricLabs/fabric/

Appendix A

Flaws inherent to Decentralized Exchanges (DEXs)

DEXs[9] have been touted as a solution[10] to exchange risk, but adoption among DEX users continues to be insignificant. DEXs typically include protocols that are chain-dependent. For example, 0x and Airswap only support ERC-20 tokens, Stellar's DEX only supports Stellar-based tokens, and Newdex and DEXEOS supports EOS-based tokens only.

Well-known DEX problems include the latency inherent in submitting or cancelling orders on-chain and transaction fees involved in confirming orders or cancellations when order books are hosted on-chain. Here, slow execution, bad user experience and illiquidity are unavoidable.

Some exchanges, such as 0x, AirSwap, EtherDelta, and IDEX, employ off-chain order books but are still plagued with latency and user experience problems. More concerning are their organizational centralization and systemic risks[11]. In addition, DEXs expose order information to miners, who may then be able to strategically front-run orders and cancellations. This presents issues of privacy, which are compounded by users' concerns about exchanges knowing their real-world identities[12].

[9] https://stanford-jblp.pubpub.org/pub/deconstructing-dex

[10] https://medium.com/@FEhrsam/why-decentralized-exchange-protocols-matter-58fb5e08b320.

[11] https://medium.com/@peckshield/0x-exchange-contract-vulnerability-details-explained-b0cbc31a76e

[12] https://www.ccn.com/crypto-exchange-shapeshift-sees-criticism-for-mandating-memberships-with-kyc-norms