DEXs and Shamir's

November 25, 2019

guha.jayachandran@sjsu.edu

How to Trade?

- Directly with counterparty or

- Through an exchange

Why do Exchanges Exist?

Why do Exchanges Exist?

So you don’t need to know your counterparty.

The exchange brings everyone together.

The exchange matches buyers and sellers.

How Do Exchanges Work?

- Simple!
- Order book maintains bids and asks.
- Match is made, trade occurs, ledger is updated.
- Points of centralization?

Cryptocurrency vs. Stock Exchange

Cryptocurrency vs. Stock Exchange

In cryptocurrency,“cryptocurrency exchange” encompasses what is called a “brokerage” for stocks

What does brokerage do?

Why do Cryptocurrency Exchanges Exist?

So you don’t need to know your counterparty.

The exchange brings everyone together.

The exchange matches buyers and sellers.

You don’t need to trust your counterparty, only the exchange.

(The key “exchange” is between fiat and cryptocurrency.)

Downsides of Centralized Exchanges

- Security
  - We've seen the hacks!
- Fees
  - Like stock brokers and foreign currency exchanges
- Limited trading pairs

Decentralized Exchange

Can we construct a protocol such that there is no central party—while still not needing to trust counterparties?

Atomic Swap

Two parties that want to make a trade
can do it without any trust in each other
or any trusted intermediary

(Note this is usable not just for money,
but any on-chain information)

Sample Algorithm

1. Alice picks a random number x

2. Alice creates TX1 that says "Pay p BTC to Bob if the pre-image of H(x) is known and signed by Bob OR is signed by Alice and Bob"

3. Alice creates TX2 that says "Pay p BTC from TX1 to Alice, locked 48 hours in the future, signed by Alice"

4. Alice sends TX2 to Bob

5. Bob signs TX2 and returns it to Alice

6. Alice submits TX1 to the network

7. Bob creates TX3 that says "Pay q of alt-coin to Alice if the pre-image of H(x) is known and signed by Alice or is signed by Alice and Bob"

8. Bob creates TX4 that says "Pay q alt-coins from TX3 to Bob, locked 24 hours in the future, signed by Bob"

9. Bob sends TX4 to Alice

10. Alice signs TX4 and sends it back to Bob

11. Bob submits TX3 to the network

12. Alice spends TX3, revealing x (the pre-image of H(x))

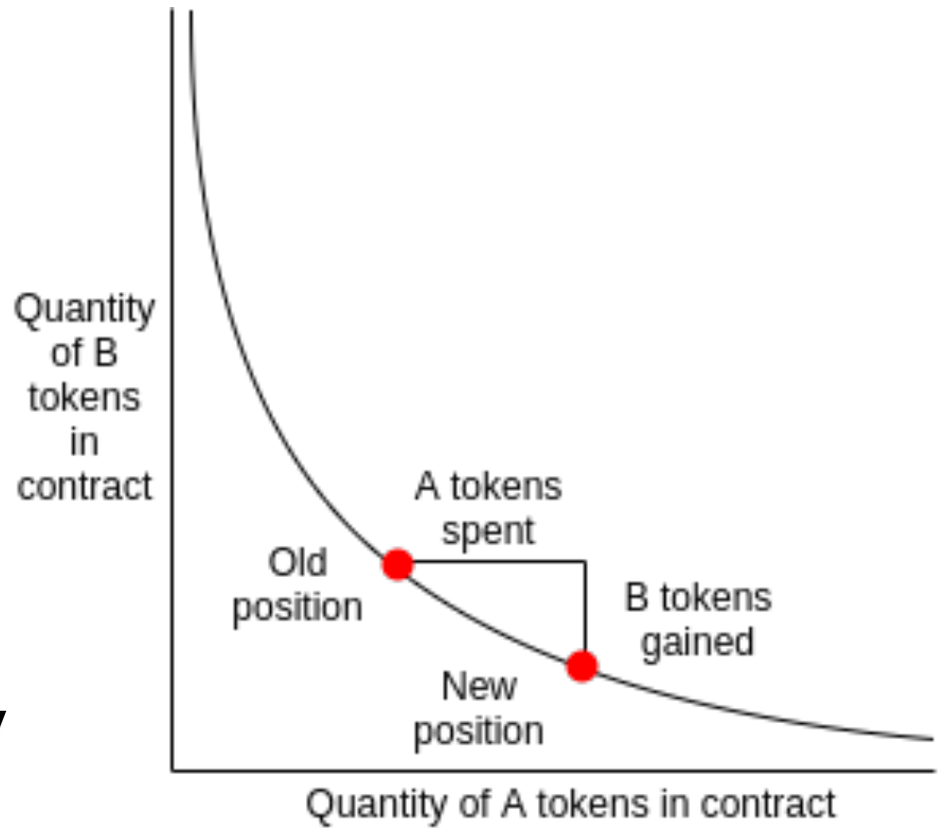
13. Bob spends TX1 using x

Example: 0x

- Peer to peer exchange on Ethereum of ERC20 tokens
- Smart contract
- Relayers exchange order book entries off-chain and settlement is on-chain

Example: Uniswap

- Again Ethereum/ERC20
- On-chain entirely
- 0.3% commission shared amongst liquidity providers
- Exchange rates determined by formula a*b = k



Examples of Not *Really* DEXs

- IDEX blocks New York
- Bancor lost customer funds

Current Status

No DEX that has nearly as much
liquidity as the centralized exchanges

Downsides of DEXs

- Fiat onramp lacking
- More potential for illegal activity
- No way to revert transactions
- No recourse if you lose your private keys

How to Keep Your Keys Safe?

- Memorize seed phrase
- Safe deposit box
- Etc.

Horcruxes…

- In Harry Potter, to kill Voldemort you need to destroy 7 magical objects hidden all around the world

- Can we take inspiration? To get our key, someone should need multiple pieces!

Better than Horcruxes…

- M of N pieces needed to reassemble the secret

- We do this with Shamir’s Secret Sharing (SSS)

Shamir's Secret Sharing

- Invented by Adi Shamir
  - He is also the "S" in RSA

- Call each piece (horcrux) a share

- Formally: Method to divide secret S into n shares S_1 to S_n such that:
  1. Knowledge of any k or more shares allows us to recover S.
  2. Knowledge of fewer than k shares leaves S completely undetermined.Call each piece (horcrux) a share

- Information theoretically sound

SSS

- Construct the below polynomial where a_0 is the secret (*S*) and a_1 to a_{k-1} are random natural numbers

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{k-1} x^{k-1}$$

- Evaluate this polynomial at *n* values of *x* to obtain *n* points
- Each of these points is a share
- Given *k* of those shares, you can interpolate to find a_0

Illustration

- Secret: 536837

- Let’s say we want 2 of 3 shares required

Illustration

- Secret: 536837

- Let’s say we want 2 of 3 shares required
  - k=2, n=3

Illustration

- Secret: 536837
- Let’s say we want 2 of 3 shares required
  - k=2, n=3
- Then f(x) = 53687 + rx where r is any natural we choose
- Evaluate at 3 different values of x to produce 3 shares
- Need only 2 of them to figure out f(x) and recover the secret

Finite Field Arithmetic

- Problem with preceding solution?

Finite Field Arithmetic

- Get information even with less than *k* shares
- Solution: Choose a prime *p* that is bigger than the number of shares and every a_i and calculate the points as f(*x*) mod *p*.

Finite Field Arithmetic