Barry Whitehat , Kobi Gurkan

Intro

We can post order on chain. But there is no way to advertise an order such that only a person who has a matching order can see the order.

This is a problem because people who see you want to make an order can use that information to move the market before your order is processed.

Therefore its interesting to find a way to match orders without publishing them.

The socialist millionaire problem solves this by allowing us to check if two values match privately. If not we do not gain any information about them.

Where we use this to privately search orders that match

Solution

I adversities on chain that I have an order on a certain pair. People who are looking to execute their orders on the same pair contact me directly.

I have my order a

. my counter party has their order b

. We then enter the following protocol together. a and b

are encoded as bits where the

first 64 bits denote the amount and the next 64 bits denote the price.

We then run the socialist millionaires protocol as describedhttps://en.wikipedia.org/wiki/Socialist_millionaires#Off-the-Record_Messaging_protocol

Attacks

1. An attacker can not follow the protocol and make it appear that orders do not match when they do.

We can use a zkps to prevent this. There are ZKP that are much lighter than snarks that we can use to do this.

1. State space enumeration attack

An attacker can guess what your order is and repeat millions of times until they get it right. We need to limit the rate of requests. This would require strong proof of individuality / anti Sybil in order to prevent this.

Conclusion

The order is not committed to on chain so you just have a chance to

1. Legitimately find someone who will match you order
2. Try and guess someones order.

The key point is that if someone lies they get one chance to guess your order. In the current implementation everything is public. In the worst case this devolves to the status quo with a bunch of extra steps. Using reputation systems and bonds seem like a good way to prevent users from abusing this.

We can also use reputation systems to limit the rate of requests.