Razor network

Technical lightpaper

This is a summary of the protocol. More details can be found in the whitepaper.

Design Rationale

We are building a decentralized oracle protocol using an ethereum based blockchain.
Schellingcoin [1] is a well known mechanism for building such a protocol.
Single round Schellingcoin games are vulnerable because:

1. P+e attacks [2]
2. Because the rounds have small periods, these rounds are automated. To overcome selective misinformation attack [3], the penalty needs to be minimum, so that the penalties are cancelled out over large number of rounds.
3. Having too small penalty would mean that stakers don't have a lot to lose and can vote lazily/collude. This affects the economic security.
4. Having high penalty means stakers will not recover from selective misinformation attacks.

Hence we will be using an iterated version of Schellingcoin game, where we have multiple rounds. This is similar to Augur, Kleros and Vitalik's recommendations to secure the protocol.

Razor protocol summary

- Razor can be used for automatic or manual processing of the queries.
- Automated processing delivers within a minute. Manual can take a few days.
- The client asks the protocol to fetch result from a URL. They have to pay:
  - The fee for the round
  - Validity bond
- They validity bond is on per URL, per client basis, rather than per query basis. It can be paid by anyone (app developers or users)
- If the client desires quick and cheap results, the first round is a short, automated round. The time for first round is around 60 seconds. The stakers automatically fetch the URL and report it to the contract using a commit-reveal scheme. Results are aggregated using weighted median. Incoherent stakers are penalized and coherent stakers are rewarded (according to median absolute deviation)
- In case someone is not happy with the result, the result can be disputed.
- To dispute a result, a dispute bond must be fulfilled. It can be paid by different users collectively. The losing stake of the round is automatically added to the dispute bond. The dispute bond must be fulfilled within time T, where T is the time period of current round.
- If the dispute bond is not fulfilled within the time limit, the result is considered "Confirmed"

- The dispute round is a manual round with period of a few days.
- If dispute round resolves the round as “invalid source”, and the result is confirmed, the validity bond is confiscated and distributed to losing stakers.
- The dispute round itself can be disputed in a similar way. The participating stake, dispute bond, economic security, bribe required to compromise, etc. doubles every round.
- If stakers collude or are attacked in round 1, successfully compromising the result, the round can be repeated with different/more stakers.
- Dispute rounds can themselves be disputed, query will be resolved in either of the following ways:
    - They can be disputed to higher and higher rounds, till the dispute bond is so high it cannot be fulfilled theoretically (it is of size more than the total supply)
    - There is a fork state similar to augur, where the protocol will fork and the market will decide the more valuable coin.
- This addresses both the selective misinformation and P+e attacks.

Example workflow of an application using Razor

- Apps need to call the oracle, wait for results to be confirmed and then use the result.
- Apps need to decide if they would like the first round to be manual or automated. The automated round requires a URL and selector. It is optional in manual rounds.
- Apps need to decide if in case of dispute, they would want to wait for the dispute rounds to be resolved, or they can cancel the tx in case of a dispute. (the dispute rounds will however continue on razor, even though the tx is cancelled at application side)
- Everyone is incentivised to watch the results and raise dispute automatically since they will earn a profit of 50% of the dispute bond contribution.
- This includes users, app developers, app stakeholders, razor stakers (especially the losing stakers), etc. To make sure the app doesn't break down due to a wrong oracle price, the app developers need to make sure they are actively monitoring and disputing the results.

Utility token

An erc20 compatible utility token called “Schell” will be used for staking purpose in Razor network.

Scalability

We will be deploying our own EVM chain with proof of stake and Honey Badger BFT.
This is because honey badger has the following features:

1. Censorship resistant due to threshold encryption
2. Asynchronous
3. Instant finality
4. O(N) communication complexity

The transactions will be bridged to other blockchains using a decentralized bridge.

[1]https://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed/
[2]https://blog.ethereum.org/2015/01/28/p-epsilon-attack/
[3] https://razor.network/whitepaper.pdf