A Truth-Inducing Sybil Resistant Decentralized Blockchain Oracle

Yuxi Cai*, Georgios Fragkos‡, Eirini Eleni Tsiropoulou‡, Andreas Veneris*†
*Department of Electrical and Computer Engineering, University of Toronto
†Dept. of Computer Science, University of Toronto
‡Dept. of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA

***Abstract*—Many blockchain applications use decentralized oracles to trustlessly retrieve external information as those platforms are agnostic to real-world information. Some existing decentralized oracle protocols make use of majority-voting schemes to determine the outcomes and/or rewards to participants. In these cases, the awards (or penalties) grow linearly to the participant stakes, therefore voters are indifferent between voting though a single or multiple identities. Furthermore, the voters receive a reward only when they agree with the majority outcome, a tactic that may lead to *herd behavior*. This paper proposes an oracle protocol based on peer prediction mechanisms with non-linear staking rules. In the proposed approach, instead of being rewarded when agreeing with a majority outcome, a voter receives awards when their report achieves a relatively high score based on a peer prediction scoring scheme. The scoring scheme is designed to be incentive compatible so that the maximized expected score is achieved only with honest reporting. A non-linear stake scaling rule is proposed to discourage Sybil attacks. This paper also provides a theoretical analysis and guidelines for implementation as reference.**

***Index Terms*—Decentralized Oracles, Peer Prediction, Staked Voting.**

I. Introduction

Distributed Ledger Technology (DLT), or blockchain technology, originated as a public ledger of monetary cryptocurrency transactions [1], but today it has evolved into a platform for multiple other applications with the use of smart contracts [2], [3], [4]. Blockchain platforms are unable to directly access information that is external to their system. This presents a problem as smart contracts often need access to such deterministically verifiable data so to reach consensus of execution [5].

Trusted entities that fetch external data onto blockchains are called *oracles*. The Astraea protocols [6], [7] is a series of decentralized blockchain oracle proposals. Those protocols leverage staked voting mechanisms agnostic to the blockchain consensus mechanisms while preserving decentralization and permissionless participation, to aggregate information from the participants. In both protocols, the aggregation and reward mechanisms are simple and deterministic. Both the outcome and reward are determined by majority voting, in other words, by the *popularity* of a particular answer. However, this may result in undesirable behaviors. One of the Astraea protocols, the paired-question protocol [7], ensures that there are equal expected amount of `True` and `False` proposals by requiring the submitters to submit antithetic proposal pairs. This indeed discourages one of the dishonest behaviors, *i.e.,* lazy voting (*e.g.,* always voting for `True` or `False`). However, a voter may report the exact opposite of their true opinion if there is a belief that their opinion is a minority, also known as a *herd behavior* [8]. Further, it is not immune to Sybil attacks, that is, when a voter creates pseudonymous identities to obtain disproportionately large influence on the system. Such attacks could be discouraged if one increments the cost of creating a new identity [9]. Although staked voting associates a potential cost linear to the number of replicated identity and the paired-question protocol further increases the cost by awarding the attacker when the antithetic questions have different majority outcomes, a rational voter is still indifferent between reporting with a single or multiple identities in the Astraea protocols.

This paper presents a peer prediction-based protocol with non-linear scaling of stake for decentralized oracles. Compared to the paired-question protocol described earlier, there are two main enhancements: *(1)* Rewards for voters are determined by a score, which is calculated with a light-weight scoring rule by referencing the voting behavior of other voters wrt the submitted answer; and, *(2)* Voting weight is scaled sub-linearly while award portion is scaled super-linearly wrt the submitted stake. The oracle assigns questions to voters and collects reports consisting of two components: a binary information answer and a popularity prediction. The oracle answer is determined by the majority of the information answer, weighted by the associated stakes and adjusted by a sub-linear function. Then, the oracle assigns a score to each report based on the accuracy and degree of agreement with peers. Only the top-scored voters are awarded, while the share of award is determined by their stake adjusted by a super-linear function.

One of the benefits of the peer prediction-based mechanism presented here when compared to the previous Astraea protocols is that the proposed system can be incentive compatible even when the voter believes in a minority opinion. That is, any minority voters are encouraged to vote according to their true opinion as they are expecting to receive an award. The other benefit is that, with the non-linear scaling stake scheme, an honest voter is incentivized to stake more onto a single report while the penalty to a participant to bias the oracle outcome via a Sybil attack is increased. In other words, the proposed framework penalizes Sybil attacks.

The remainder of the paper is organized as follows. Section II reviews existing blockchain oracles. Section III describes a general oracle model as well as defines notation used in the remaining paper. Section IV follows with a description of the proposed protocol, a proof of incentive compatibility of the scoring scheme, and an analysis on the expected outcome. Section V presents a guideline on the stake scaling rule, and discusses the advantages of the proposed protocol. Section VI concludes this paper.

II. Prior Art

A. Blockchain Oracles

In this paper, we define a *decentralized oracle* as one with the following two properties:

- Permissionless: members of the public can join without permission from existing users, and
- Equi-privileged: all system users have identical priority.

Previously known as `Oraclize.it`, the oracle in `Provable` [10] fetches data from a web source specified by a user along with cryptographic proofs to ensure that the retrieved information is genuine from the chosen source. Similarly, `Town Crier` [11] makes use of Intel's Software Guard Extensions hardware (IntelSGX) [12] so as to prevent alteration of data by malicious operating systems. Evidently, both protocols rely on a centralized server to handle query requests. This exhibits a strong form of centralization that violates the equi-privilege property of a decentralized oracle as set above. Prediction market `Augur` [4] utilizes a validation-dispute protocol in which token holders report and/or challenge reported outcomes. However, for each market a designated reporter has a privilege to report before the others, which violates both the permissionless and equi-privilege properties above. `Chainlink` [13] builds a marketplace to aggregate information retrieved from multiple oracles. The protocol requires the query submitter to specify the data source, which limits the source of data and subjects the system to denial of service attacks.

Multiple attempts have been made to solve the problem of proving of data authenticity from Transport Layer Security (TLS), a crytopographic protocol providing communication security, without violating decentralization. `TLS-N` [14] requires the server to include authenticated TLS records in its transactions, which is a significant change to the protocol on the server-side. `Practical Data Feed Service (PDFS)` [15] introduces data transparency through authenticating, recording and verifying TLS transactions with smart contracts. Finally, `DECO` [16] allows proof of data authenticity through zero-knowledge proofs (*i.e.*, eliminates need to disclose full data), therefore eliminates the need for trusted hardware or server-side modifications. Although useful, this is a computationally-expensive operation.

B. Voting-based decentralized oracles

In the original Astraea [17] mechanism, there are three groups of users: *submitters*, *voters*, and *certifiers*. A submitter submits a proposition by paying a bounty to reward participants. Voters submit a relatively small stake and are randomly assigned a proposition to answer. Certifiers submit a relatively large stake and choose a proposition to answer. There are two reward pools for rewarding certifiers and their interaction with the system is shown in Figure 1. The outcome of a proposition is determined by comparing the majority answer by the certifiers to the one by the voters:

- Voters and Certifiers Agree: The oracle outcome is the majority answer. Both certifiers and voters are rewarded for agreement with the majorities and penalized for disagreement proportionally to their stake, or
- Voters and Certifiers Disagree: The oracle outcome is undetermined. All certifiers lose all their stake, while voters are not penalized (*i.e.*, their stake is refunded).

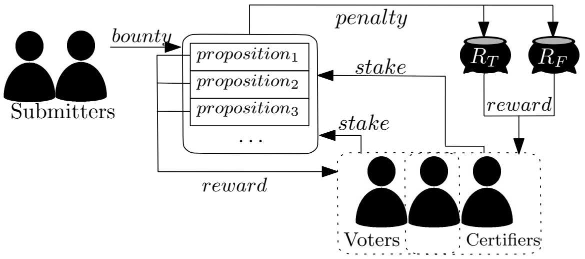

Figure 1: Overview of monetary flow in Astraea protocol.

To avoid draining of the reward pool, the proposed mechanism depends on the assumption that the submitters are equally likely to submit both `True` and `False` propositions, and hence the honest oracle outcomes are equally likely to be either of the answers. This protects against a lazy equilibrium but makes the underlying system analysis difficult.

An improvement was proposed to simplify the reporting and outcome determination of Astraea, and to disincentivize lazy voting [7]. The *paired-question* protocol requires a submitter to always submit two questions with opposite binary `True` or `False` answers. Further, the protocol has only one voting group instead of two. Similar to Astraea, a voter stakes to be assigned a random proposition and receives a reward only if the majorities of the two antithetic questions disagree with each other while the voter agrees with the majority. An extensive analysis of this protocol shows that it efficiently lowers the expected payoff of lazy voting, making honest voting the preferable voting strategy.

Shintaku [18] has also one voting group. After posting a stake, a voter receives two random propositions. Voters submit both answers to the oracle and are rewarded only when they *(i)* agree with the majority, and *(ii)* answer the proposition-pair differently. However, lazy voting can have positive payoffs unless penalties for disagreement are at least twice as large as rewards for agreement [7], a rather strict requirement.

In all the above protocols, both the aggregation and reward mechanisms are majority voting, whose outcome depends only on the popularity of the reported answers. However, under the majority voting protocols, a voter may incline to report the exact opposite of their true opinion if there is a *prior belief* that their opinion may be a minority. In other words, the expected payoff of dishonest voting (*i.e.*, voting against one's private opinion) is higher when a voter expects their private opinion to be a minority. Consequently, there may be cases where the majority voting protocol discourages honest reporting if the voters have an expectation on the "popularity" of the answers. Voters are also indifferent between voting with a single or multiple identities as their voting/reward shares increase linearly in the submitted stakes, which may make the protocol(s) prone to Sybil attacks.

III. Preliminaries

A. A Decentralized Oracle Model

Without loss of generality, we assume the proposed decentralized oracle operates as a smart contract on a blockchain platform such as Ethereum [19] or Hyperledger [20]. This platform is hereafter referred to as the oracle *executor*. Any user can submit *Boolean propositions* to the executor at any time, or join as a voter. The executor maintains a list of *active* Boolean propositions, which are open for any voter to respond to. When the duration for a proposition is reached as specified during submission, the proposition is considered *closed*. The executor ceases to accept new responses, aggregates responses, calculates scores, and distributes rewards to voters.

We model the interaction between the voters and the decentralized oracle as an incomplete information game borrowing from the setting of the Bayesian Truth Serum (BTS) mechanism [21]. BTS and its refinements [22], [23] are peer prediction mechanisms, which are incentive compatible to incentivize truthful report of private signals (*e.g.*, opinions or experiences), while an observable objective outcome is not available (*e.g.*, "is Picasso the greatest artist of all time?").

We define a user who reports data to an oracle as a *voter*. The set of all voters participating in the decentralized oracle protocol is $\mathcal{V}$. For a particular proposal a random subset of the voters is chosen. The size of the subset is much less than $|\mathcal{V}|$ so that, effectively, to make voters impossible to choose which proposal to vote on. In this work, we assume that all voters are risk-neutral and individually rational, *i.e.*, they seek to maximize their own expected payoff. We adopt a belief system based on *Bayesian inference* assuming all voters share the same belief system consisting of *signals* and *states* [24] Each voter i observes a signal, *i.e.*, a *private opinion (PO)*, on the proposition represented by the binary random variable $\mathrm{PO}_i \in \{1, 0\}$, with 1 representing `True` and 0 representing `False`. State T is a random variable that can adopt values in $\{1, ..., m\}$ $(m \geq 2)$, representing all possible true states of the world (*i.e.*, it is possible that voters who think Picasso is the greatest artist of all time make up 70% of all voters, while it is possible that only 30% of them think so). Each state is a probabilistic distribution on the possible outcomes, consisting of $\mathrm{P}r(\mathrm{PO}_i = 1|T = t)$ and $\mathrm{P}r(\mathrm{PO}_i = 0|T = t)$ $(t \in \{1, ..., m\})$, of the proposition.

One important assumption is the *Common Prior Assumption (CPA)*, as justified in [25], consisting of a shared probabilistic distribution over all states ($\mathrm{P}r(T = t)$) and initial beliefs in each state ($\mathrm{P}r(\mathrm{PO}_i = po|T = t)$), where po denotes the realized value of their private opinion PO_i. Under CPA, the description of the states can be further simplified to $\mathrm{P}r(\mathrm{PO} = 1|T = t)$ and $\mathrm{P}r(\mathrm{PO} = 0|T = t)$ $(t \in \{1, ..., m\})$. We require the common prior belief to be admissible, which is defined as follows [22]:

Definition 1. *The common prior belief is* admissible *iff:*

1) *There are two or more states, i.e.,* $m \geq 2$*;*
2) *All states have positive probability, i.e.,* $Pr(T = t) > 0$*;*
3) *States are distinct; and,*
4) *The signal beliefs conditional on state are fully mixed, i.e.,* $0 < Pr(PO = po|T = t) < 1$ $\forall t \in \{1...m\}$ *and* $po \in \{1, 0\}$.

It is notable that admissibility is a weak requirement as any prior belief with two or more unique states can be mapped to an admissible prior because of conditions 2) and 4) above.

Further, we assume that all voters are *Bayesian thinkers* who update their probabilistic beliefs on states based on the common prior belief and their private opinion. The resulted private prediction is later an input to the user's reporting strategy function. The update process is defined as follows.

Definition 2. *The posterior belief is the updated probabilistic belief according to Bayes Theorem on each state given a set of received signals.* Private prediction PP_i *is the popularity prediction on* 1 *(or* `True`*) by voter* i *with a private opinion* po_i *as the received signal.*

$$\mathrm{P}r(T = t|\{po_i\}) = \frac{\mathrm{P}r(po_i|T = t) \cdot \mathrm{P}r(T = t)}{\mathrm{P}r(po_i)}$$

$$\begin{aligned}\mathrm{PP}_i &= \mathrm{P}r(1|\{po_i\}) \\ &= \sum_{t \in \{1...m\}} \mathrm{P}r(1|T = t) \cdot \mathrm{P}r(T = t|\{po_i\})\end{aligned}$$

B. Voter Responses

We define a response tuple $RT = (\mathrm{IR}, \mathrm{PR})$, consisting of an information report $RT.\mathrm{IR}$ and a prediction report $RT.\mathrm{PR}$. An information report is a binary opinion ($RT.\mathrm{IR} \in \{1, 0\}$) and a prediction report is the predicted proportion of information reports being 1 ($RT.\mathrm{PR} \in [0, 1]$). Limiting the discussion on the submitted tuples for a particular proposition, let $\mathbf{RT} = \{RT_1, RT_2, ..., RT_n\}$ be the set of voters' responses, where n denotes the number of voters answering the proposition and $n << |\mathcal{V}|$. It is notable that n is not fixed while we shall see later that it needs to be lower-bounded for a reasonable chance of a "correct" oracle outcome. Each voter i has a *voting strategy* $\sigma_i((\mathrm{PO}_i, \mathrm{PP}_i)) = RT_i$ if they are chosen to vote on the proposition. For example, an *honest* voter has $\sigma_i((\mathrm{PO}_i, \mathrm{PP}_i)) = (\mathrm{PO}_i, \mathrm{PP}_i)$, while a *lazy* voter has either $\sigma_i((\mathrm{PO}_i, \mathrm{PP}_i)) = (1, 0.5)$ or $\sigma_i((\mathrm{PO}_i, \mathrm{PP}_i)) = (0, 0.5)$ for any proposition. We divide $\mathbf{RT}$ into subsets of responses sharing the same IR; then we have $\mathbf{RT_1} = \{RT \in \mathbf{RT} : RT.\mathrm{IR} = 1\}$ and $\mathbf{RT_0} = \{RT \in \mathbf{RT} : RT.\mathrm{IR} = 0\}$. Finally, we define two tuples of random variables: let $\Gamma = RT$ denote the private belief tuple of a voter selected randomly on the proposition, and $A = RT$ denote the answer tuple of a random voter.

1) Popularity and Correctness: Objective truth is not directly accessible by the oracle even if there is one, let alone when the proposition is subjective. So to rigorously define the "correctness" of the oracle answer, we set out the below [7]:

Definition 3. *The* Most Probable Private Opinion (MPPO) *is a randomly selected voter's most likely private belief on a particular proposition.*

$$\mathrm{MPPO} \triangleq \begin{cases} 1 & , \mathrm{P}r(\Gamma.\mathrm{IR} = 1) > 0.5 \\ 0 & , \mathrm{P}r(\Gamma.\mathrm{IR} = 1) < 0.5 \\ \emptyset & , \mathrm{P}r(\Gamma.\mathrm{IR} = 1) = 0.5 \end{cases}$$

Definition 4. *The oracle answer is correct if it is equal to* $MPPO$.

For the rest of the paper, we assume there is a defined correct answer. The probability of a random voter providing an answer with IR equal to MPPO is defined as follows.

$$c \triangleq Pr(A.\mathrm{IR} = \mathrm{MPPO})$$

2) Prediction means: For prediction reports, $\overline{P}_{-i,1}$ denotes the geometric mean of prediction reports $RT.PR$ in $\mathbf{RT_1}$ excluding voter i (the corresponding set of responses is denoted as $\mathbf{RT_{-i,1}} = \mathbf{RT_1} - \{RT_i\}$). Namely, $\overline{P}_{-i,1} = G(\mathbf{RT_{-i,1}})$, with G denoting the geometric mean. Similarly, $\overline{P}_{-i,0}$ denotes the geometric mean of prediction reports in $\mathbf{RT_0}$, excluding voter i (the corresponding set of responses is denoted as $\mathbf{RT_{-i,0}} = \mathbf{RT_0} - \{RT_i\}$). Namely, $\overline{P}_{-i,0} = G(\mathbf{RT_{-i,0}})$.

The prediction means serve as references to compare with chosen voter's prediction report, which is later cooperated into the award measurement.

C. Scoring Rule

A *binary scoring rule* assigns a score on a prediction $q \in [0,1]$ based on a binary outcome $w \in \{0,1\}$. We define a binary scoring rule as *strictly proper* if the voters uniquely maximize their expected score by honestly reporting their truthful prediction. Let q be a prediction, and w be the realized binary outcome, the strictly proper binary quadratic scoring rule (R_q) [26] is defined as follows.

$$R_q(q,w) = \begin{cases} 2q - q^2 & , w = 1 \\ 1 - q^2 & , w = 0 \end{cases}$$

D. Voting Weight and Reward Share

As a mechanism against Sybil attack, the protocol relates the voting weight and reward share of a voter i to their submitted stake s_i. We define the voting weight of the response by voter i toward an oracle outcome as $f(s_i)$. The normalized weight is therefore $\frac{f(s_i)}{\sum_{i' \in \{1,\ldots,n\}} f(s_{i'})}$. In a traditional staked voting scheme [6], [7], the weight increases linearly in s_i ($f(s_i) = s_i$). The voting weights are then utilized to find out the oracle outcome. We define the reward share of the response by voter i during reward distribution as $g(s_i)$. In a traditional staked voting scheme, the count increases linearly in s_i ($g(s_i) = s_i$). The reward share, normalized by the total share, determines the portion of total reward that a voter receives if eligible.

IV. A Truth-Inducing Protocol

This section introduces the proposed decentralized oracle protocol. Initially, a description of the general process is provided, followed by a proof on Bayes-Nash incentive compatibility and an analysis of the expected oracle outcome.

A. Description

Similar to the *paired-question* protocol, at any time a *submitter* can add propositions to the active proposition list, and a *voter* can vote on an active proposition.

*1) **Submitting proposals**:* To create a new query, in a single transaction, a submitter provides:

- A proposition-pair with potentially antithetic answers,
- A bond which is returned to the submitter if the submitted propositions have opposite oracle outcomes,
- A bounty B which is used to reward voters, and
- A duration which specifies the amount of time available for voting before the proposition is closed.

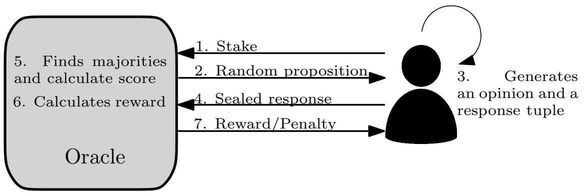

Figure 2: Overview of interaction between a voter and the oracle in the proposed protocol.

The bounty is used to pay voters, the bond is deposited for quality control purpose, and the duration specifies the voting period on the two propositions. After the duration of the proposition, there are two possible cases, whether the answers of j and j' converge to different outcomes:

- Yes, the bounty funds the voter rewards, and the bond is returned to the submitter, or
- No, the bounty is returned to the submitter, while the bond is equally split among the other active propositions.

*2) **Placing votes**:* The voting process is presented in Figure 2: *(i)* A voter i posts a stake s_i within a range $s_i \in [s_{min}, s_{max}]$ to the oracle executor; *(ii)* The oracle assigns a random active proposition to the voter; *(iii)* The voter generates a response tuple RT_i based on their voting strategy σ_i, private opinion PO_i and private prediction PP_i; and *(iv)* The voter returns the sealed response tuple RT_i to the oracle. Once the proposition closes, the voters reveal their responses and the oracle determines the outcome and rewards.

The purpose of a random proposition assignment is to increase the cost of (unwanted) collusion. Consider an entity that intends to control majority of responses to a proposition with n voters. Recall, n is a small fraction over the complete number of voters. The selection of voters can be seen as a series of n Bernoulli trials where the possible outcomes are sets of either colluded voter or non-colluded voters. Evidently to control majority responses out of the n randomly chosen voters, the attacker must control a significant portion of total voters.

*3) **Outcome determination**:* Once the duration of a proposition expires, it is considered closed. The oracle determines the weighted majority of the information reports, which is the oracle outcome of the proposition. Each response is weighted by the submitted stake adjusted with a sub-linear function. The oracle outcome o is therefore determined as follows:

$$o = \begin{cases} 1 & , \sum\limits_{i \in \{i : RT_i \in \mathbf{RT}_1\}} f(s_i) > \sum\limits_{i' \in \{i' : RT_{i'} \in \mathbf{RT}_0\}} f(s_{i'})) \\ 0 & , \sum\limits_{i \in \{i : RT_i \in \mathbf{RT}_1\}} f(s_i) < \sum\limits_{i' \in \{i' : RT_{i'} \in \mathbf{RT}_0\}} f(s_{i'})) \\ \emptyset & , otherwise \end{cases}$$

For the rest of the paper, we assume there is always a defined outcome for the proposition. Section V-A later suggests, justifies and analyzes some example functions. The actual function to be used is left to the implementation.

The oracle then checks if the outcomes for the antithetic propositions diverge. If the outcomes converge to the same answer, the submitter loses their bond, which is distributed over all other active propositions, while the voters receive a refund of their stake. Otherwise, the submitter's bond is returned, and the payoffs for voters are specified in the following subsection.

*4) **Rewarding rule**:* After checking that the outcomes for the antithetic propositions are different, the oracle assigns a score to each response $RT_i \in \mathbf{RT}$, as follows:

1) Calculate the *prediction score* $u_{i,PR}$ by applying the quadratic scoring rule on the prediction report and the information report of a randomly chosen reference response tuple $RT_{i'}$:

$$u_{i,PR} = R_q(RT_i.\mathrm{PR}, RT_{i'}.\mathrm{IR})$$

2) Calculate the *information score* $u_{i,IR}$ by subtracting 1 with the mean squared error between the prediction report and the geometric mean of the prediction reports of all reports sharing the same information report:

$$u_{i,IR} \triangleq \begin{cases} 1-(\overline{P}_{-n,1}-RT_i.\mathrm{PR})^2 & \text{if } RT_i.\mathrm{IR}=1 \\ 1-(\overline{P}_{-n,0}-RT_i.\mathrm{PR})^2 & \text{if } RT_i.\mathrm{IR}=0 \end{cases}$$

3) Calculate the total score u_i as a sum of the two scores above: $u_i = u_{i,\mathrm{IR}} + u_{i,\mathrm{PR}}$.

After calculating the score u_i for each voter's response tuple, the response tuples are ranked according to the associated score. The voters who have submitted responses with high-ranked scores should be rewarded, while the portion of voters being rewarded is left open to the implementation. A guideline for setting those system parameters can be found in the following section. The key idea is to reward the honest voters regardless of whether they are in agreement or disagreement with the outcome. In order to encourage rational behavior, the oracle distributes the bounty for the proposition among voters in proportion to the stake deposited adjusted by a super-linear function.

Let R denote the set of voter that are eligible for a reward. For each $i \in R$, the voter i receives a normalized reward share, $\frac{g(s_i)}{\sum_{i' \in R} g(s_{i'})}$, of the total reward. Recall that the bounty available to the paired-propositions is B, the actual reward for voter i, assuming they have submitted response for only one of the paired questions, is $\frac{g(s_i)}{2\sum_{i' \in R} g(s_{i'})}B$.

B. Protocol Analysis

1) A Bayes-Nash Incentive Compatible Scoring Rule: In this section, we show that honest reporting uniquely maximizes the expected score of the voter if they believe that all other voters are honest. Since the score is directly related to the rewards a voter is able to receive (*i.e.*, the higher the score is, the more likely a voter can receive a reward based on the rewarding schemes), a rational voter will seek to maximize her expected score.

We now provide a formal proof that the proposed protocol is strictly Bayes-Nash incentive compatible given an admissible prior.

Lemma 1. ***Strict Properness of Quadratic Scoring Rule [26].*** *Let $q \in [0,1]$ be the private prediction about a binary event of a voter. Suppose the voter is rational, and the score is calculated based on a quadratic scoring rule. The voter uniquely maximizes their expected score by reporting $RT.PR = q$.*

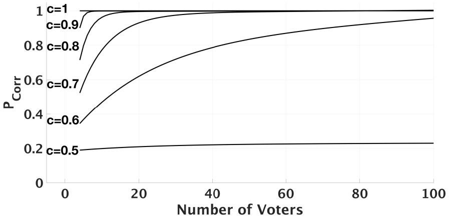

Figure 3: Probability of correctness as a function of n and c

Lemma 2. ***Ranges of Posterior Belief [22].*** *It holds that $1 > \Pr(1|\{1\}) > \Pr(1) > \Pr(1|\{0\}) > 0$ for all admissible priors.*

Both Lemmas 1 and 2 above are known results from prior literature. We refer the interested reader to [22], [26] for further details.

Lemma 3. ***Strict Properness of Prediction Score.*** *A voter uniquely maximizes their expected information score by truthfully reporting their private prediction if all other voters are honest.*

Proof: The expected probability that a random reference response tuple containing a information report of 1, from the perspective of voter i, is $\Pr(RT_{i'}.\mathrm{IR} = 1) = \Pr(1|PO_i)$. According to Lemma 1, as prediction score is calculated with a quadratic scoring, the voter uniquely maximizes the expected score by reporting $RT_i.\mathrm{PR} = Pr(1|PO_i)$. This agrees with the strategy of a honest voter, hence honest voting maximizes the expected prediction score. ■

Lemma 4. ***Strict Properness of Information Score.*** *A voter uniquely maximizes their expected information score by truthfully reporting their private opinion if all other voters are honest.*

Proof: Given all other voters are honest and following Lemma 2, a voter would expect $(PP_i - \overline{P}_{-n,PO_i})^2 < (PP_i - \overline{P}_{-n,\neg PO_i})^2$. Therefore, reporting $RT_i = PO_i$ yields strictly higher information score from the perspective of the voter. ■

Theorem 1. *The proposed scoring rule is Bayes-Nash incentive compatible.*

Proof: By Lemma 3, the pure strategy of reporting the actual private prediction uniquely maximizes the expected prediction score. By Lemma 4, honest reporting of private opinion uniquely maximizes the expected information score. It follows that the strategy of honest reporting maximizes the overall expected score. Therefore, the proposed scoring rule is Bayes-Nash incentive compatible. ■

2) Expected Outcome: The outcome determination of the proposed protocol is weighted majority-based voting of the submitted information report. Consider a proposition answered by n honest voters. Recall that c is the probability that a randomly selected response agrees with MPPO. The probability of correct oracle outcome, P_{Corr} is the probability that a majority of voters believes in MPPO.

In this subsection, we assume all voters submit the same amount of stake for the sake of simplicity. Given that generation of private opinions by any voter i and i' $(i \neq i')$ is independent, then generation of private opinion is simply a series of n Bernoulli trials, each with probability c to agree with MPPO. Hence, P_{Corr} can be calculated as follows:

$$P_{\text{Corr}} = 1 - B\left(\left\lfloor \frac{n}{2} \right\rfloor, n, c\right)$$

where $B\left(\left\lfloor \frac{n}{2} \right\rfloor, n, c\right)$ is the cumulative binomial density function.

Given all honest voters, the probability of correct output associated with different c is shown in Figure 3. If only a few voters are chosen, only propositions with widely accepted answers are likely to come out correctly. On the other hand, even if a proposition is highly contentious (with c close to 0.5), the oracle will agree with MPPO with high probability provided there are enough voters. This is the same as any other oracle protocol based on majority voting while the proposed protocol results in this outcome without a rewarding scheme that relies on the most popular outcome, thereby avoiding herding effects. A reasonable minimum number of voter is 30 as a relatively controversial proposition (with $c = 0.6$) would have a chance of around 70% to come out correctly.

V. Improving Resistance to a Sybil Adversary

This section presents an implementation of the scaling rule in voting-based oracles. Its efficacy in discouraging Sybil attacks is demonstrated followed by a discussion about the advantages of the proposed system.

A. Stake Scaling Functions

There are two stake scaling functions in the proposed protocol that makes it different from other staked voting systems. Recall that the protocol adjusts the voting weight of a voter as a sub-linear function of their submitted stake. Let this function be denoted as $f(s)$. The second scaling function is the one applied during reward allocation. This function is a super-linear function, which we denote as $g(s)$. For the sake of simplicity, let $f(s_{min}) = g(s_{min}) = 1$, hence a report with minimal stake represent a unit of voting weight/reward share. Such a paired stake scaling rule has several advantages. Firstly, it improves Sybil resistance by increasing the awards per stake on a single identity super-linearly comparing to the linear increment on multiple identities. Secondly, it prevents a single entity from having dominant voting power and discourages the forming of voting pools by a sub-linearly increasing voting weight.

For our analysis, we consider the family of sub-linear functions of the form $f(s) = \alpha\sqrt{s} + (1-\alpha)s$ with $\alpha \in (0, 1]$. We also choose the family of super-linear functions of the form $g(s) = \beta s^2 + (1-\beta)s$ with $\beta \in (0, 1]$. By varying α and β, we adjust the rates of change of voting weights and reward shares in response to changes in stake amount. To demonstrate, we assume there are a fixed number of honest voters chosen by the system. Each of those voters has staked s_{min}. Additionally, there is one voter i who has staked s_i. We now consider an extreme case where all voters are eligible to receive a reward, namely $x = 1$. Varied by the parameters α for $f(s)$ and β for $g(s)$, the normalized voting weight and the expected normalized reward share for voter i are shown in figure 4 and 5 respectively. Evidently, in any staked voting system when voters control more than 50% of the total stake, they can manipulate the outcome. Therefore, we only consider a voter with strictly less than 50% of the total stake. Namely, s_{max} should be set so that an entity cannot stake a unreasonably high amount on a response.

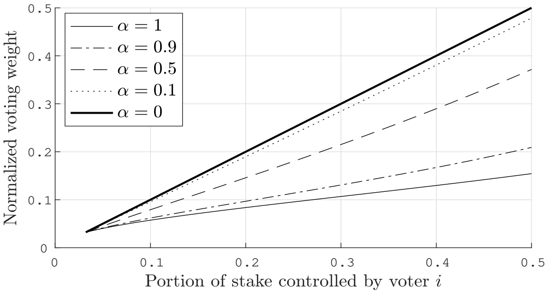

Figure 4: Normalized voting weight varies by portion of stake controlled with different suggested scaling functions.

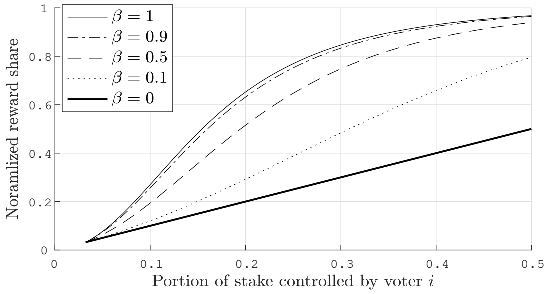

Figure 5: Normalized reward share varies by portion of stake controlled with different suggested scaling functions.

B. Expected Utility and Sybil Resistance

In this subsection, we will demonstrate the efficacy of the paired stake scaling rule in improving Sybil resistance. The reward of a voter is based on the calculated score of the voter's report instead of the oracle outcome. Recall that the voter accepts a reward when: *(i)* there exists a majority in the information report, *(ii)* the proposition has an opposite majority answer from the antithetic question, and *(iii)* their report receives a score that is ranked top among all reports. In the following analysis, we assume that both *(i)* and *(ii)* conditions are satisfied, and all voters are equally likely to be top-scored by the protocol hence awarded. It is worth mentioning that only a minority voter would ever have incentive to perform such an attack.

Consider an example scenario where there are 30 honest voters chosen for a proposition of interest. As adjusted in IV-B2, this quorum is a reasonable number of voter required. Let each of them set a stake of $s_{min} = 1$, and there is one additional voter i with stake s_i. Assume that the voter i gains a utility of u_h if the oracle outcome is $o = PO_i$, or gains u_L otherwise. The total available reward for the proposition is $\frac{B}{2}$ and the award portion is $x \in (0, 1]$, namely, $\lfloor 30x \rfloor$ voters are rewarded. Additionally, we assume that voters are chosen randomly if there is a tie in the assigned score, consequently the number of voter rewarded is constant. Furthermore, to model the costs of a voter, we assume that the voters incur a fixed cost K and a variable cost k. The fixed cost K is the cost

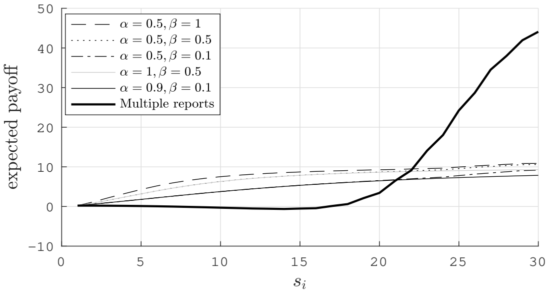

Figure 6: The expected payoff of voter i on a single report versus multiple reports. $c = 0.9$ and $x = 1$.

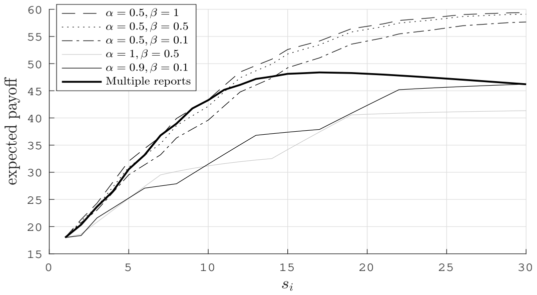

Figure 7: The expected payoff of voter i on a single report versus multiple reports. $c = 0.55$, $u_h = 50$ and $x = 1$.

for generating and submitting the first response. On the other hand, the variable cost k is the cost of submitting an additional response with a pseudonymous identity. Evidently, $k > K$. This is because, in order to submit an additional response, the voter needs to either buy an identity from voters who have been assigned the proposition or to create/stake on a sufficient amount of identities before the proposition assignment (given that a random subset of voters is chosen from the complete set of participants for any given proposition).

In Figure 6, we assume that $u_h = 50$, $u_L = 0$, $\frac{B}{2} = 10$, $x = 1$, $K = 0.1$, $k = 0.3$ and $c = 0.9$. Notably, $c = 0.9$ means that the voter is against a mostly agreed answer (*i.e.,* by 90% of the voters). Although $u_h = 50 > 10 = \frac{B}{2}$, the voter is only incentivized to perform a Sybil attack when they have more than $\frac{20}{20+30} = 40\%$ of the total stake. This holds even with the choice of α and β leading to the lowest expected payoff. In Figure 7, we show a scenario where the protocol is no longer Sybil resistant. With all other parameters unchanged, let $c = 0.55$ such that the voter is a minority in a relatively controversial proposition. Most of the suggested combinations of scaling functions no longer discourage Sybil attack in such a setting. This is because, with a contentious proposition, the expected probability to perform a successful attack is higher. Therefore, when the utility received from a favorable answer is high comparing to the potential award, the voter would choose to perform such an attack. In contract, if the utility from the favorable answer is less than the total available reward, an attack is not reasonable even if the chance of success is high. This is illustrated in Figure 8, with $u_h = \frac{B}{2} = 10$. In the scenario described above parameter β affects the shape more noticeable than α. A higher β leads to higher expected payoff for non-attacker if the all voters are awarded.

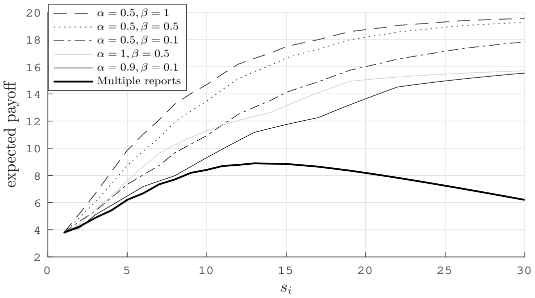

Figure 8: The expected payoff of voter i on a single report versus multiple reports. $c = 0.55$, $u_h = 10$ and $x = 1$.

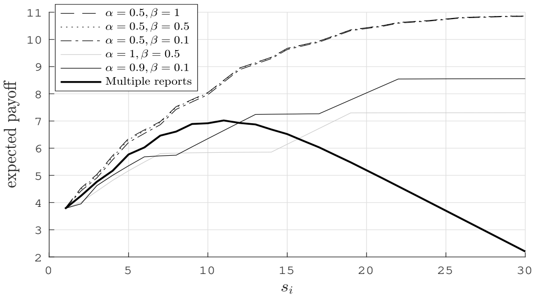

Figure 9: The expected payoff of voter i on a single report versus multiple reports. $c = 0.55$, $u_h = 10$ and $x = 0.1$.

Relaxing the award portion so that $x < 1$, the voter is further disincentivized to perform an attack given that not all identities can receive an award. We consider the scenario where only the top 10% of the voters are rewarded, that is $x = 0.1$. As shown in Figure 9, most combinations of scaling functions provide a strictly higher payoff for the non-attacker while two of them encourage attacks with stake amount less than 12 or 16 respectively. With this setting, it is also noticeable that α affects the shape of the payoff curve more effectively than β. Conclusively, if the reward fraction is small, a larger α should be chosen to discourage attacks.

To summarize, the protocol is Sybil resistant if the total reward available is relatively high comparing to the utility a potential attacker receives from having the favorite oracle outcome. Meanwhile, α and β should be chosen taking the expected utility from the favorable outcome, total available reward as well as the reward fraction into consideration.

C. *Adversarial Effects*

One possible adversarial behavior is to push the oracle outcome toward $\neg MPPO$ through a Sybil attack. As discussed previously in Section V-B, we show that by choosing the correct system parameters, staking on a single report gains higher expected payoff than on multiple ones. Consequently, it is not incentive compatible to push against $MPPO$. Even when such an attack is reasonable, it requires the adversarial player to control a considerably large portion of total voters, given the remaining voters are honest, in order to manipulate the result for a single proposition. The chances are decreased even further by the overall pair-question setting because the adversarial voter doubles their costs by manipulating the additional antithetic proposition.

As the rewards depend only on the scores of the voters, another possible adversarial behavior is to achieve higher score than any honest voters. However, as proven before, there is no other strategy that can maximize the expected score from the point of view of a voter. Consider an extremely knowledgeable adversarial voter who has knowledge of the behavior of all other voters (*i.e.*, all the sealed response tuples). It is notable that the cost of surveying the opinions of all voters is high while the adversarial voter will also need to share the award with other top-ranking voters. Even if such a voter exists, the adversarial voter will not be able to secure a maximum score given the randomness in choosing the reference agent.

This randomized process can be approached in many ways. A possible approach is to utilize the hash of the block, a random number generator such as Randao [27] on Ethereum, or through the use of a Verifiable Delay Function [28]. Moreover, an adversarial voter may affect the prediction means for information score calculation by replicating identities. A possible solution is to use the median of prediction instead of the mean, hence it is more robust against outliers and manipulation.

D. Advantages of the Proposed Protocol

In the previous ASTRAEA protocols, the reward depends on an agreement with majority. Consider a belief model discussed in the previous section, a rational voter would switch to a popular answer to improve expected rewards. However, under the proposed protocol, honest voters are incentivized to report their private opinion regardless of expected popularity. One may argue that both previous ASTRAEA protocols may be more efficient as they encourage faster convergence to a majority answer. However, under circumstances where the popularity of each answer is very "valuable" (such as elicitation of feedback or governance decision making), the proposed protocol is a *much* better option as the popularity of each answer provides an honest measure of how supported the outcome is for future decision-making.

Another significant advantage is Sybil resistance introduced by the stake scaling functions. Under the right choice of rule, Sybil attack is strictly unfavorable even with a considerable amount of stake. However, a voter with a higher stake is getting more reward than voters with the same stake in a linear staking system. One may argue that this leads to uneven distribution of wealth. Evidently, this can be limited by adjusting the max amount of stake per response, s_{max}.

VI. CONCLUSION

This work illustrates a truth-inducing decentralized oracle protocol where voters are rewarded based on the score associated with their reports with a non-linear scaling system. First, we propose a peer prediction-based scoring rule, and show that its scoring scheme is Bayes-Nash incentive compatible. Then, we list the suggested scaling function as well as guidelines to the system parameters. Finally, we compare the behavior of the proposed protocol with previously ASTRAEA protocols to demonstrate its benefits. As a possible future extension, the addition of a reputation system would allow distribution of awards based on previous performance and incentivize sustainable participation.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[2] C. Mussenbrock and S. Karpischek. (2017) Etherisc whitepaper. [Online]. Available: https://etherisc.com/files/etherisc_whitepaper_1.01_en.pdf

[3] IBM. Blockchain for supply chain. [Online]. Available: https://www.ibm.com/blockchain/supply-chain/

[4] J. Peterson, J. Krug, M. Zoltu, A. K. Williams, and S. Alexander, "Augur: a decentralized oracle and prediction market platform."

[5] G. Greenspan. (2016) Why many smart contract use cases are simply impossible. [Online]. Available: https://www.coindesk.com/three-smart-contract-misconceptions

[6] J. Adler, R. Berryhill, A. Veneris, Z. Poulos, N. Veira, and A. Kastania, "Astraea: A decentralized blockchain oracle," *arXiv preprint arXiv:1808.00528*, 2018.

[7] M. Merlini, N. Veira, R. Berryhill, and A. Veneris, "On public decentralized ledger oracles via a paired-question protocol," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019, pp. 337–344.

[8] B. Çelen and S. Kariv, "Distinguishing informational cascades from herd behavior in the laboratory," *American Economic Review*, vol. 94(3), pp. 484–498, 2014.

[9] J. Douceur, "The Sybil Attack," in *IPTPS '01 Revised Papers from the First International Workshop on Peer-to-Peer Systems*, 2002, pp. 251–260.

[10] Provable. [Online]. Available: https://provable.xyz/

[11] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town crier: An authenticated data feed for smart contracts," in *Proceedings of the 2016 aCM sIGSAC conference on computer and communications security*. ACM, 2016, pp. 270–282.

[12] V. Costan and S. Devadas, "Intel sgx explained." *IACR Cryptology ePrint Archive*, vol. 2016, no. 086, pp. 1–118, 2016.

[13] S. Ellis, A. Juels, and S. Nazarov. (2017) Chainlink a decentralized oracle network. [Online]. Available: https://link.smartcontract.com/whitepaper

[14] H. Ritzdorf, K. Wust, A. Gervais, G. Felley, and A. Juels, "Tls-n: Non-repudiation over tls enabling ubiquitous content signing," in *Network and Distributed System Security Symposium (NDSS)*, 2018.

[15] J. Guarnizo and P. Szalachowski, "PDFS: Practical Data Feed Service for Smart Contracts," in *24th edition of ESORICS*, 2019.

[16] F. Zhang, S. K. D. Maram, H. Malvai, S. Goldfeder, and A. Juels, "DECO: Liberating Web Data Using Decentralized Oracles for TLS," in *24th edition of ESORICS*, 2019.

[17] J. Adler, R. Berryhill, A. Veneris, Z. Poulos, N. Veira, and A. Kastania, "ASTRAEA: A Decentralized Blockchain Oracle," in *Proceedings of the 2018 IEEE Conference on Blockchain*, 2018, pp. 1145–1152.

[18] R. Kamiya. (2018) Shintaku: An end-to-end-decentralized general-purpose blockchain oracle system. [Online]. Available: https://gitlab.com/shintaku-group/paper/raw/master/shintaku.pdf

[19] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[20] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, vol. 310, 2016.

[21] D. Prelec, "A bayesian truth serum for subjective data," *Science*, vol. 306(5695), p. 462–466, 2004.

[22] J. Witkowski and D. C. Parkes, "A robust bayesian truth serum for small populations," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI, 2014, pp. 1492–1498.

[23] G. Radanovic and B. Faltings, "A robust bayesian truth serum for non-binary signals," in *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI'13)*, 2013.

[24] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 2nd ed. San Mateo, CA, USA: Morgan Kaufmann, 1988.

[25] J. Harsanyi, "Games with incomplete information played by 'bayesian' players, part iii. the basic probability distribution of the game," *Management Science*, vol. 14, no. 7, p. 486–502, 1968.

[26] R. Selten, "Axiomatic characterization of the quadratic scoring rule," *Experimental Economics*, vol. 1, p. 43–62, 1998.

[27] randao.org. (2017) Randao: Verifiable random number generation. [Online]. Available: https://randao.org/whitepaper/Randao_v0.85_en.pdf

[28] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch, "Verifiable delay functions," *IACR Cryptology ePrint Archive*, vol. 2018, p. 601, 2018.