Draft (November 2018): A Smart Contract Oracle for Approximating Real-world, Real Number Values Using Kleros

William George and Clément Lesaege

Abstract

We present an architecture that makes use of the Kleros dispute resolution system, specifically in the case where the possible outcomes that this system can rule between are binary, as the basis for an oracle that approximates a value in a dense totally ordered set, such as $\mathbb{R}$. We imagine this to be particularly applicable as a price oracle. Then, we provide an analysis of the number of applications of the Kleros system required, the cost to an attacker to delay the system by forcing additional applications of Kleros, and the precision that the oracle ultimately obtains.

1 Introduction

A key challenge of smart contract systems is the fact that many useful contracts require access to information that does not natively live on the blockchain. While miners can verify the value of a hash or the validity of a digital signature, they cannot determine who won an election, whether there is a flood in Paris, or even what is the price of ether in US dollars, even though this information might be necessary to execute prediction market, insurance, or financial contracts respectively [5] [6] [7]. Already in the Ethereum white paper [1], the need for oracles is discussed as necessary to overcome the limitation of "value-blindness" of such financial contracts towards crypto-assets.

We propose a system based on Kleros. Kleros is a smart contract based dispute resolution mechanism that relies on randomly selecting tokenholders of the Kleros token pinakion (PNK) to serve as jurors for a given dispute. Then a system of rewards and punishments compensates these jurors for their work in examining the dispute and incentivizes them to vote honestly, see [9]. As such, Kleros allows one to bring real-world knowledge - which side of a given dispute is correct - onto the blockchain. At current development, Kleros is capable of being used for disputes where the jurors are choosing between two possible outcomes, though more complicated structures will be a subject of future work.

In this paper, we will discuss how Kleros can be used to estimate the value of a real-world quantity that is in a dense totally ordered set (namely a totally ordered set such that if x and y are in the set, there exists some z in the set

such that $x < z < y$). In particular, this proposal can be used to estimate real number valued quantities, such as required for a price oracle. We will analyze this proposal, seeing that the precision of the output is tunable to user needs while the number of applications of Kleros is reasonably bounded in terms of resources of attacking or incorrect parties.

2 Related work

There are several existing proposals for cryptographic oracles. Notably Maker DAO [10] requires oracles for the price of ether in USD to guarantee that enough ether is in escrow to collateralize the DAI in circulation. This oracle is a median of values provided by trusted authorities such as leading exchanges, which are chosen (and can be replaced) by MKR token holders. Thus, this oracle is decentralized in the sense that it is ultimately responsive to token holders, albeit via a delegated system. It is worthwhile to note that, in this case, MKR holders have a strong incentive to choose good oracles as the quality of the oracle directly affects whether MKR holders must become the collaterization of last resort for DAI; generalizing this idea to other oracles is not obvious. Also based on a delegated decentralization is the model of ChainLink [8].

More general, and perhaps more akin to our system, is the system used to provide responses to Augur prediction markets [11]. Augur provides oracles for these prediction markets by allowing holders of the Augur token REP to dispute responses provided by default responders designated by the creator of the market. Also, this system is used to obtain the price of the Augur token REP in order to determine if the fees should be adjusted up or down in order to perform a "market cap nudge" so that buying a majority of the REP to perform a 51% attack is financially not worthwhile relative to the amount of value at stake in Augur prediction markets [11].

Similar to Augur, Gnosis [3] offers a prediction market system based on oracles. Gnosis is "oracle agnostic," meaning that a Gnosis prediction market contract can reference any oracle [4]. That said, the Gnosis team has developed oracles themselves which can be used for their markets, such as their "ultimate oracle" mechanism [2]. Gnosis allows particularly for prediction markets based on scalar values such as prices, based on predictions of whether the price will rise or fall [4].

Of particular note when comparing to our system are existing oracles based on the concept of Schelling points [12], such as those were submitters to the oracle are penalized if their submission falls outside of the 25th to 75th percentile range of all submissions, with the design to encourage coherence around the true result [13]. As already discussed in [13], this first version of a Schelling point based oracle is potentially vulnerable to "micro-cheating" as submitters risk little penalty if they provide small variations on the true value, as long as this variation is not too extreme. Such "micro-cheating" may an attack to nudge the output of the oracle and potentially affect the decision making of other submitters, leading to more substantially deviations over time. The current

proposal, as it is based on Kleros [9] which also uses a Schelling point based mechanism, uses similar ideas, but we will see that it is less vulnerable to such "micro-cheating attacks."

3 Proposal

3.1 Assumptions on Kleros and discussion of actors

For this paper, we will assume that Kleros can be used to decide binary disputes. Namely, given two statements $\mathcal{A}$ and $\mathcal{B}$, one of which corresponds to reality, Kleros will return one of the two results, after a delay of some number of Ethereum blocks and in exchange for paying arbitration fees to the jurors who make this judgment. This decision can then be appealed some (limited) number of times by invested parties resulting in an increased delay for the final result and additional arbitration fees. In some of our results we will assume that this mechanism works in an idealized way; namely, that the jurors always decide on the option that actually corresponds to reality. In other of the results discussed below this assumption will not be necessary.

As this schema builds upon Kleros, an important set of actors here will be the Kleros jurors. As usual [9], they will be randomly selected based on the PNK they have staked in the appropriate subcourts, they will be compensated for their work arriving at a decision in arbitration fees denominated in ETH, and they will be encouraged to be coherent with the results of the other jurors to avoid penalties where they lost their staked PNK.

Also, in this proposal we will have the additional role of respondents. Respondents will provide information about the value to be estimated $v \in \mathcal{S}$, where $\mathcal{S}$ is in a dense totally ordered set (such as $\mathbb{R}$), as described below. Respondents pay a deposit which they risk losing if the information provided is ultimately judged to be incorrect, see Section 4. Moreover, respondents will play the role typically played by the parties in a Kleros dispute, where they (in this case automatically) raise disputes against each other, and then pay arbitration fees which compensate the jurors.

In this work, we will consider attacks from attackers that have the capacities of respondents. Namely, we will analyze attacks that submit malicious responses or call appeals in a hostile manner. Attacks aimed at altering the decisions of the jurors will be considered in other work that considers Kleros more generally and only briefly touched on here. (Indeed, for the results in this paper where we make idealized hypotheses about the decisions of the Kleros jurors, such attacks fail by assumption.) Furthermore, we do not consider attacks on the underlying infrastructure of the smart contract platform, such as 51% attacks or network denial-of-service attacks on Ethereum.

We will describe how to use the Kleros dispute resolution mechanism as an oracle to determine an estimate of the value based on information provided by the respondents.

3.2 Kleros-based oracle algorithm

The procedure we propose to approximate the true value of some quantity is based on a sort of modified binary search of the responses, where, rather than split the list of responses at the median when performing a comparison, we detect incoherences that prevent a consensus answer from being accepted and then take a comparison with respect to the median of the list of these incoherences. We are performing these operations on elements in $\mathcal{S}$, which a priori is not closed under averaging, so the normal median may not be defined. However, if we need to take the median of a set D with an even number of elements, namely in the case that requires computing an average, as $\mathcal{S}$ is a dense totally ordered set, we can find some (not necessarily unique) element z of $\mathcal{S}$ such that half of the elements of D are on either side of z. We suppose that for a given $\mathcal{S}$ one has some way of efficiently choosing such a z, and we consider it to be a median of D. In the remainder of this paper, in the context of generic dense totally ordered sets, we use the words median and average in this sense. In the case $\mathcal{S} = \mathbb{R}$, we take the normal median.

In detail, we consider the following:

Algorithm 1. *Input: Each respondent $\mathcal{USR}_i$ submits two distinct values - a lower bound $l_i \in \mathcal{S}$ and an upper bound $u_i \in \mathcal{S}$, $l_i < u_i$, giving an interval (l_i, u_i) in which this respondent believes the true value of the question is located.*

- *Sort the lower bound responses into a list $\mathcal{L}$ and the upper bound responses into a list $\mathcal{U}$, where in each case identical values are considered single elements.*
- *Compute the lists*

$$\mathcal{L}_0 = \{l_i \in \mathcal{L} : \exists u_j \in \mathcal{U} \text{ with } u_j \leq l_i, \ \nexists\, l_k \in [u_j, l_i) \cap \mathcal{L}\}$$

  and

$$\mathcal{U}_0 = \{u_i \in \mathcal{U} : \exists l_j \in \mathcal{L} \text{ with } l_j \geq u_i, \ \nexists\, u_k \in (u_i, l_j] \cap \mathcal{U}\}\,.$$

- *Compute*

$$\mathcal{C}_0 = \{median(l_i, u_j) : l_i \in \mathcal{L}_0,\, u_j \in \mathcal{U}_0 \text{ with } u_j \leq l_i, \ \nexists\, l_k \in [u_j, l_i) \cap \mathcal{L}, \ \nexists\, u_k \in (u_j, l_i] \cap \mathcal{U}\}$$

  (So, if we considered $\mathcal{L}$ and $\mathcal{U}$ in the same line, essentially $\mathcal{L}_0$ would consist of lower bounds which have an upper bound to their immediate left and $\mathcal{U}_0$ would consist of upper bounds that have a lower bound to their immediate right. Then $\mathcal{C}$ consists of the midpoints between each of these pairs.)

- *If $\mathcal{C}_0 \neq \emptyset$*
  - *For each $z \in \mathcal{C}_0$ perform the following in parallel:*

* *Ask Kleros jurors if*

$$\textit{desired value} \leq z$$

or

$$\textit{desired value} > z.$$

* *Allow appeals of their decision as necessary following the fee structures of [14], where here the two sides of the dispute are as follows:*

$$\textit{desired value} \leq z$$

or

$$\textit{desired value} > z$$

  - *If one side of the dispute pays its required fees but not the other, the arbitrator is considered to rule in favor of the side that paid its fees.*
  - *If neither side pays its fees, the previous ruling stands.*

– *Take* $\mathcal{C}_1 = \mathcal{C}_0$.

– *While* $\mathcal{C}_1 \neq \emptyset$

* *If* $\#\mathcal{C}_1$ *is odd, calculate* $m = \textit{median}(\mathcal{C}_1) \in \mathcal{C}_1$. *If* $\#\mathcal{C}_1$ *is even and positive, choose one of the two middle-most values of* $\mathcal{C}_1$ *as* m *in some predictable way (such as by always taking the value on the left).*
* *Eliminate all* l_i *and* u_i *that are on the wrong side of what the jurors decided with respect to* m *from* $\mathcal{L}$ *and* $\mathcal{U}$.
* *Add* m *to* $\mathcal{L}$ *if the jurors have ruled that the true value is higher than* m, *and add* m *to* $\mathcal{U}$ *otherwise.*
* *(Re)calculate*

$$\mathcal{C}_1 = \{\textit{median}(l_i, u_j) : l_i \in \mathcal{L}_0,\ u_j \in \mathcal{U}_0 \textit{ with } u_j \leq l_i,\ \nexists\ l_k \in [u_j, l_i) \cap \mathcal{L},\ \nexists\ u_k \in (u_j, l_i] \cap \mathcal{U}\}$$

based on the updated $\mathcal{L}_0$ *and* $\mathcal{U}_0$.

Output the average of the largest remaining element of $\mathcal{L}$ *and the smallest remaining element of* $\mathcal{U}$.

We will structure the submitter deposits to be large enough to cover the initial round of all of the disputes for required by the for loop, namely every point in $\mathcal{C}_0$. This will require that the submitter deposit d be greater than or equal to f, the total amount of arbitration fees required to pay the jurors in the first round of each dispute. In contrast, the arbitration fees for any appeals are submitted independently from the submitter deposits, typically by parties interested in winning the stake of the other side.

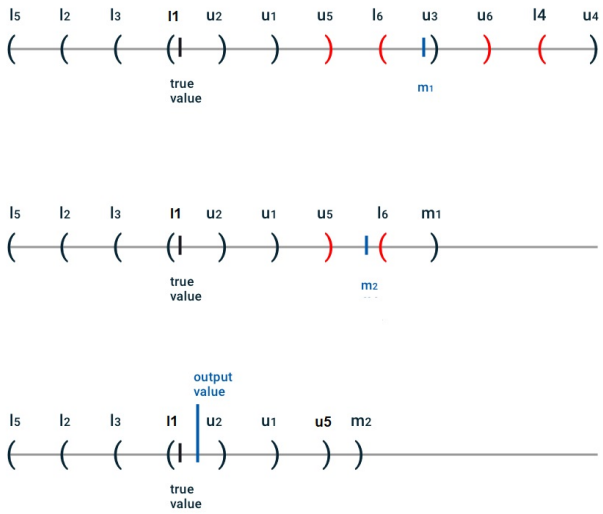

Figure 1: Image needs to be changed to account for fact that m_1 is calculated differently in even number of incoherences cases. An example of Algorithm 1. Six respondents $\mathcal{USR}_i$ each submit (l_i, u_i). Two calls to Kleros are required to determine the output, where $m_i = \text{median}(\mathcal{C}_0)$ in the ith round. In each round the elements in $\mathcal{C}_0$ are marked in red. The execution shown corresponds to the jurors ruling that the true value v is such that $v \leq m_1$ in round one and $v \leq m_2$ in round two.

Proposition 1. *Enough fees are paid by submitters who are ultimately ruled incorrect to cover the initial round of all disputes. Specifically,*

$$\#\text{ submissions ruled incoherent} \geq \#\mathcal{C}_0 = \#\text{ rulings required}$$

Proof. As each submitter pays a deposit that includes f, the cost of arbitration in an initial round, there is $f \cdot (\#$ submissions ruled incoherent$)$ available to cover the fees of the total initial round of disputes. By construction, there is a ruling with respect to each point in $\mathcal{C}_0$. So it suffices to prove the first inequality.

Take $c \in \#\mathcal{C}_0$ such that $c \geq v$, where v is the ultimate output of the oracle. For each such c there are some $l_i \in \mathcal{L}_0$, $u_j \in \mathcal{U}_0$ such that $u_j \leq c \leq l_i$ and there is no $l_k \in (u_j, l_i) \cap \mathcal{L}$. Then l_i was the lower bound of an incoherent submission.

We claim that this process produces a distinct l_i for each $c \in \mathcal{C}_0$. Indeed, if c_1, $c_2 \in \mathcal{C}_0$ as above with $u_{j,1} \leq c_1 \leq l_{i,1}$ and $u_{j,2} \leq c_2 \leq l_{i,2}$ but $l_{i,1} = l_{i,2}$, then either

- $u_{j,1} \in (u_{j,2}, l_{i,2}]$ which contradicts the definition of $\mathcal{C}_0$
- $u_{j,2} \in (u_{j,1}, l_{i,1}]$ which again contradicts the definition of $\mathcal{C}_0$ or
- $u_{j,1} = u_{j,2}$ which, as $l_{i,1} = l_{i,2}$, implies that $c_1 = c_2$.

Similarly, for each element of $\#\mathcal{C}$ less than v there corresponds some $u_j \in \mathcal{U}_0$ that is the upper bound of an incoherent submission.

□

We see that all of the juror rulings that are necessary to evaluate the while loop were, in fact, decided.

Proposition 2. *During the while loop, each $\mathcal{C}_1$ that is computed is a subset of $\mathcal{C}_0$. Hence, for each m computed during the loop, jurors will have either*

- *ruled desired value $\leq m$ or*
- *desired value $> m$*

Proof. The sets $\mathcal{L}$ and $\mathcal{U}$ are only modified during the while loop. There, if the jurors rule that desired value $\leq m$ all elements greater than or equal to m are eliminated from $\mathcal{L}$ and $\mathcal{U}$ and m is added to $\mathcal{U}$. Due to the local way that $\mathcal{C}_1$ is defined, as the elements of $\mathcal{L}$ and $\mathcal{U}$ less than m remain unchanged, the only way a new element could be added to $\mathcal{C}_0$ is if there existed some $l_i \in \mathcal{L}_0$ such that $\text{median}(l_i, m) \in \mathcal{C}_0$. However, when computing $\mathcal{C}_1$, m is a strict upper bound for $\mathcal{L}$, so there will not exist any such l_i. A similar argument applies if the jurors ruled that desired value $> m$.

□

Remark 1. *Note that is possible that jurors in different disputes will rule in ways that are incoherent; e.g. a panel of jurors will rule that $v \leq m_1$ and some other panel of jurors will rule that $v > m_2$ even if $m_2 \geq m_1$. This*

does not prevent the algorithm from halting as the while loop gives priority to the decisions required along the path of a binary search. This gives a potential attacker an incentive to focus her attack on specific disputes that occur near the beginning of that search; however in alternative models where the the disputes are resolved in series rather than in parallel by performing such a binary search, there is already the same incentive for attacker to target the early disputes.

Remark 2. *At the expense of additional gas, after each appeal round in algorithm 1, one can test whether the required disputes for the while loop to terminate have been finalized, i.e. have not been appealed. The appealed disputes should still be arbitrated to allow for the correct redistribution of PNK to participating jurors, but they need not unnecessarily delay the finalization of the result of the oracle.*

A priori, it is conceivable that at some point of the algorithm $\mathcal{L}$ and $\mathcal{U}$ become empty and the last step of the algorithm fails. We see that this cannot, in fact, occur.

Proposition 3. *Suppose that there is at least one submission (l_*, u_*) to the oracle. Then if Algorithm 1 halts, it returns a value in $\mathcal{S}$.*

Proof. We will see that neither $\mathcal{L}$ nor $\mathcal{U}$ becomes the empty set. Then, in particular, the last step of Algorithm 1 is well-defined because there *is* a largest remaining element in $\mathcal{L}$ and a smallest remaining element of $\mathcal{U}$.

Let $l \in \mathcal{L}$, which we have assumed to be non-empty at the beginning of the algorithm. This element l is only later removed from $\mathcal{L}$ if l is on the "wrong side" of some m compared to the decision of Kleros jurors. Namely, if in some round of the while loop

- if $l \leq m$ and jurors rule that $v > m$ or
- if $l > m$ and jurors rule that $v \leq m$.

In the first case, m replaces l as an element of $\mathcal{L}$. In the second case, by the way in which m is calculated, there must have been some other element $l_0 \in \mathcal{L}$ such that $l_0 \leq m$, and this element l_0 is not eliminated in that round of the while loop. Hence $\mathcal{L}$ never becomes the empty set.

A similar argument shows $\mathcal{U} \neq \emptyset$.

□

Moreover, thinking of $\mathcal{C}_1$ as a set of inconsistencies that prevents the oracle from finding an answer that is a consensus among the different responses, we see that the number of these inconsistencies is reduced by half after each round of the outer while loop.

Lemma 1. *Each round of the outer while loop reduces the length of $\mathcal{C}_1$ by at least half.*

Proof. Consider a given round of the while loop. Denote $n = \#\mathcal{C}_1$ in this round. Note that the addition of m to $\mathcal{L}$ or $\mathcal{U}$ cannot create a new element of $\mathcal{C}_1$, as it is either the smallest element of $\mathcal{L}$ or the largest element of $\mathcal{U}$.

If n is odd, then $\frac{n-1}{2}$ elements in $\mathcal{C}_1$ are on either side of m. Hence at least this many elements are eliminated, whichever way the jury rules. Moreover, m itself is given as the median of an upper bound u_i to the left and a lower bound l_j to the right. Depending on the ruling of the jurors, either u_i or l_j is eliminated and its position is replaced by m, which we have seen cannot be in $\mathcal{C}_1$ in the subsequent round. So m is also removed from $\mathcal{C}_1$.

If $n = 2k$ is even, then one of the two most central elements of $\mathcal{C}_1$ is used as m. Thus, on one side of m there will be $k-1$ elements of $\mathcal{C}_1$ and on the other k. Hence there are at least k elements that are eliminated by the choice of the jurors, in addition to m itself which again is also removed from $\mathcal{C}_1$.

□

Lemma 1 has the immediate consequence that:

Corollary 1. *Algorithm 1 halts.*

4 Incentivizing respondents to submit short intervals

For the moment we imagine that there is no up-front cost paid by the party that requires the oracle, and hence respondents are not, on average compensated. Indeed, while there may be fee insurers that are motivated by the incentive structures described in [14] to pay arbitration fees despite not being respondents, we expect that most of the arbitration fees will be paid by the respondents themselves. Then, while some may make gains from paying fees for the position that is ultimately coherent and winning stake from the opposing side's fees, the amount that must be paid to jurors will collectively mostly come from the respondents.

In the event that there are many parties with an interest in the result of the oracle, it is nonetheless not unreasonable to expect submissions. Effective models for Askers paying a fee that compensates respondents is a subject for future work.

Thus, instead, respondents place a deposit D which they risk losing if their response is deemed to be incorrect. These deposits are then redistributed to jurors that submitted answers deemed correct. Additionally, in the event of a dispute the jurors pay the arbitration fees as described in the algorithm. So, a juror's only financial incentive (that is internal to the system) to submit honest answers is to obtain rewards drawn from the deposits and appeal fee stake of incorrect respondents.

A priori a user might then want to submit a very large interval such as $(-\infty, \infty)$ in order to be guaranteed to be correct. (Note again, parties with an external financial interest in the result of the oracle may nonetheless want an

incentive to submit useful estimates). We will design the redistribution mechanism so that respondents have an incentive to submit more precise estimates. To do this we will weight their payouts by an inverse exponential of the length of the submitted intervals.

For each user $\mathcal{USR}_i$ who submits an interval $I_i = (l_i, u_i)$, $\text{length}(I_i) = u_i - l_i$, if the ultimate response to the oracle is not in I_i, the user loses his deposit D. If the response is in I_i, the user receives

$$\frac{\#\text{ incorrect responses} \cdot D - \text{cost of first round arbitration}}{\sum_{j \text{ such that } \mathcal{USR}_j \text{ correct}} \alpha^{-\text{length}(I_j)}} \cdot \alpha^{-\text{length}(I_i)},$$

where $\alpha > 1$ is some fixed constant.

As such, the sum of the lost deposits from the incorrect users is equal to the sum of the payouts to the correct users plus the amount required to pay the jurors for their arbitration in the first round of each dispute. Notice that if a respondent submits an infinite length interval such as $I_i = (-\infty, \infty)$, then $\alpha^{-\text{length}(I_i)} = 0$, so he obtains no reward, whereas users who submit more precise intervals obtain a higher reward.

It is a reasonable subject for future experiments to determine how tweaking the weighting of these payoffs, particularly the constant α, would change user behavior to encourage them to submit precise answers for high payoffs, accepting the risk that an overly precise answer risks being ruled incorrect even if answered in good faith.

5 Bounds on time griefing in terms of attacker resources

In this section, we estimate an attacker's ability to delay the oracle as a function of her resources. First note,

Proposition 4. *If all intervals submitted by the respondents are correct, that is to say the true value $v \in I_i$ for all i, then no calls to Kleros are required.*

Proof. If $v \in I_i = (l_i, u_i)$ for all i, then $l_i < v$ for all v. Similarly $v < u_j$ for all j. Hence $\#\mathcal{C}_0 = \emptyset$ and no calls to Kleros are made.

□

Now we consider situations where we have an attacker. All of the disputes of the for loop are performed in parallel, however an attacker can attempt to delay the result of the oracle by appealing one or more of these decisions.

Suppose that each appeal round of Kleros takes t time and all other operations take negligible time. Further suppose that for each appeal the number of jurors doubles and consequently the appeal fees that are paid double (with these fees being refunded to the winning parties and hence ultimately paid by the losing parties). Suppose that the initial cost of appeal fees is A. Denote by R the attacker's financial resources.

Proposition 5. *Suppose that all solutions submitted other than the attackers are ruled correct, namely the output value is in the submitted interval. Then, the maximum number of appeals required is*

$$O_A(\log_2(R)),$$

and hence the maximum amount of time an attacker can delay a result is

$$O_A(t \cdot \log_2(R)),$$

where the implicit constants are allowed to depend on A.

Proof. If the attacker repeatedly appeals a given decision, then the appeal fees for the mth appeal are

$$A \cdot 2^m.$$

So, the total cost of the first m appeals together is

$$\sum_{i=1}^{m} A \cdot 2^m \leq A \cdot 2^{m+1}.$$

Then, even if the attacker uses all of her resources in appealing a single decision, we must have

$$A \cdot 2^{m+1} \leq R \Rightarrow m \leq \frac{1}{A} \log_2(R) - 1 = O_A(\log_2 R).$$

(Note that as the jurors are assumed to be always correct, the attacker will ultimately lose her appeal so she these resources will, in fact, be consumed.)

□

6 Bounds on running time of loops

In Section 5, we examined an attacker's ability to delay the execution of the oracle by forcing additional appeals. In this section, we will examine the effect of one or more incoherent submitters on the running times of the for and the while loops. This is particularly relevant in evaluating the gas costs of the oracle.

As above, we denote the cost of submitting an incorrect solution (l_{attack}, u_{attack}) by D, via a lost deposit when it is determine that the ultimate answer to the oracle is not in the submitted interval. Once again, we denote by R the collective financial resources of submitters who submit intervals that are ultimately ruled to be incoherent, which without loss of generality we can assume to all be controlled by a single attacker.

Proposition 6. *Suppose that all solutions submitted other than the attackers are ruled correct, namely the output value is in the submitted interval. Then there are at most $\frac{R}{D}$ many disputes that must be resolved ruing the for loop. Consequently, the while loop requires at most $\log_2\left(\frac{R}{D}\right)$ rounds.*

Proof. The attacker can only place at most $\frac{R}{D}$ incorrect solutions. So, by Proposition 1,

$$\#\text{ rulings required} \leq R/D.$$

Then, by Lemma 1 at most $\log_2\left(\frac{R}{D}\right)$ rounds of the while loop are required. □

7 User-calibrated precision

In this section, under idealized assumptions about the results produced by Kleros jurors, we study the precision of the output of the oracle. Particularly, we see that it depends on the lengths of the intervals submitted by the respondents.

Proposition 7. *Suppose that the true value that the oracle is trying to determine is v, and that the jurors are always correct in determining whether a value is greater or less than v. Suppose that some respondent submits the interval $I = (l_0^*, u_0^*)$ such that $v \in I$. Furthermore, suppose that this respondent is willing to pay any required appeal fees in $\lceil \log_2(\#\mathcal{C}_0) \rceil$ specifically chosen cases on which he will ultimately be ruled correct. Then the response output by the oracle is in I.*

Proof. Consider all of the disputes that would occur going through the various rounds of the while loop, assuming the jurors always rule that the true value is on the side of v. As this is a single branch of the tree of possible executions of the while loop, there are $\lceil \log_2(\#\mathcal{C}_0) \rceil$ many such disputes. Suppose that the respondent is willing to finance the side of any appeals in each of these disputes consistent with a true value of v (or at least finance whatever portion of these fees is not covered by other parties or insurers). Note, as we have assumed that the jurors will always rule in a way that is consistent with a true value of v, the respondent will win each of these disputes.

Then, suppose we have passed through the while loop k times. We will iteratively define an interval I_k such that

$$\text{ultimate response} \in I_k \subseteq I.$$

We define these intervals as either of the form $I_k = (l_k^*, u_k^*)$ or the form $I_k = (l_k^*, u_k^*]$. We take $I_0 = I = (l_0^*, u_0^*)$ and then for $k > 0$:

$$I_k = \begin{cases} (l_{k-1}^*, m_k] & : \text{ if jurors rule that } v \leq m_k,\ m_k < u_{k-1}^* \\ I_{k-1} & : \text{ if jurors rule } v \leq m_k,\ m_k \geq u_{k-1}^* \\ (m_k, u_{k-1}^*) & : \text{ if jurors rule that } v > m_k,\ m_k \geq l_{k-1}^*,\ I_{k-1}^* = (l_{k-1}^*, u_{k-1}^*) \\ (m_k, u_{k-1}^*] & : \text{ if jurors rule that } v > m_k,\ m_k \geq l_{k-1}^*,\ I_{k-1}^* = (l_{k-1}^*, u_{k-1}^*] \\ I_{k-1} & : \text{ if jurors rule that } v > m_k,\ m_k \leq l_{k-1}^* \end{cases}$$

Note that $v \in I_0 = I$ by assumption, and if $v \in I_{k-1}$ and v is on the correct side of m_k by the honesty of the jurors, $v \in I_k$. By induction, $v \in I_k$ for all k. In

particular, I_k is, in fact, a non-empty interval. Moreover, each $I_k \subseteq I_{k-1} \subseteq I$ by construction.

By construction, for each k, the endpoints of I_k consist of elements of $\mathcal{L}$ and $\mathcal{U}$. As the algorithm halts by Corollary 1, eventually, after w rounds of the while loop, all lower bounds in $\mathcal{L}$ will be (strictly) less than all upper bounds in $\mathcal{U}$. Then as the output is between the highest lower bound and the lowest upper bound,

$$l_j < \text{ultimate response} < u_i, \text{ for all } i, j$$

In particular, the response is (strictly) between l_w^* and u_w^*, so

$$\text{ultimate response} \in I_w \subseteq I$$

as claimed.

□

A consequence of Proposition 7 is that if users require that the oracle output very precise values, they need only submit very precise interval estimates as respondents in which they are nonetheless confident that the true answer lies.

Another consequence of Proposition 7, is that, again under idealized assumptions on Kleros, honest respondents will never be penalized.

Corollary 2. *Suppose that the true value that the oracle is trying to determine is v, and that the jurors are always correct in determining whether a value is greater or less than v. Suppose that some respondent submits the interval $I = (l_0^*, u_0^*)$ such that $v \in I$. Furthermore, suppose that this respondent is willing to pay all required appeal fees in $\lceil \log_2 (\#\mathcal{C}_0) \rceil$ specifically chosen cases on which he will ultimately be ruled correct. Then the user will not lose any appeal fees or deposits.*

Proof. By Proposition 7, the eventual value output by the procedure will be in I, hence the user will not lose his deposit. As discussed in the proof of Proposition 7, in the cases in which the user is assumed to contribute appeal fees (if necessary), he takes positions consistent with a true value of v and hence is always on the winning side.

□

8 Conclusion

We have presented a completely crowd-sourced oracle for values in smart contracts from dense totally ordered sets that we expect to be particularly applicable as a price oracle. This proposal uses the Kleros dispute resolution system, which we only require to be able to handle binary disputes, to obtain information about the real world. Furthermore, the amount of times this system must be called is limited to a reasonable bound in terms of the resources of parties who propose incoherent answers, not calling the system at all if all respondents submit mutually consistent answers. Furthermore, the precision with which the

oracle returns its final answer is tuned to the precision of the most precise correct respondent so that the system can be as precise as its users require it to be.

References

[1] Ethereum white paper: A next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper. Consulted: April 2018.

[2] Gnosis faq. https://gnosis.pm/faq. Consulted: April 2018.

[3] Gnosis whitepaper. https://gnosis.pm/resources/default/pdf/gnosis-whitepaper-DEC2017.pdf. December 2017.

[4] Nadja Beneš. Getting to the core: An overview of our contract architecture. Gnosis blog https://blog.gnosis.pm/getting-to-the-core-4db11a31c35f. August 2017.

[5] Vitalik Buterin. Ethereum and oracles. Ethereum Blog, https://blog.ethereum.org/2014/07/22/ethereum-and-oracles/. July 2014.

[6] Michael del Castillo. Thomson Reuters to power blockchain contracts with experimental service. Coindesk, https://www.coindesk.com/thomson-reuters-power-blockchain-contracts-new-experimental-service/. June 2017.

[7] Jules Dourlens. Oracles, data outside of the blockchain. Ethereum Developers, https://ethereumdev.io/oracles-data-outside-blockchain/. October 2017.

[8] Steve Ellis, Ari Juels, and Sergey Nazarov. ChainLink: A decentralized oracle network. https://link.smartcontract.com/whitepaper. September 2017.

[9] Clément Lesaege and Federico Ast. Kleros: Short paper v1.0.5. https://kleros.io/assets/whitepaper.pdf. January 2018.

[10] The Maker Team. The Dai stablecoin system. https://makerdao.com/whitepaper/DaiDec17WP.pdf. December 2017.

[11] Jack Peterson and Joseph Krug. Augur: a decentralized, open-source platform for prediction markets. https://bravenewcoin.com/assets/Whitepapers/Augur-A-Decentralized-Open-Source-Platform-for-Prediction-Markets.pdf. January 2018.

[12] T.C. Schelling. *The Strategy of Conflict*. Harvard University Press, 1980.

[13] Vitalik Buterin. SchellingCoin: A minimal-trust universal data feed. Ethereum blog: https://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed/. March 2014.

[14] William George. Draft: appeal fees and insurance proposal. https://drive.google.com/file/d/14LvM9GoE40uZefypWuL-hwVn5C3d4n0H/view?usp=sharing.