Draft (May 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].

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 activated 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 activated 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 5. 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.*
- *While $l_i \geq u_j$ for some $l_i \in \mathcal{L}$ and $u_j \in \mathcal{U}$:*
  - *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

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

(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.)

  - *Compute $\mathcal{C}_0 = \mathcal{L}_0 \cup \mathcal{U}_0$.*
  - *Calculate $m = median(\mathcal{C}_0)$.*
  - *While $l_i \geq m \geq u_j$ for some $l_i \in \mathcal{L}$ and $u_j \in \mathcal{U}$*
    - *Set arbitration fees of $f = c + s$, where c is the cost of arbitration that must be paid to jurors and s is some additional stake.*
    - *Require respondents $\mathcal{USR}_i$ such that $l_i \geq m$ to collectively pay f. If insufficient fees are contributed before a fixed time limit, all such $\mathcal{USR}_i$ lose their deposits and these responses are removed.*
    - *Similarly require respondents $\mathcal{USR}_j$ such that $u_j \leq m$ to collectively pay f. If insufficient fees are contributed before a fixed time limit, all such $\mathcal{USR}_j$ lose their deposits and these responses are removed.*
    - *After removal of responses due to failure to pay f, if it is no longer the case that $l_i \geq m \geq u_j$ for some $l_i \in \mathcal{L}$ and $u_j \in \mathcal{U}$*
      - *Refund arbitration fees paid and exit the inner while loop.*
    - *Else*

· *Ask the jurors if*

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

or

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

· *Set* $c_{total} = c_{total} + c$ *and* $s_{total} = s_{total} + s$.

· *In case of appeal (which doubles the number of Kleros jurors), double each of* c *and* s.

· *After all appeals are completed, eliminate all* l_i *and* u_i *that are on the wrong side of what the jurors decide 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.*

– *If call(s) to Kleros were required in the preceding loop*

∗ *Pay to the jurors* c_{total} *which is split from the fees paid by respondents that are incoherent with the ultimate ruling (namely their submitted interval is entirely on the wrong side of* m*) proportionally based on contribution.*

∗ *Pay to the respondents who are coherent with the ultimate ruling* s_{total}*, where the amount they are paid is proportional to the amount of fees they paid. This* s_{total} *is drawn from the fees paid by respondents that are incoherent with the ruling, taken proportionally based on their contribution to the fees.*

∗ *Refund the remaining fees.*

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

It is conceivable that at some point of the algorithm, all respondents (whose responses have not already been eliminated) will be required to pay arbitration fees and that they will all refuse. At this point, $\mathcal{L}$ and/or $\mathcal{U}$ may become empty (though there may be some values added to $\mathcal{L}$ or $\mathcal{U}$ as the median of some $\mathcal{C}_0$ that would remain). In this case, the last step of the algorithm will fail.

On the other hand, assuming at least one respondent is willing to pay arbitration fees, or is not required to pay them, in any given round, we see that the algorithm does, in fact, produce an output.

Proposition 1. *Suppose that for any given round of the inner while loop, where the median of* $\mathcal{C}_0$ *has been computed to be* m_0*, there is some* $\mathcal{USR}_*$ *who submitted the interval* (l_*, u_*) *and has not had this response removed in a previous round due to failure to pay arbitration fees, such that either*

- $m_0 \in (l_*, u_*)$ *(i.e.* $\mathcal{USR}_*$ *is coherent with an eventual juror decision of the current round in either case) or*

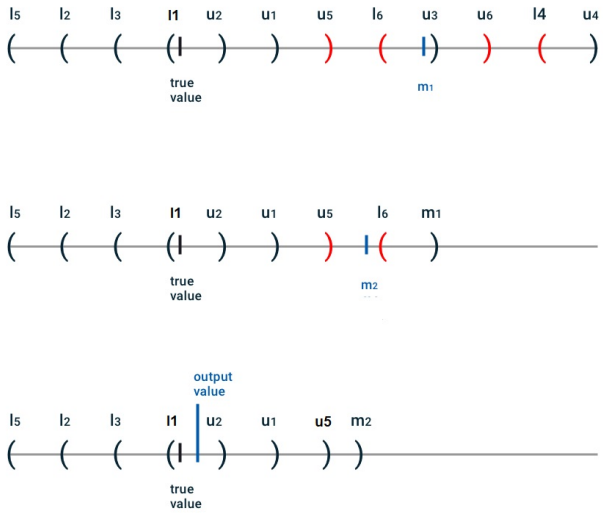

Figure 1: An example of Algorithm 1. Five respondents $\mathcal{USR}_i$ each submit (l_i, u_i). Two calls to Kleros are required to determine the output, where $m_i = \text{median}(\mathcal{C}_0)$ in the ith round. In each round the elements in $\mathcal{C}_0$ are marked in red. The execution shown corresponds 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.

- *$\mathcal{USR}_*$ is in one of the groups of respondents required to pay arbitration fees, and these fees are paid.*

Then if Algorithm 1 halts, it returns a value in $\mathcal{S}$.

Proof. We will see that under these assumptions 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 $\mathcal{U}$.

Note that an element l is only removed from $\mathcal{L}$ if

- l was the lower bound proposed by some user who was removed due to failure to pay arbitration fees or
- l is on the "wrong side" of m compared to the decision of Kleros jurors, where m is the median of $\mathcal{C}_0$ for some round of the while loop.

We see that if ever the jurors rule that the desireed value $v > m_*$, for some m_*, then $\mathcal{L}$ never subsequently becomes empty. In this case, m_* is added to $\mathcal{L}$ and m_* is the smallest remaining element of $\mathcal{L} \cup \mathcal{U}$. Then in later rounds, m_* remains a lower bound because the only elements ever added to $\mathcal{L} \cup \mathcal{U}$ are medians of the $\mathcal{C}_0$'s and as $\mathcal{C}_0 \subseteq \mathcal{L} \cup \mathcal{U}$, these values are themselves lower bounded by m_*. Then, as m_* is not the lower bound proposed by a user, it will never be removed due to a failure to pay arbitration fees. So m_* can only be removed from $\mathcal{L}$ if, in some later round, it is ruled to be on the "wrong side" of the median m_{**} of the $\mathcal{C}_0$ in that round. Then $m_{**} \geq m_*$, so the jurors must have ruled that $v > m_{**}$, and then m_{**} replaces m_* in $\mathcal{L}$.

Then, for the current round of the outer while loop, in order for there to have been a call to Kleros, there must be respondents $\mathcal{USR}_i$ and $\mathcal{USR}_j$ who are not removed for failure to pay arbitration fees such that $l_i \geq m_0 \geq u_j$, with $l_i \in \mathcal{L}$ and $u_j \in \mathcal{U}$. Then,

$$u_j > l_i \geq m_0 \geq u_j > l_j.$$

It is possible that l_j is no longer in $\mathcal{L}$ having previously been eliminated, but as $u_j \in \mathcal{U}$ this cannot be because $\mathcal{USR}_j$ was eliminate for failure to pay arbitration fees in some previous round. Indeed l_j could only have been eliminated without u_j simultaneously being eliminated if at some point jurors decided $v > m_*$ for $u_j \geq m_* \geq l_j$. Then m_* would have been added to $\mathcal{L}$, and by above we know that $\mathcal{L}$ never becomes empty. So we can assume l_j is still in $\mathcal{L}$ in the current round. If the jurors then rule that $v > m_0$, we know by above that $\mathcal{L}$ does not become empty. On the other hand, if the jurors rule that $v \leq m_0$, then l_j is not eliminated in this round.

So the only way that $\mathcal{L}$ can become empty in a given round is if all elements in $\mathcal{L}$ are eliminated without a call to Kleros, namely as a result of all remaining $l \in \mathcal{L}$ corresponding to respondents who refuse to pay arbitration fees. However, by assumption, this is not the case for $\mathcal{USR}_*$.

So it suffices to see that that l_* is still in $\mathcal{L}$ in the current round. We have excluded the possibility that $\mathcal{USR}_*$ was eliminated due to refusal to pay arbitration fees in some previous round. So, in order for l_* to have been eliminated in some previous round, either

- jurors would have had to have ruled that $v > m \geq l_*$ or
- jurors would have had to have ruled that $v \leq m < l_* < u_*$

for some m. In the first case, m would have been added to $\mathcal{L}$, and $\mathcal{L}$ would never subsequently become empty. In the second case m would have become an upper bound on $\mathcal{L} \cup \mathcal{U}$ for all subsequent rounds, so $m_0 \leq m < l_* < u_*$. Hence $m_0 \notin (l_*, u_*)$. So $\mathcal{USR}_*$ must be in one of the groups required to pay arbitration fees in the current round. Being in a group required to pay arbitration fees means that $l_* \in \mathcal{L}$ is such that $l_* \geq m_0$ or $u_* \in \mathcal{U}$ is such that $u_* \leq m_0$. As $m_0 < u_*$, this implies that $l_* \in \mathcal{L}$ in the current round.

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

□

Furthermore, we see that this algorithm has certain basic properties in terms of covering its required costs.

Proposition 2. *In the event of a call to Kleros, sufficient fees are paid by parties that are ruled incorrect to pay the jurors and refund the fees of correct parties.*

Proof. Once m is calculated for a given round of the while loop, Kleros is only called if $l_i \geq m \geq u_j$ for some $l_i \in \mathcal{L}$ and $u_j \in \mathcal{U}$ *after* the removal of responses of users due to refusal to pay arbitration fees. Note that l_i and u_j must correspond to respondents rather than having been added to $\mathcal{L}$ and $\mathcal{U}$ as values of m because by construction when m is added to $\mathcal{L}$, it is a strictly less than all remaining elements in $\mathcal{U}$. Similarly, when m is added to $\mathcal{U}$, it is strictly greater than all remaining elements of $\mathcal{L}$.

As $l_i \geq m$ and $m \geq u_j$, m is in neither (l_i, u_i) nor (l_j, u_j), so both $\mathcal{USR}_i$ and $\mathcal{USR}_j$ will have been have been in groups of respondents required to pay arbitration fees. Moreover, one of $\mathcal{USR}_i$ and $\mathcal{USR}_j$ will then be ruled incorrect, and the contribution of f from the group of this user covers the cost of arbitration. □

Moreover, thinking of $\mathcal{C}_0$ 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 $\mathcal{C}_0$ by at least half.*

Proof. Note that $\mathcal{C}_0$ is a list of elements that are naturally grouped in pairs - a lower bound $l_i \in \mathcal{L}$ which is greater than or equal to an upper bound $u_j \in \mathcal{U}$. Let n be the number of these pairs.

Any round of the while loop that ends in the respondents withdrawing enough responses so that we do not have $l_i \geq m \geq u_j$ for some $l_i \in \mathcal{L}$ and $u_j \in \mathcal{U}$, results in either

- all elements in $\mathcal{L} \cup \mathcal{U}$ greater than or equal to m being upper bounds or
- all elements in $\mathcal{L} \cup \mathcal{U}$ less than or equal to m being lower bounds.

In either event, $\mathcal{C}_0$ is reduced by half, as half of the pairs previously in $\mathcal{C}_0$ were, by the definition of m, on each side of m.

So assume that the round requires a call to Kleros. Note that the addition of m to $\mathcal{L}$ or $\mathcal{U}$ cannot create a new element of $\mathcal{C}_0$, as it is either the smallest element of $\mathcal{L}$ or the largest element of $\mathcal{U}$.

If n is even, by construction, half of the pairs are on each side of m, so half of these are eliminated by the decision by the jurors. If n is odd, then $\frac{n-1}{2}$ pairs are eliminated, and there is one pair that is centered around m with 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 $\mathcal{C}_0$ in the subsequent round. So this pair is also removed from $\mathcal{C}_0$.

□

Lemma 1 has the immediate consequence that:

Corollary 1. *Algorithm 1 halts.*

4 Incentivizing respondents to pay arbitration fees

Each time the median $m = \text{median}(\mathcal{C}_0)$ is computed, there are three groups of respondents:

- respondents $\mathcal{USR}_i$ such that $m \in (l_i, u_i)$
- respondents $\mathcal{USR}_j$ such that $u_j > l_j \geq m$ and
- respondents $\mathcal{USR}_k$ such that $l_k < u_k \leq m$.

Respondents in the first group are coherent with any decision of jurors with respect to whether the value is less than or greater than m. The other two groups are each required to collectively pay arbitration fees of $f = c + s$ lest their responses be deleted and their deposits lost.

This present a "common good" problem where each user $\mathcal{USR}_j$ such that $l_j \geq m$ may want some other respondent to pay the arbitration fees for his group so that he does not lose his deposit, but is unwilling to take the risk of paying these fees himself. However, as respondents who pay these fees and are ruled correct receive an additional payment from the stake portion s of the fees

paid by respondents who are ruled incorrect, respondents have an incentive to participate in this system.

However, even respondents who are generally willing to pay arbitration fees in exchange for ensuring that they do not unnecessarily lose their deposits and for a chance to win the stake posted by the other side can be discouraged by a "bank attack," a general attack on the Kleros appeal system. In this attack, well-funded attackers repeatedly appeal a given decision until honest participants are unwilling or unable to pay their arbitration fees. This could be the case even though arbitration fees are ultimately refunded to the party that is ruled correct in the final appeal. We imagine that a solution to this attack is to make available a sort of "arbitration fee insurance." An insurer offering this policy would cover the arbitration fees of a user that he believed would ultimately prevail. In exchange, the insurer would receive some or all of the stake s that is submitted as part of the fees $f = c + s$ and which ultimately is used to provide a bonus to the party ruled correct at the end of an appeal. A variety of arbitration fee insurers might exist in a given ecosystem, but we specifically imagine that such an insurer could be structured as a DAO, where investors to the DAO can decide which cases to insure.

How large s needs to be relative to c in order to encourage respondents to pay arbitration fees and for such arbitration fee insurers to be viable is a subject for future experiments.

5 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; in fact, any arbitration fees that must be paid to jurors must be paid by the respondents themselves. 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) is to submit honest answers 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

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

where $\alpha > 1$ is some fixed constant. As such, the sum of the payouts to the correct users is equal to the sum of the lost deposits from the incorrect users. 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.

6 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 3. *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. So the outer while loop in the algorithm is never applied and there are no calls to Kleros.

□

Now we consider situations where we have an attacker. Suppose that each call to Kleros takes t time and all other operations take negligible time. Further suppose that for each appeal the number of jurors double 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 and that the cost of submitting an incorrect solution (l_{attack}, u_{attack}) is D, via a lost deposit when it is determine that the ultimate answer to the oracle is not in the submitted interval. Denote by R the attacker's financial resources.

Proposition 4. *Suppose that jurors are always correct in determining a value is greater or less than a true value. Suppose that all solutions submitted other*

than the attackers are correct, namely the true value is in the submitted interval. Then, the maximum number of calls to Kleros required is

$$O_{A,D}(\log_2(R)^2),$$

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

$$O_{A,D}(t \cdot \log_2(R)^2),$$

where the implicit constants are allowed to depend on A and D.

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.)

On the other hand, the attacker can only place at most $\frac{R}{D}$ incorrect solutions. Each such solution can contribute at most four elements to $\mathcal{C}_0$ (each endpoint of the submitted interval can cause at most one incoherence each of which contributes at most two elements to $\mathcal{C}_0$). Then, by Lemma 1 at most $\log_2\left(4\frac{R}{D}\right)$ rounds of the while loop are required, each of which requires at most one call to Kleros plus appeals.

Thus, regardless of the attacker's strategy in dividing her resources between submitting (incorrect) solutions and appealing, she can force at most

$$\log_2\left(4\frac{R}{D}\right) \cdot \left(\frac{1}{A}\log_2(R) - 1\right) = O_{A,D}(\log_2(R)^2)$$

many calls to Kleros which takes

$$O_{A,D}(t \cdot \log_2(R)^2)$$

time.

□

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 5. *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 $v \in I$ and pays all fees required so that this interval is not withdrawn. Then the response output by the oracle is in I.*

Proof. 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 $k > 0$:

$$I_k = \begin{cases} (l_{k-1}^*, m_k] & : \text{ if jurors rule that } v \leq m_k,\ m_k < u_{k-1}^* \\ I_{k-1} & : \text{ if jurors rule } v \leq m_k,\ m_k \geq u_{k-1}^* \\ (m_k, u_{k-1}^*) & : \text{ if jurors rule that } v > m_k,\ m_k \geq l_{k-1}^*,\ I_{k-1}^* = (l_{k-1}^*, u_{k-1}^*) \\ (m_k, u_{k-1}^*] & : \text{ if jurors rule that } v > m_k,\ m_k \geq l_{k-1}^*,\ I_{k-1}^* = (l_{k-1}^*, u_{k-1}^*] \\ I_{k-1} & : \text{ if jurors rule that } v > m_k,\ m_k \leq l_{k-1}^* \\ I_{k-1} & : \text{ if enough responses are withdrawn that there is no} \\ & \quad \text{Kleros call with respect to } m_k \end{cases}$$

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 5 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 5, 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 $v \in I$ and pays all fees required so that this interval is not withdrawn. Then the user will not lose any arbitration fees, appeal fees, or deposits.*

Proof. By Proposition 5, the eventual value output by the procedure will be in I, hence the user will not lose his deposit.

In any given round of the while loop that might result in a call to Kleros, either

- $m \in I$ in which case the respondent is not required to pay arbitration/appeal fees
- $m \notin I$ - the user will be in a group required to collectively pay arbitration/appeal fees. We assume that the user pays these fees if they are not paid by other members of the group. However, as $v \in I$ by assumption, by properties of intervals all of I will be on the correct side of m as ruled by the (assumed correct) jurors. Hence the user will win this dispute and any appeals.

□

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.