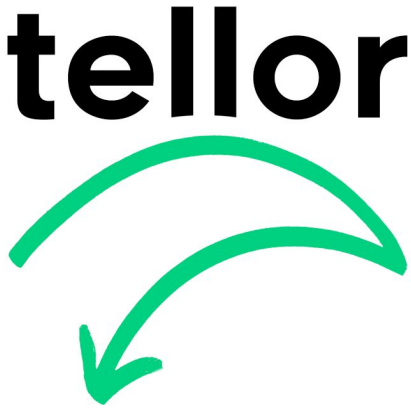


A Decentralized Oracle - Whitepaper

Abstract

Smart contracts on Ethereum are fully self contained and any information or access to the off-chain data is restricted. By creating a system where inputs to a data series are secured by a hybrid Proof-Of-Work and Proof-Of-Stake consensus mechanism, the "Tellor Oracle" allows for trustless access to off-chain information. This paper highlights the structure of this system and gives an in-depth overview on the incentives and assumptions used to ensure an honest input of data to the oracle.

Table of Contents

I. Introduction

Smart contracts on Ethereum are fully self contained and any information or access to off-chain data is restricted. Tellor solves this problem by creating a system where parties can request the value of an API call and miners compete to add this value to an onchain data bank, accessible by all Ethereum smart contracts. Inputs to a data series are secured by a hybrid Proof-Of-Work(PoW)/Proof-Of-Stake(PoS) consensus mechanism. The main Tellor smart contract creates time series of each API call and aims to become the standard source of data for decentralized applications. This oracle, "Tellor", is secured by the same consensus protocol as Bitcoin and utilizes a similar incentives mechanism through the issuance of Tellor's token, Tributes, that are used to request data from the data bank.

II. Background

Blockchain networks, the Ethereum network specifically, allows for fast and secure transfers and creation of digital goods in addition to the storage and execution of tamperproof programs that can manage digital assets.[1] These programs, once deployed on-chain, cannot be changed. They are available to everyone with access to the chain, will execute based on the defined parameters and interactions (transactions), and are verified by the blockchain's consensus mechanism. These characteristics allow anonymous parties to enter into binding digital agreements, or smart contracts. However, because of the redundancy built into the consensus mechanisms of these networks, there is no native vehicle for reading off-chain data (e.g. internet API's). If a smart contract relies on off-chain data to evaluate or execute a function, parties current rely on these options:

- *Manually feed the data to the contract*—the data can be easily compromised and is not a trustless mechanism
- *Trust a centralized party to provide the data*—efficient, but not trustless
- *Rely on a group of trusted known parties (Proof-of-Authority consensus)*—not trustless[2]
- *Proof-of-Stake consensus mechanism (incentivize users to provide data and reach consensus)*—unnecessary for API calls and can lead to long waiting times to reach consensus and confidence on the data can erode if there is a conflict of interest[3]

None of these strategies have proven to be optimal: efficient, trustless and decentralized. Unfortunately, for smart contracts to bring true utility, off-chain data is necessary.

---

[1] www.ethereum.org

[2] MakerDAO's DAI uses a set of 15 known parties to provide data to their DAI contracts.

[3] Some projects that have tried to incentivize their users to provide data are Augur, Gnosis, Aeternity

The **Tellor Oracle** provides an efficient, trustless and decentralized alternative for off-chain data. It provides the infrastructure for decentralized applications to query off-chain data on-chain by properly incentivizing miners to provide data.

III. The Tellor Oracle

The Tellor Oracle is an on-chain data bank where miners compete to add the data points. To create a properly incentivized system we have created a native token, called "Tributes." Parties pay Tellor Tributes to submit a request for an API query to the Oracle. Based upon the reward assigned to each API query, the Oracle selects the best funded query every ten minutes to create a challenge for miners to solve. Each query collects specific data (e.g. ETH/USD or BTC/USD prices) and makes it available on-chain. The Tellor Tributes are used to pay fees to miners, who help add the official data point, and is used to secure the network via a miner staking and data validation voting system. Five submissions are necessary to determine the official data point. Created with a similar structure as 0xBitcoin[4], the Tellor Oracle uses a mineable proof-of-work (PoW) token but along with the PoW solution, miners are also required to provide an off-chain data point. The first five miners to provide the PoW solution and off-chain data point are rewarded with newly minted tokens and the accumulated payout for the specific data request**.** Proof-of-work has proven to be the gold standard for crypto economic consensus mechanisms and Tellor utilizes it within a hybrid model to secure the oracle. In addition to the security provided by the PoW process, we have added an additional layer of security through Proof-of-Stake in which miners are required to stake before they are allowed to mine and risk losing their stake if their submitted values are successfully challenged.

A. Implementation

The Tellor Oracle deploys only one smart contact. It holds and distributes the token supply, informs miners which values to submit, has a built in proof-of-stake methodology for challenges, and holds the historically mined values that contracts can read from.

The contract provides the miners and users the API string, along with necessary fields for data collection and PoW, and allows miners to submit the proof and off-chain data, sorts the values, allows the users to retrieve the values and to bid on which data series is mined next. The contract targets for new values to be mined every 10 minutes. Which data series is mined is determined by which series has the greatest "tip" going to the miners.

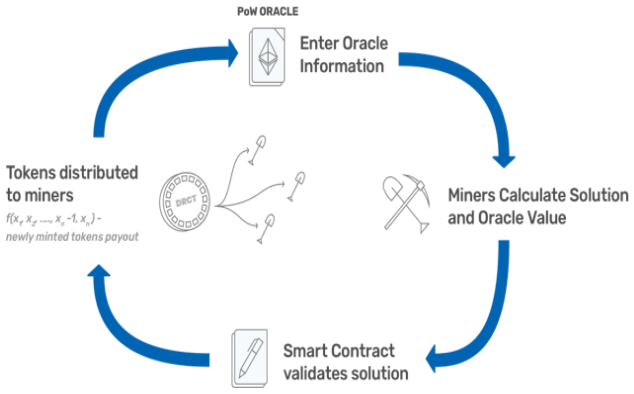
The basic flow for adding and retrieving data is as follows:

1. The user submits a query to the Oracle using Tributes to incentivize miners to choose the query over other submissions.
2. Other users who want the same data pay or 'tip' this data series so miners are further incentivized to mine it.

[4] https://0xbitcoin.org/

3. Every 10 minutes, the Oracle selects the best funded query and provides a new challenge for miners to solve.
4. Miners submit their PoW solution and off-chain data point to the Oracle contract. The Oracle contract sorts the values as they come in and as soon as five values are received, the official value (median of the five) is selected and saved on-chain. The miners are then allocated their payout (base reward and tips).
5. Anyone holding Tellor Tributes can dispute the validity of a mined value within one day of it being mined by "staking" a fee. The Tellor token holders will vote on the validity of the data point and if the data point is deemed to be false, the miner will lose their stake. However, if the vote determines the value is correct, the reporting party's fee is given to the reported miner.

Figure 1: Tellor Oracle



The official value appended to the timeseries is determined by a decentralized mechanism where five values are collected before the winning value is selected. The first five values received are sorted as they are submitted and the miner with the median value is given the highest reward since that will become the 'official' value and the other four miners get a lower reward that decreases the further they are from the median. Once validated and processed the value is available for on-chain contracts to use.

The data collection is decentralized since mining, and by extension data submission, is open to everyone who stakes. To avoid dispersion, incentives are structured to provide the highest reward to the miner that submits the median value. Using the median value instead of the average protects the value from being manipulated by a single party submitting an extreme value.

During the time that the value is being confirmed (one day), parties can challenge this submission. The challenge and data value are put up to vote by Tribute holders. This is described in detail in the Mining and Security section.

B. Incentives

Two types of incentives are implemented in this hybrid model, 1) rewards for PoW submissions and 2) structural incentives to promote accurate value submissions.

Miners are given two types of rewards:

1) A base reward per every successful submission
2) Tips given to miners to incentivize the selection of a query

Miners are incentivized to provide accurate values through 3 processes:

1) Every miner is required to stake 1000 tokens
2) Mining rewards are based upon submission of median value
3) Every accepted value can be challenged and put to vote by all Tellor Tribute holders

Base reward

Similar to the way Ethereum rewards 'Uncles', or miners who were close to winning, the first five miners to submit a PoW and off-chain value are awarded the native Tellor token.

As miners submit the PoW and off-chain value, the values are sorted and as soon as five values are received, the median value and the timestamp are saved to create an on-chain time series. The miner that submits the median value is awarded a larger quantity of the total payout because that becomes the "official value". The current incentive structure leverages game-theory to disincentivize dispersion and adversarial submissions. The miner that reports the medial value is awarded 10 Tributes, the two values closest to are awarded 5, and the two values furthest from it are awarded 1. By rewarding the median with the highest payout and closing down the competition to the first five, miners are incentivized to quickly provide correct data.

If several miners delay their submission to be the fifth submission to ensure being the median value, they risk not receiving a payout at all. The reward will go to the first submission received as ordered by the Ethereum miner.

Figure 2: Tellor Oracle Base Reward Structure

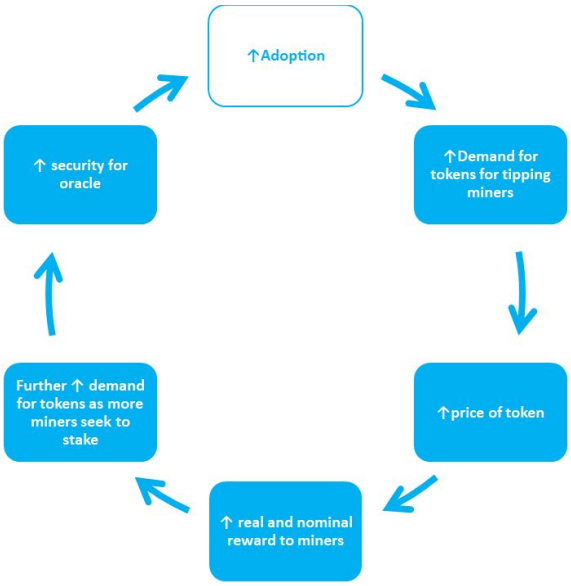
Tips

Users incentivize miners to retrieve their value by posting a bounty to ensure the query they are interested on is mined. Akin to paying a higher gas fee for a prioritized transaction, this is a tip to the miners and is paid out in the same staggered reward structure as the base reward. As the ecosystem

expands, securing data to finalize or execute a contract will lead to higher tips and higher incentivization of miners to continue to mine.

Since the time target of the oracle is 10 minutes, there are only 144 queries per day on average. As the Tellor oracle is adopted, the queue will fill and price competitions will take place. This is a self fulfilling cycle as adoption increases so does demand, miner rewards, security and further adoption.

Figure 3: Tellor Oracle Adoption Cycle



Incentives to submit proper value

- Every miner is required to stake 1000 tokens

Miners have to stake 1000 Tributes to be able to mine. Proof-of-stake allows for economic penalties to miners submitting incorrect values.

- Mining rewards are based upon submission of median value

Uncle rewards can be used to reduce the chance of a miner gaining 51% of hashing power a smart contract pays miners to mine uncles. The Tellor Oracle utilizes uncles by rewarding five miners per data value instead of just one.

- Every accepted value can be challenged and put to vote by any Tellor Tribute token holder

Blockchains are secured via multiple avenues. The first is in the random selection process provided by PoW. The second is that even if the first security measure fails, the blockchain can create forks and different chains until the honest miners win out. Since our oracle system does not have the ability to fork easily, Tellor implements a finality period of one day after original data submissions by allowing parties to challenge data submissions.

C. Mining and Security

Mining

Miners are given the following information from the Tellor oracle contract

- Current Challenge
- API id
- Difficulty
- API to query

The Algorithm

One of the main challenges for a mineable token or any process that relies on mining is the surplus of solo ASICS currently available since if they are used on a small ecosystem these specialized systems can quickly monopolize it. Tellor's proof of work challenge is designed to be different than the Bitcoin mining challenge. This setup requires miners to invest a significant amount of time to update the mining algorithm and should disincentivize miners to become part of the ecosystem too early, allowing Tellor to grow and mature before larger players join.

The code to determine a successful mine for a given challenge and difficulty is:

```
bytes32 _solution = keccak256(abi.encodePacked(currentChallenge,msg.sender,nonce));
bytes32 n = sha256(abi.encodePacked(ripemd160(abi.encodePacked(_solution))));
require(uint(n) % difficulty == 0);
```

The difficulty adjustment is based on the difference between the time target and the time it took to solve the previous challenge. For example, if the time target is 10 minutes and the PoW challenge is submitted in 6 minutes, the difficulty will increase by 4 on the next challenge .

Submission

Miners submitting values must submit the following in order to be a valid submission:

- Successful solution
- Api Id
- Value of referenced query

Security

Security is achieved through the Tellor Oracle's architecture (mining algorithm and selection process for median value) and incentives implemented for miners to promptly submit the correct values (see the "Incentives" section for further details). Ultimate security however is provided by the Proof-of-Stake dispute resolution. Since any value that is disputed will be put to a vote by all token holders, the simple cost to break is:

Token Holder Voting Share * Price of Tributes

This PoW/PoS hybrid model allows for Tellor to take advantage of the efficiency and minimalism of a pure PoW design as well as the final security of PoS. The main problem with PoW consensus mechanisms is that 51% attacks are relatively trivial on smaller chains. The problem with a pure PoS mechanism is that stakers are not properly incentivized to mine (since usually economic punishments are needed) and the general security of negative reinforcement properties do not promote competition in speed and accuracy. Both of these issues are solved through Tellor's hybrid model and the security of the Oracle should suffice for relatively large purposes shortly after launch.

Looking at our formulas, we can summarize that security increases when:

- The share of token holders voting PoS disputes increases
- The price of the token increases
- Demand for the Oracle increases

The Dispute Process

Tellor implements a finality period one day after original data submissions. This allows for any party to challenge data submissions and multiplies the cost to break the network by the number implicit successful confirmations needed (10 blocks without a challenge).

A challenger must submit a stake to each challenge. Once a challenge is submitted, the potentially malicious miner who submitted the value is placed in a locked state for the duration of the vote. For the next week, tribute holders vote on the validity of the mined value. All Tribute holders have an incentive to maintain an honest Oracle and can vote on the dispute. A proper submission is one that corresponds to a valid API query within the time period between the release of the challenge and the submission of the value.

If found guilty, the malicious miner's stake goes to the reporter; otherwise the fee paid by the reporter is given to the wrongly accused miner.

D. Adoption

A rule in distributed systems is that a system is only as decentralized as its least decentralized feature. Therefore any smart contracts or protocols relying on oracles that claim decentralization need a purely decentralized oracle to maintain the integrity of their system. Tellor is building partnerships and working on commitments of use from currently deployed applications and working with these partners to ensure a smooth technical transition to Tellor. On the availability front, Tellor's dev-share and oracle payout structure will provide incentives to early miners to ensure a robust and secure system.

The Tellor team is working hard to ensure ease of use and availability.

To read data, for smart contracts that are currently using a centralized service, the update will be familiar and simple for testing and on mainnet. These are the steps users need to take:

1. npm install tellor

2. On their contracts use “is usingTellor” to access these functions: ***requestData, retreiveData, getLastQuery.***

To request data, users will need Tributes to call this function:

1. ***requestData*** function allows the user to specify the API, timestamp and tip (this can be thought of as “gas”, the higher the tip/payout the higher the probability it will get mined next) for the value they are requesting. If multiple parties are requesting the same data at the same time, their tips are combined to further incentivize miners at that time period and/or API.

To read data, users will need to call these two functions:

1. ***retreiveData*** function allows the user to read the data for the given API and timestamp
2. ***getLastQuery*** function allows the user to read data from the latest API and timestamp mined.

This is an example of a function that would need to be added to a contract so that it can read data from an oracle contact if the contract holds Tributes:

```
contract Oracle is usingTellor {
        ...
        function getLastValue() public returns(uint,bool) {
                (value,ifRetrieve)  = getLastQuery();
                return (value, ifRetreive);
        ...
        }
```

IV. Tellor Tributes

The Tellor Tribute, the native token of the Tellor Oracle, incentivizes miners to provide data, keeps the oracles secure and allows DApps to request and access to on-chain data. The following sections provide an overview of the expected Tribute supply, current price constraints, and uses for the dev share.

A. Supply

The total supply of Tellor is determined by usage and mining rates. For the maximum supply, Tellor’s supply will grow at the rate of the base reward * 144 queries per day. The graph below shows the Tellor supply and the growth rate assuming full utilization:

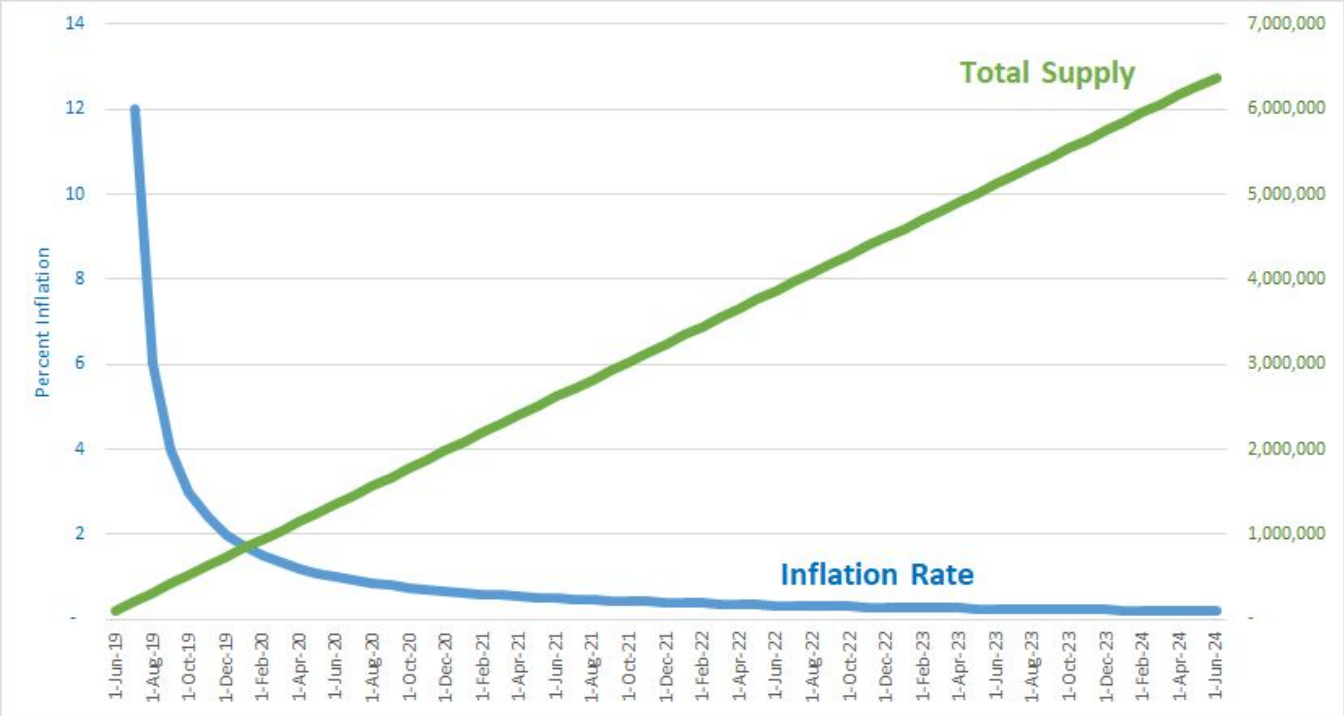


Chart 1: Expected Tellor's annual supply and growth rate (Assuming June 2019 Launch)

B. Price Constraints

Tellor's main competitors are centralized oracle services that can charge low fees for API data retrieval. Tellor is a premium oracle service which provides 144 API queries per day to the applications needing a secure oracle.

The system has several supply and demand constraints affecting the price:

- The security of the system is highly correlated with the price of the token and any monopoly or erosion of confidence in the system will decrease this security and hence the price.
- As the token price increases, less tokens are needed for each query to the miners (assuming demand in USD is stable). This will free supply and decrease the price of the token to an equilibrium level.
- As the price of the token rises, this creates increased security in the Oracle system by further incentivizing miners through the base reward.

C. Dev Share

The Tellor Oracle implements a ten percent dev share. This dev share will be managed by the Daxia team and utilized in the following ways:

- Ensure accurate voting by taking part in PoS challenges
- Create and distribute efficient miners

- Market and Promote the Tellor Oracle to ensure adoption which leads to greater mining incentives
- Create developer tools for utilizing the Tellor Oracle in production deployments
- Fund research and improvements to the oracle

Tellor will use the initial tokens to provide liquidity to users of the oracle. There is no pre-mine or token offering. Tokens will be sold on as needed basis by Tellor to provide liquidity to the users of the oracle. If you want to partner with us to utilize the oracle in your smart contracts, please contact us for details.

V. Potential Applications

Within the context of Ethereum, oracles can be thought of as authoritative sources of off-chain data. These data points allow smart contracts to receive and condition executional instructions. The biggest use case for off-chain data have been ICO's which generally required the ETH/USD price. However, this is highly useful for a wide-array of derivative scenarios.

As Tellor is a contract mechanism that allows oracle data to be derived in a competitive, decentralized manner - we envision a wide array of use cases for this product. Namely:

1. **Exchange-rate data:** interval based exchange-rate values may be used to create trustless financial derivatives
2. **Weather data logs:** for example, we may calculate insurance premium calculation based on a weather forecast
3. **Static/pseudo-static data:** logging and indexing various identifiers, country codes, currency codes
4. **Prediction Market Probability/Odds:** i.e. "What is the likelihood that X event will happen"
5. **Prediction Market Resolution:** i.e. determining who won the most recent presidential election or sporting event
6. **Damage verification:** What were the net total results in damage for insurance contracts
7. **Pseudorandom number generation:** to select a winner in a distributed-lottery smart contract, etc.

Daxia derivatives contracts will be the first to transition to the new oracles, within the next six months.

VI. Future

The Tellor team is already looking toward furthering research in the field of decentralized Oracles. Several solutions have already been identified as potential ways to increase security and speed of the Tellor Oracle.

- Zero-knowledge submissions

Zero-knowledge proofs (ZKP) allow for parties to prove that they know a value without revealing the value.[5] For the Oracle, parties could submit a ZKP for the mining solution along a hidden query value. Once the five parties are chosen as successful miners, they are then required to post the unhidden value (Oracle query) which corresponds to the hidden value submitted with the ZKP.

- TLS Notary Proofs

TLS Notary Proofs give assurances that a website was queried accurately and that no error was returned.[6] The Tellor Oracle has plans to utilize different levels of assurances that can be returned with the query to ensure that miners are accurately reporting data from the requested query.

- Plasma Implementation

One of the big concerns for the future of Tellor is its reliance on mainchain Ethereum transactions. This can get expensive and will be unnecessary in the near future as miners and validators should be able to operate on a trustless off-chain plasma component. Even using a basic Plasma implementation, the mining parties staking to perform the PoW are natural validators for a sidechain and security properties of the Oracle can still hold even if the mainchain contract off loads all mining to the plasma chain. Values, token distribution and staking will be handled by the mainchain, however, these are the more efficient pieces of the current Oracle structure.

- Optimistic Implementation

The implementation of a complementing non-mining oracle system that allows for data submission by any party for the data requests. This complementary system assumes data submitters have the best intentions. This "optimistic" approach can allow for disputes by requiring a PoS and/or can be based on submitter reputation. This implementation would be considered less secure and would cater to projects/Dapps/users that may not be time sensitive and can "shop" around for data.

- Automatic Reporting and monitoring

Off-chain analysis for detecting outliers and reporting these to "gain" the "bad" miner's stake. For example, reporting a value/miner, if the mean differs from median by certain amount.

VII. Conclusion

The Tellor Oracle is already developed and functioning on the Ethereum Rinkeby testnet and is available on Github at https://github.com/DecentralizedDerivatives/MineableOracle.

The Tellor Oracle provides a decentralized option for off-chain data. Tellor plans to continue research on creating a secure, scalable, and on-demand Oracle to help to smart contracts achieve their true

[5] https://en.wikipedia.org/wiki/Zero-knowledge_proof
[6] https://tlsnotary.org/

potential. By creating an oracle schema that uses an incented construct to derive the validity of off-chain data, we:

- **Reduce** the risks associated with single-party oracle providers, who can cut access to API data, forge message data, etc.
- **Build** the foundation for a superior oracle system where data is derived from a distributed set of participants which have both an economic interest and a stake in the validity and success of the oracle data.
- **Create** an effective, secure, and incentivized system for off-chain data which disincentives dispersion and adversarial submissions.

If you are interested in using the Tellor Oracle, contributing to its development, or becoming a miner, please contact us at info@daxia.us.

APPENDIX 1 - SECURITY CONSIDERATIONS

Minimum Cost to Attack Analysis

The cost to successfully attack and break the Tellor oracle is determined by several factors:

1) The price of the oracle Tellor Tributes
2) Average price (in fiat) per query (Demand)
3) Stake amount for mining
4) Voting share of dev team(in tributes)

where:

P = Price of Tellor Tributes
D = Average price (in fiat) per query (Demand)
S = Stake amount in tokens
V = Voting share of dev team(in tributes)

Assuming that miners will only mine up to the reward amount minus a needed premium (assume 10%). The cost of a 51% attack on Tellor:

Cost to 51% attack = Reward to miner

where per Query Reward to winning miner = 4.4 * P + D (4.4 is 22/5 which is average reward assuming no gaming of median function)

Additionally, to 51% attack the network, you would need to gain all ¾ mining rewards to ensure you capture the median value

Cost to 51% attack = 3 * (4.4P + D)

Each miner would also be required to stake tokens (S) in order to mine so:

Cost to 51% attack = 3 (S + 4.4P + D)

Now we have also added the ten confirmations and placing the miners on hold.

Cost to 51% attack = 30(S + 4.4P + D)

This simple analysis though fails to account for the fact that invalid values will never be accepted if reported. So the actual cost to break is:

Cost to attack = max(V * P, 30(S*P + 4.4P + D))