Decentralised Argumentation for Data Vetting in blockchains

Subhasis Thakur[1 [0000−0001−6579−724X]] and John Breslin[1 [0000−0001−5790−050X]]

National University of Ireland, Galway, Ireland
{subhasis.thakur, john.breslin}@nuigalway.ie

Abstract. Although it is difficult to overwrite the data kept in blockchains, there are numerous incidents of false data insertion in blockchains. Current blockchain technologies can not prevent such false insertion into blockchains. In this paper, we present a blockchain model that can prevent such immutable lies. We have several contributions in this paper: we developed a decentralised argumentation protocol that allows auditors to decide the validity of a claim, we developed an incentive system for the auditors not to withheld any evidence for or against a claim, we developed methods to execute the decentralised argumentation protocol in blockchain offline channels for high scale execution of the proposed data vetting method. We prove that the proposed data vetting method executed in the blockchain offline channel network is correct.

Keywords: Data vetting · False data · Blockchain offline channels · Data audit

1 Introduction

Blockchain can prevent data overwriting by acting as a decentralised database. However, such data security is often misrepresented as data correctness. There are numerous examples of false data insertion into blockchains. Supply chain data is one of the most vulnerable for such activity. It is estimated that current supermarkets have on average 33 thousand items including items procedure from 2,400 Kms away [1]. Such an FSC network usually consists of numerous actors. Blockchain can prevent data overwriting but it cannot prevent the inclusion of incorrect information. There are frequent incidents of product mislabelling in supply chains. Studies have revealed a product mislabelling rate of 78% in meat products [1] [12] [14] as these products contains unspecified meat from unknown sources. A study revealed that 16 out of 23 companies have mislabelled products which include traces of donkey meat and GMO organism [8]. The research found chicken and duck were present in red meats and red meat in chicken food items [5]. There are mislabelled organic products [9], wrong third-party certification [13]. Often the mislabelling is caused by products procured outside the EU [4]. Data vetting for supply chains by a centralised entity such as a supermarket chains may not be trustworthy.

[1] https://www.fmi.org/our-research/supermarket-facts

In this paper, we propose a data vetting method for blockchains. We used multi-agent argumentation to develop the data vetting method. Argumentation allows autonomous agents to engage in a turn-taking protocol to prove or disprove a statement. We used IPFS blockchain as a data store for arguments which are the evidence for or against a claim. Our solution allows a group of auditors to engage in argumentation as they access this argument data store and argument attack relation distributed Hash table. We have developed payment schemes for the auditors as they get paid for their auditing work in fair and there is economic incentive to prevent malicious behaviour of the auditors. We implement the decentralised argumentation protocol for data vetting in a blockchain offline channel environment. Our main results are as follows: (1) We developed a decentralised argumentation protocol for data vetting with economic incentives for the auditors to deter them from data withholding or biased argumentation. (2) We have implemented the decentralised argumentation protocol in a blockchain offline channel environment. This allows high scale execution of the proposed data vetting method. The paper is organised as follows: in Section 2 we discuss related literature, in Section 3 we present the decentralised argumentation method for data vetting, and we conclude the paper in Section 4.

2 Related Literature

In [6] argumentation framework was proposed. It included various semantics of valid sets of arguments. [2] extended Dung's argumentation framework to a value-based argumentation framework. [11] proposed method to merge multiple argumentation frameworks for a combined decision-making platform. [15] have explored computational models of argumentation in agent-oriented programming languages. [17] explored Pareto optimality in argumentation frameworks. [10] proposed decision-making methods using multi-agent argumentation. [18] provided game-theoretic modeling of strategic argumentation. [7] explored strategic argumentation methods where an expert withheld arguments. IPFS blockchains was proposed in [3]. We advance the state of the art in argumentation theory and data vetting in the following directions: (1) We have developed a decentralised argumentation protocol that improves current state-of-the-art centralised argumentation protocols. (2) We have developed economic incentives for the auditors to prevent them data withholding and biased argumentation. The state-of-the-art argumentation protocols lack such economic incentives. (3) We have developed a high scale and secure execution model for such decentralised argumentation in a blockchain offline channel environment.

3 Decentralised argumentation for data vetting

An overview of the proposed data vetting solution is shown in Fig. 1. It is as follows: (1) There is an offline channel network created from a proof of work or state based public blockchains. Our solution will also work with any permisssioned blockchain network where Hashed Timed Locked contracts or similar

scripts can be executed. (2) The auditors and the principles (who issue data vetting jobs) are part of this blockchain offline channel network. These entities are also participant of a peer to peer data network. (3) The peer to peer data network will host the argument graph data. (4) Argument graph data will be stored in this peer to peer data network by a verified set of data contributors. The principles (who issue the data vetting jobs) will verify the identity of such data contributors. (5) Data contributors will create labelled graph data with various parameters of an application scenario. For example, in case of a perishable supply chain, such parameters include RFID data, weather data, IoT data on products and trucks moving the products, etc. The data contributors will create argument graphs (by identifying the attack relations) and upload it to thee peer to peer data network.

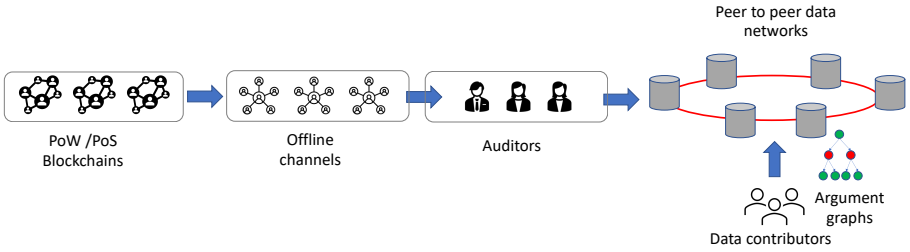


Fig. 1: Overview

In this proposed data vetting solution, our scope for this paper are as follows: (1) We present a formulation of multi-agent argumentation framework to produce argument graphs to be used for data vetting. (2) We present a decentralised argumentation protocol for data vetting using the blockchain offline channel network. (3) We present an incentive model for the auditors (who are vetting the data using this decentralised argumentation protocol).

We assume the following: (1) We assume that the data contributors are either honest or there exists incentives to encourage them to create correct argument graph. (2) We assume that the peer to peer data network can efficiently store and retrieve graph data. (3) In future, we will develop additional solutions to overcome these assumptions.

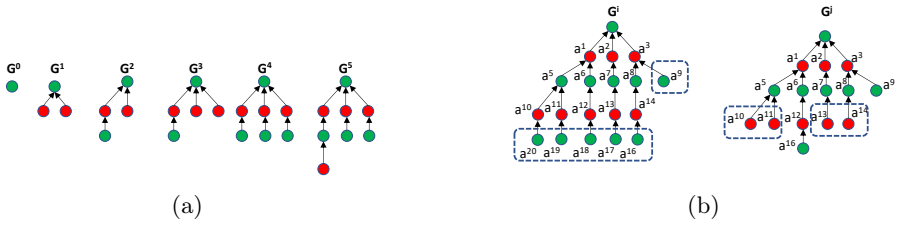
3.1 Argumentation

Argumentation Framework (AF) allows a set of autonomous agents to make decisions from a set of conflicting data or evidence. An AF is represented as a tuple $< A, \succ >$ where A is a set of arguments and $\succ$ is a total order among the arguments. An argument is an abstract information that may be represented as data or evidence. $A_1 \succ A_2$ $(A_1, A_2 \in A)$ means argument A_1 defeats argument

A_2. Given a set of arguments, we can use Dung's argumentation semantics to define the subset of arguments that are valid.

Definition 1. *An argument $A_1 \in A$ is acceptable with respect to a subset of arguments $E \subseteq A$ if E defends A_1, i.e., for any $A_2 \in A$ if $A_2 \succ A_1$ then there exists $A_3 \in E$ such that $A_3 \succ A_2$. A set of arguments is conflict free if it do not contain a pair of arguments A_1, A_2 such that $A_1 \succ A_2$. A set of arguments $E \subseteq A$ is admissible if and only if E is conflict free and all arguments in E are acceptable with respect to E.*

We will use graphs to represent admissible arguments. Given an argumentation framework $(A, \succ)$ and an argument $a^* \in A$ (will be referred to as the root argument), we will sequentially form the admissible set of arguments that includes a^* or prove that such set does not exist. We will sequentially form trees with root as a^*. $G^0 = (a^*, \emptyset)$ will represent the root argument tree with only one argument $a^* \in A$ as the vertex and there is no edge in this graph. A sequence of augmented graphs from G^0 will represent the sequential formation of an admissible set of arguments that includes a^* or represent the failure to do so. $G^1 = (a^* \cup A^1, \succ^1)$ where $A^1 \in A$ is set of arguments that attacks a^* and $\succ^1$ represents edges for such attack relations. $G^2 = (a^* \cup A^1 \cup A^2, \succ^1 \cup \succ^2)$ will represent the augmented argument graph from G^1 such that $A^2 \in A$ attacks any argument in A^1 and $\succ^2 \in \succ$ represents such attack relations. $G^3 = (a^* \cup A^1 \cup A^2 \cup A^3, \succ^1 \cup \succ^2 \cup \succ^3)$ will represent the augmented argument graph from G^2 such that $A^3 \in A$ attacks any argument in $a^* \cup A^2$ and $\succ^3 \in \succ$ represents such attack relations. Hence the argument graph to be created during an even (odd) numbered step can produce arguments to attack arguments presented by graph augmentation during odd-numbered previous steps. In any augmented graph $G^i = (a^* \cup A^1 \cup \cdots \cup A^{i-1}, \succ^1 \cup \succ^2 \cdots \cup \succ)$, a^* is part of an admissible set of arguments if there is no leaf node the argument graph at an odd-numbered distance from the root node.



(a) (b)

Fig. 2: (a) Sequence of argument graphs. Note that the graph G^5 is not admissible, (b) In the argument graph G^i (left) 'all winners' payment scheme can be applied and auditors who provided the leaf nodes will get paid evenly. In the argument graph G^j, 'strict winner' payment scheme can be applied as the auditors who provided $a^{10}, a^{11}, a^{13}, a^{14}$ will be rewarded. In G^j 'chosen strict winner' payment scheme can be used as both 'all winners' and 'strict winners' payment scheme are applicable..

We will use this notion of a sequence of argument graphs to represent the process of decentralised data vetting where each auditor will construct an argument graph to argue for or against the validity of the root argument.

3.2 IPFS as Argument Store

IPFS works as a decentralised database where a peer can access content using the name or identification information of the content rather than the location of the content. IPFS is built on a peer-to-peer network where distributed hash tables are used to store location information of contents. Content will be stored at multiple peers of the network and all peers maintain and synchronize its Hash tables as the location information of the contents. In order to add new content to the IPFS, a peer will generate a unique ID of the content and update its local hash table. Next, it informs its neighbors in the peer-to-peer network about the updated hash tables who will update their hash tables and inform their respective neighbors. This process continues until all local hash tables are synchronized. If a peer wants to access the new content then it will look into the hash table and access the content from the peer who is storing it. Upon accessing it, the peer will have the option to keep a local copy of the content if the peer is acting as a host. Hence multiple copies of the document are stored across the peer-to-peer network. The content file can be split into multiple small files and the content identification information may have links to these small chunks of the content. As the content is stored in multiple locations, it is difficult to censor and overwrite content kept in IPFS.

3.3 Decentralised Argumentation

In the previous section, we describe a sequence of argument graph creation. We will present a protocol for decentralised argumentation protocol for the data vetting which uses such a sequence of argument graphs. It is as follows: Let there are n auditors $(V_1, \ldots, V_n)$ who want to check the validity of data kept in a blockchain. A be the set of all arguments and $\succ$ be the attack relations among the arguments. The sets $A, \succ$ will be kept in an IPFS blockchain as mentioned before. The auditors are arranged in a total order $>$ and they will argue about the validity of an argument in that order. The argumentation protocol is as follows:

1. The auditors will argue about the validity of an argument $a^* \in A$. An argument will be regarded as a valid argument if it is part of an admissible subset of arguments $S \subseteq A$.
2. The auditors will gradually construct the admissible set or fail to do as each of them may construct a new argument graph. The auditors will take turns to create the argument graph using the ordering $>$ among the auditors. Let $>$ is such that $V_1 > V_2 > \cdots > V_n$.
3. V_1 will have the first opportunity to create the first argument graph G^1 from $G^0 = (a^*, \emptyset)$. V_1 will have a fixed finite time to create such an argument

graph. If V_1 creates G^1 (as shown in Fig. 2(a)) then, it will inform all other auditors. If V_1 fails to do so then V_2 will have to opportunity to create G^1 and so on.

4. This process will continue until either the argument graph reaches a critical height or a fixed time duration has passed.
5. Outcome of this protocol is proof of the validity of the root argument as the existence of an argument graph that is admissible and it is not a subgraph of any other argument graph.

The auditors will be paid for their data vetting service. We will evaluate the following payment schemes:
(1) All winners: In this payment scheme auditors who have created any of the leaf nodes of the final argument graph will get paid evenly. (2) Strict winners: In this payment scheme auditors who have created the leaf node (s) of the final argument graph which has an odd-numbered distance from the root node will get paid evenly. (3) Chosen strict winners: In this payment scheme, the payment scheme 'strict winners' will be chosen over 'all winners' in an argument graph where both payment schemes can be applied.

The 'all winner' payment scheme rewards any auditor who has developed an argument that can not be proved wrong. This payment scheme can be used for both cases where there is an admissible argument graph and where there is no admissible argument graph that includes a^* (in this case, there may be leaf nodes both at an odd and even-numbered distance from a^*). Let δ be the total reward for vetting data. It will be distributed evenly among the winners according to these payment schemes.

Let Δ be the total reward for vetting the data with root argument a^*. The amount of reward will gradually increase as the argument graph is built by the auditors. Given an argument graph G^i with maximum distance p from the root a^*, the total reward will be δ^p (p and δ are positive integers and $p, \delta > 1$). δ^p will be evenly distributed among all auditors who have constructed at least one leaf node of G^i. If the auditing for the argument a^* is permitted to be executed for p iterations then $\delta^p \leq \Delta$.

We assume that there is a finite cost to create arguments. Hence auditors may strategically reveal their arguments to minimise their expense. We assume that all auditors are equally likely to construct an argument, find an appropriate argument. Any auditor will choose one of the following strategy when its turn to present arguments:
(1) Reveal: It may choose to create a new set of arguments by constructing a query to the IPFS data store for arguments and by doing so it will incur a certain cost. This strategy will be useful for an auditor if it is not a winner according to the payment scheme and latest argument graph. (2) Withhold: It may choose not to create a new set of arguments to avoid the cost of querying the IPFS data store. It can do so if it is currently getting paid as per the most recent argument graph and payment scheme. (3) Biased: It may choose either to attack or support the root argument.

Lemma 1. *Any auditor is better off with the strategy to reveal arguments for all payment schemes if $\delta > d^p$ where total reward at iteration p is δ^p and d is the average degree of the argument graph.*

Proof. Consider the argument graph formation shown in Fig. 3(c). Let the argument graph (G^{p-1}) till iteration $p-1$ is the most recent and the argument graph (G^p) for the iteration p can be constructed from it. Assume that an auditor is a winner at the current argument graph G^{p-1} however it has an argument to create G^p (at least subset of G^p which is a strict superset of G^{p-1}). The maximum payoff of the auditor for G^{p-1} is δ^{p-1} (in the best-case scenario, the auditor has constructed all arguments in the leaf nodes of G^{p-1}) and the minimum payoff of the same auditor for G^p is δ^p/d^p (in the worst-case scenario, exactly one auditor has constructed argument corresponding to one leaf node of G^p) where d is the average degree of the entire argument graph stored in the IPFS data store. Hence, the auditor will always reveal arguments if the following holds:

$$\delta^{p-1} < \frac{\delta^p}{d^p}, \frac{1}{\delta} < \frac{1}{d^p}, \delta > d^p. \tag{1}$$

This condition is sufficient for the auditors to always reveal for all payment schemes. Case 1: If the 'all winner' payment scheme is chosen then the auditor will get at least δ^p/d^p. Case 2: If 'strict winner' or 'chosen strict winner' scheme is chosen and the auditor can augment G^{p-1} then it will always do so as it will get at least δ^p/d^p.

Lemma 2. *An auditor is better off by not choosing 'bias' strategy.*

Proof. Let the 'bias' is against the root argument. In this case, (a) if the maximally augmented argument graph does not correspond to an admissible set then the arguments against the root argument may pay off. However, any non-biased auditor may also present the same argument against the root argument. Thus 'bias' strategy does not bring more payoff than the 'non-bias' strategy. (b) if the maximally augmented argument graph does correspond to an admissible set then, the 'bias' strategy will have less payoff than the non-bias strategy.

Let the 'bias' is for the root argument. In this case, (a) if the maximally augmented argument graph does not correspond to an admissible set then the arguments for the root argument will never pay off. (b) if the maximally augmented argument graph does correspond to an admissible set then the arguments for the root argument will bring the same payoff with a non-biased strategy.

Theorem 1. *For any argument, the admissible set of arguments will be formed or the failure to do so will be proved.*

Proof. According to Lemma 1 and 2, an argument graph will always be formed at every iteration as it brings more payoff to the auditor who augments the argument graph. Hence the maximal augmented argument graph will be formed irrespective of whether it corresponds to an admissible set of arguments or proof of failure to form an admissible set of arguments.

3.4 Blockchain implementation

In this section we will discuss blockchain implementation of the above-mentioned decentralised argumentation protocol. We will use blockchain offline channels to implement the decentralised argumentation protocol for data vetting. Blockchain offline channel allows two peers to securely transfer tokens without updating the blockchain. We will use the protocol developed in [16]. Briefly, it is as follows (illustrated in Fig. 3(a)):

1. Peers A and B will exchange two sets of Hashes of random strings.
2. A sends the Hash K_A^1 to B and B sends the Hash K_B^1 to A.
3. A will send a Hashed-time locked contract ($HTLC_A$) to B as follows:
   (a) From the multi-signature address M_{AB} 1 token will be given to A and 1 token will be given to another multi-signature address M'_{AB}.
   (b) From the multi-signature address M'_{AB} 1 token will be given to B after time T unless A claims these tokens by revealing the key to hash K_B^1.
4. B will send a Hashed-time locked contract ($HTLC_B$) to A as follows:
   (a) From the multi-signature address M_{AB} 1 token will be given to B and 1 token will be given to another multi-signature address M'_{AB}.
   (b) From the multi-signature address M'_{AB} 1 token will be given to A after time T unless B claims these tokens by revealing the key to hash K_A^1.
5. After exchanging the HTLCs, A and B will fund the multi-signature address M_{AB} by one token. A (B) will include $H_A^1, \dots$ ($H_B^1, \dots$) in the transaction funding M_{AB}.
6. Next, A and B can change the share of the fund in M_{AB} by exchanging new HTLCs. However, they must reveal the key to K_A^1 and K_B^1.

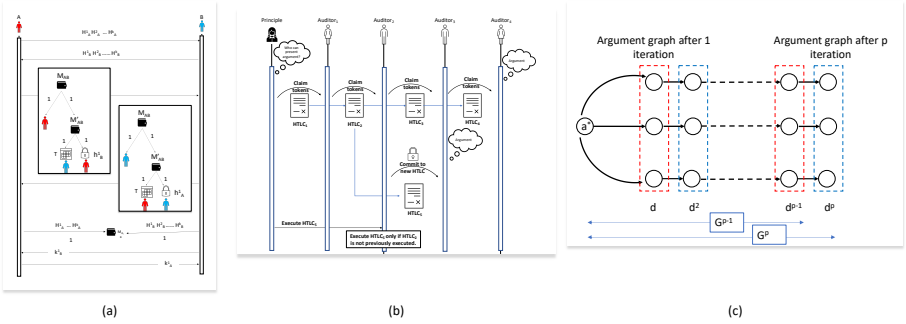


Fig. 3: (a) Protocol to create offline channel between two peers, (b) Explanation of the Decentralised argumentation in offline channels, (c) Evaluation of strategy of the auditors.

We will use an offline channel network created using this protocol to implement the decentralised argumentation protocol. All actors of the data vetting

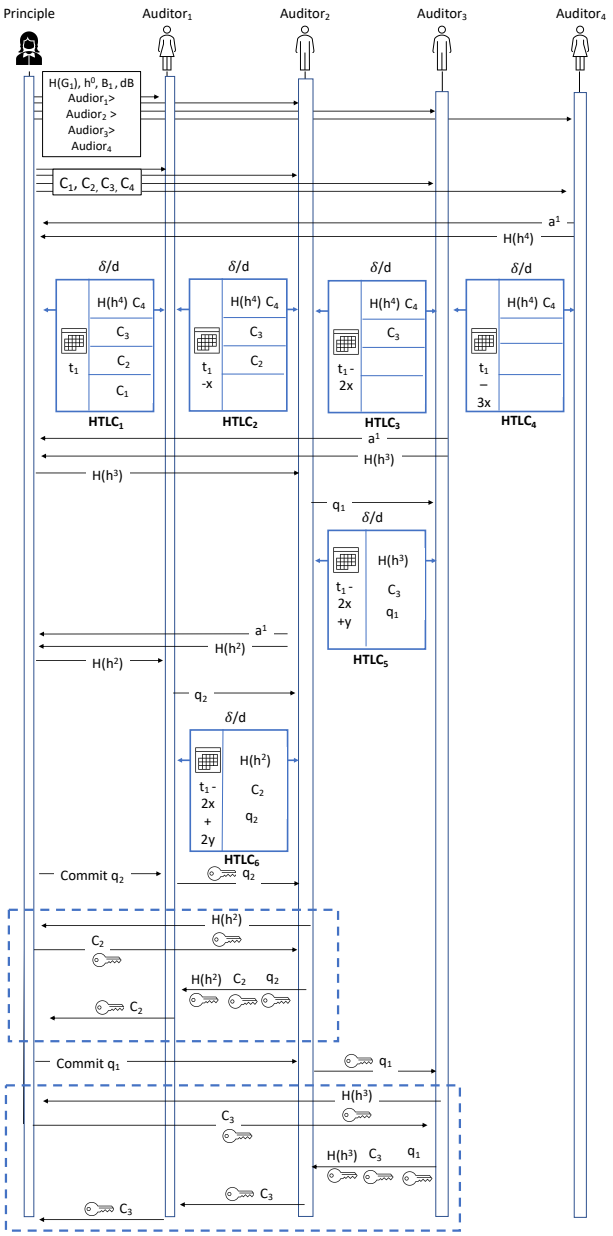


Fig. 4: Decentralised argumentation in offline channels

problem, auditors, principle (who wants to audit) are part of an offline chan-

nel network. We will use the following protocol to implement the decentralised argumentation protocol. It is as follows (shown in Fig. 4):

1. The principle will inform all auditors about the graph data stored in the IPFS data sore as Hash of such a graph ($H(G_1)$), the start time of vetting B_1, time limit of an auditor dB to construct an argument graph, order among the auditors, and the current argument graph h^0.
2. Argument graph h^0 can be augmented by finding attacks on the leaf nodes. If h^0 has p leaf nodes then all such leaf nodes can be attacked at most d times where d is the average degree of the graph G_1. The protocol for augmentation of h^0 for a leaf node a^1 is as follows:
   (a) The principle will allocate reward δ/d for every argument a^1 which can be attacked at this iteration of argument graph formation.
   (b) Let there are 4 auditors $Auditor_1$, $Auditor_2$, $Auditor_3$, and $Auditor_4$. The principle will send 4 random Hashes C_1, C_2, C_3, C_4 to the auditors. Next, it will ask the auditor to augment h^0 for the leaf node a^1.
   (c) Let $Auditor_4$ first finds an argument to attack a^1. It will inform principle by sending Hash of the argument $H(h^4)$.
   (d) Upon receiving the $H(h^4)$ from $Auditor_4$, will form a sequence of HTLCs among all auditors.
   (e) The HTLC between $Auditor_4$ and $Auditor_3$ will state that:
      i. $Auditor_3$ can take δ/d tokens from the multi-signature address between $Auditor_3$ and $Auditor_4$ after time $t_1 - 3x$ if :
      ii. $Auditor_4$ does not claim these tokens by providing keys to $H(h^4)$ and C_4.
   (f) The HTLC between $Auditor_3$ and $Auditor_2$ will state that:
      i. $Auditor_2$ can take δ/d tokens from the multi-signature address between $Auditor_2$ and $Auditor_3$ after time $t_1 - 2x$ if :
      ii. $Auditor_3$ does not claim these tokens by providing keys to $H(h^4)$ and C_4 or by providing key to c_3.
   (g) The HTLC between $Auditor_2$ and $Auditor_1$ will state that:
      i. $Auditor_1$ can take δ/d tokens from the multi-signature address between $Auditor_2$ and $Auditor_1$ after time $t_1 - x$ if :
      ii. $Auditor_2$ does not claim these tokens by providing keys to $H(h^4)$ and C_4 or by providing key to c_3 or by providing key to c_2.
   (h) The HTLC between $Auditor_1$ and $Principle$ will state that:
      i. $Principle$ can take δ/d tokens from the multi-signature address between $Auditor_1$ and $Principle$ after time t_1 if :
      ii. $Auditor_1$ does not claim these tokens by providing keys to $H(h^4)$ and C_4 or by providing key to c_3 or by providing key to c_2 or by providing key to c_1.
   (i) Now, it may happen that, other auditors may have found an attack relation with a^1. Let $Auditor_3$ finds such a relation. It will inform the Principle about Hash of such argument $H(h^3)$.
   (j) Principle will inform $Auditor_2$ about $H(h^3)$ and a new HTLC between ($HTLC_5$) $Auditor_2$ and $Auditor_3$ will be formed. $Auditor_2$ will send a Hash q_1 to $Auditor_3$ as commitment to the new HTLC.

(k) $HTLC_5$ will state that $Auditor_2$ will take δ/d tokens from the multi-signature address between $Auditor_2$ and $Auditor_3$ after time $t_1 - 2x + y$ if $Auditor_3$ does not claim it by revealing key $H(h^3)$, q_1 and C_3.
(l) Similarly $Auditor_2$ may find the argument to attack a^1 and corresponding new HTLC between $Auditor_1$ and $Auditor_2$ will be formed.

3. Now execution of these sequence of HTLCs will be guided by the principle as follows:
   (a) In case, all auditors except $Auditor_4$ fails to provide an argument to augment h^0 then the principle will send key to C_4 to $Auditor_4$ which it will use to claim δ/d tokens from the multi-signature address between $Auditor_4$ and $Auditor_3$. Same keys will be used by $Auditor_3$, $Auditor_2$, $Auditor_1$ to claim δ/d tokens as $HTLC_4$, $HTLC_3$, $HTLC_2$, $HTLC_1$ will be sequentially executed.
   (b) In case, other auditors provide arguments to augment h^0, the principle will choose auditors according to the announced order among the auditors to reveal their arguments against the argument a^1.
   (c) Example: the principle will ask $Auditor_2$ to reveal its argument $H(h^2)$. If h^2 is a correct augmentation of h^0, i.e., there exists a subgraph isomorphism from $h^0 \cup h^2$ to G^1 then it will reveal key to C_2 to $Auditor_2$.
   (d) The principle will ask $Auditor_1$ to reveal 'commitment' q_2 (key to the Hash q_2) to $Auditor_2$. $Auditor_1$ will do so if $HTLC_1$ has not been executed before otherwise, it will lose tokens.
   (e) After $Auditor_1$ reveals key to q_2 to $Auditor_2$, $Auditor_2$ will reveal C_2 and h^2 to $Auditor_1$ and claim δ/d tokens by the $HTLC_6$. $Auditor_1$ will use these keys to claim δ/d tokens from $HTLC_1$.
   (f) If h^2 is not a valid augmentation of h^0 (either h^2 has been revealed before by another auditor or there is no subgraph isomorphism from $h^0 \cup h^2$ to G^1) then the principle will choose $Auditor_3$ to reveal its argument.
   (g) This process will continue until either a successful HTLC execution occurs or attempts is made to execute all HTLCs.
4. The above procedure to argue against the leaf node a^1 will be repeated d times as at most d arguments can be built against a^1.
5. After augmenting the argument graph h^0 for all leaf nodes, the argument data graph kept in the IPFS blockchain can be updated. In such a case, the principle will inform all auditors about the new argument data graph in the IPFS blockchain. The next iteration of the protocol will use such an updated argument data graph.

An explanation of the above described decentralised argumentation protocol is as follows:

1. The principle will ask all auditors to argue against the argument corresponding to a leaf node of the most recent argument graph. An auditor can argue irrespective of the order among auditors.
2. $Auditor_4$ first finds an argument. The principle guides formation of a sequence of HTLCs to transfer funds from itself to $Auditor_4$.
3. Now, $Auditor_3$ finds an argument and it informs the principle.

4. The principle asks $Auditor_2$ to create a new HTLC with $Auditor_3$ which forks from $HTLC_2$.
5. A malicious principle may allow $Auditor_4$ and $Auditor_3$ to claim tokens as two sequence of HTLCs will be execute: ($HTLC_4 \rightarrow HTLC_3 \rightarrow HTLC_2 \rightarrow HTLC_1$) and ($HTLC_5 \rightarrow HTLC_2 \rightarrow HTLC_1$). But it is not possible to execute the sub-sequence of HTLCs $HTLC_2 \rightarrow HTLC_1$ twice. Hence $Auditor_2$ will loose fund as $Auditor_3$ will claim fund twice from $Auditor_2$ and $Auditor_2$ will only claim fund once from $Auditor_1$.
6. To prevent this problem, $Auditor_2$ places a commitment data into $HTLC_5$. It allows execution of $HTLC_5$ only if $Auditor_2$ allows it. Thus $Auditor_2$ will not lose any fund due to double-spending.

Definition 2. *The decentralised argumentation protocol is correct if the following holds: (1) An auditor will not lose tokens by participating in the HTLC-based argumentation protocol. (2) Two auditors will not get paid for the same argument. (3) If two auditors have produced the argument then the auditor with the highest priority according to the order among the auditors will get paid.*

Theorem 2. *The decentralised argumentation protocol proposed in this paper is correct.*

Proof. An auditor will not lose tokens by participating in the HTLC-based argumentation protocol because: (1) The set of HTLCs formed to augment the argument graph h^0 for the leaf node a^1 will form a tree structure. (2) The principle will only reveal the key to one of the Hashes C_1, C_2, C_3, C_4 as it has only assigned δ/d tokens to be transferred via these sequences of HTLCs. Hence HTLCs will be executed according to only one path among the sequence of HTLCs (shown in Fig. 3(b)).

Two auditors will not get paid for the same argument because: (1) As shown in Fig. 3(b), two auditors may get paid if there is a fork from the first sequence of HTLCs. For example, $HTLC_5$ is forked from $HTLC_2$. (2) In such instance of fork, the common auditor ($Auditor_2$ as shown in this figure.) will prevent execution of both HTLC sequences ($HTLC_4 \rightarrow HTLC_3 \rightarrow HTLC_2 \rightarrow HTLC_1$) and ($HTLC_5 \rightarrow HTLC_2 \rightarrow HTLC_1$) by not revealing q_1 to $Auditor_3$ if $Auditor_4$ has claimed tokens.

If two auditors have produced the argument then the auditor with the highest priority according to the order among the auditors will get paid, this is because the time constraint in various HTLCs are used according to the order among the auditors. As shown in Fig. 4, time constraints of HTLCs conform to the order among auditors.

4 Conclusion

In this paper, we proposed a data vetting method using multi-agent argumentation. We presented a blockchain offline channel-based execution model of such a data vetting method. In the future, we will improve IPFS blockchains for faster retrieval of argument data.

Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, co-funded by the European Regional Development Fund.

References

1. AYAZ, Y., AYAZ, N., EROL, I.: Detection of species in meat and meat products using enzyme-linked immunosorbent assay. Journal of Muscle Foods **17**(2), 214–220 (2006). https://doi.org/https://doi.org/10.1111/j.1745-4573.2006.00046.x, https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1745-4573.2006.00046.x
2. Bench-Capon, T.J.M.: Value based argumentation frameworks. CoRR **cs.AI/0207059** (2002), https://arxiv.org/abs/cs/0207059
3. Benet, J.: IPFS - content addressed, versioned, P2P file system. CoRR **abs/1407.3561** (2014), http://arxiv.org/abs/1407.3561
4. Charlebois, S., Sterling, B., Haratifar, S., Naing, S.K.: Comparison of global food traceability regulations and requirements. Comprehensive Reviews in Food Science and Food Safety **13**(5), 1104–1123 (2014). https://doi.org/https://doi.org/10.1111/1541-4337.12101, https://onlinelibrary.wiley.com/doi/abs/10.1111/1541-4337.12101
5. Chuah, L.O., He, X.B., Effarizah, M.E., Syahariza, Z.A., Shamila-Syuhada, A.K., Rusul, G.: Mislabelling of beef and poultry products sold in malaysia. Food Control **62**, 157–164 (2016). https://doi.org/https://doi.org/10.1016/j.foodcont.2015.10.030, https://www.sciencedirect.com/science/article/pii/S0956713515302528
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2), 321–357 (1995). https://doi.org/https://doi.org/10.1016/0004-3702(94)00041-X, https://www.sciencedirect.com/science/article/pii/000437029400041X
7. Dziuda, W.: Strategic argumentation. Journal of Economic Theory **146**(4), 1362–1397 (2011). https://doi.org/https://doi.org/10.1016/j.jet.2011.05.017, https://www.sciencedirect.com/science/article/pii/S0022053111000834
8. Grammenos, A., Paramithiotis, S., Drosinos, E.H., Trafialek, J.: Labeling accuracy and detection of dna sequences originating from gmos in meat products commercially available in greece. LWT **137**, 110420 (2021). https://doi.org/https://doi.org/10.1016/j.lwt.2020.110420, https://www.sciencedirect.com/science/article/pii/S0023643820314080
9. van Hilten, M., Ongena, G., Ravesteijn, P.: Blockchain for organic food traceability: Case studies on drivers and challenges. Frontiers in Blockchain **3**, 43 (2020). https://doi.org/10.3389/fbloc.2020.567175, https://www.frontiersin.org/article/10.3389/fbloc.2020.567175
10. Kakas, A., Moraitis, P.: Argumentation based decision making for autonomous agents. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems. p. 883–890. AAMAS '03, Association for Computing Machinery, New York, NY, USA (2003). https://doi.org/10.1145/860575.860717, https://doi.org/10.1145/860575.860717

11. Leite, L., Alves, T., Alcântara, J.: Merging argumentation systems. In: 2015 Brazilian Conference on Intelligent Systems (BRACIS). pp. 110–115 (2015). https://doi.org/10.1109/BRACIS.2015.45
12. Litscher, G., Cai, Y., Li, X., Lv, R., Yang, J., Li, J., He, Y., Pan, L.: Quantitative analysis of pork and chicken products by droplet digital pcr. BioMed Research International **2014**, 810209 (2014). https://doi.org/10.1155/2014/810209, https://doi.org/10.1155/2014/810209
13. MUNTEANU, A.R.: The Third Party Certification System For Organic Products. Network Intelligence Studies (6), 145–151 (December 2015), https://ideas.repec.org/a/cmj/networ/y2015i5p145-151.html
14. Naaum, A.M., Shehata, H.R., Chen, S., Li, J., Tabujara, N., Awmack, D., Lutze-Wallace, C., Hanner, R.: Complementary molecular methods detect undeclared species in sausage products at retail markets in canada. Food Control **84**, 339–344 (2018). https://doi.org/https://doi.org/10.1016/j.foodcont.2017.07.040, https://www.sciencedirect.com/science/article/pii/S0956713517303924
15. Panisson, A.R., Bordini, R.H.: Towards a computational model of argumentation schemes in agent-oriented programming languages. In: 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). pp. 9–16 (2020). https://doi.org/10.1109/WIIAT50758.2020.00007
16. Poon, J., Dryja, T.: "the bitcoin lightning net- work:scalable off-chain instant payments." (2016)
17. Rahwan, I., Larson, K.: Pareto optimality in abstract argumentation. In: Fox, D., Gomes, C.P. (eds.) Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008. pp. 150–155. AAAI Press (2008), http://www.aaai.org/Library/AAAI/2008/aaai08-024.php
18. Roth, B., Riveret, R., Rotolo, A., Governatori, G.: Strategic argumentation: A game theoretical investigation. In: Proceedings of the 11th International Conference on Artificial Intelligence and Law. p. 81–90. ICAIL '07, Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1276318.1276333, https://doi.org/10.1145/1276318.1276333