Journal of Network and Systems Management (2022) 30: 56
https://doi.org/10.1007/s10922-022-09669-1

Towards a Trustful Game-Theoretic Mechanism for Data Trading in the Blockchain-IoT Ecosystem

Seyednima Khezr[1] · Abdulsalam Yassine[2] · Rachid Benlamri[2,3]

Received: 24 November 2021 / Revised: 8 March 2022 / Accepted: 27 May 2022 /
Published online: 2 July 2022

Abstract

This paper introduces a data trading system based on the blockchain network, where a trusted data aggregator collects data from the Internet of Things (IoT) device owners and sells them in the format of different packages to multiple buyers. In this paper, we formulate infinitely repeated games between rational buyers that are competing with each other to obtain the required data records. Buyers update their bidding strategies to maximize their profits based on the outcome of previous games. We validate the existence and uniqueness of the Nash equilibrium in a one-shot game, finite, and infinitely repeated games. To ensure data owners' privacy, a novel trust mechanism design is used to impede untruthful buyers to win the game. To prevent the use of a third party such as an auctioneer, all of these methods are implemented as smart contracts on the Hyperledger blockchain. We provide extensive analysis to demonstrate that the proposed system satisfies the properties of completeness, soundness, computationally efficiency, truthfulness, budget balance, and individual rationality. Lastly, we provide simulation experiments to demonstrate the performance of our blockchain network using different metrics, such as transaction throughput, latency, and resource consumption under different parameters.

Keywords Blockchain network · Internet of things · Trust · Data trading · Data privacy · Game theory

1 Introduction

The IoT ecosystem is expanding daily, connecting the physical and digital worlds to transform the way we live and do business. With an increasing number of connected devices, a huge amount of data is instantly collected, aggregated, and

✉ Seyednima Khezr
skhezr@lakeheadu.ca

Extended author information available on the last page of the article

exploited in new applications in areas such as smart homes, smart cities, and health [1]. According to the estimates conducted by the International Data Corporation report [2], 41.6 billion IoT devices will be connected to the Internet by 2025 and generating 79.4 zettabytes of data. As the IoT devices become more instrumented and interconnected, data will grow exponentially [3, 4]. Data from IoT devices has spawned a new data economy in which people and companies can sell and exchange data [5]. As data continues to pile up, the data economy will continue to emerge and enable new IoT data marketplaces. Several companies, such as Terbine and Dfintech, have developed real-world applications to manage and monetize IoT generated data. These applications allow IoT device owners to sell their data to various stakeholders.

However, current online data marketplaces suffer from three main concerns. First, online data marketplaces such as CitizenMe, Terbine, Datacoup, DataExchange, and Factual, to name a few, use a centralized marketplace for their data trading systems, which is vulnerable to cyber-attacks and data leakage [6, 7]. Furthermore, it has been evidenced that existing applications that assist the IoT device owners to sell their data in exchange for money fail to clearly explain how, where, or with whom the users' data are being shared [8]. Second, existing applications also tend to package the owners' data for sale to other companies repeatedly [9]. Such information changes hands or ownership and the monetary benefit that companies are receiving as a result of selling the data packages is not passed back to IoT device owners [10]. Since data records can be sold repeatedly to multiple buyers, a key question to be addressed is how to devise a strategic negotiation model that maximizes the benefit of data owners and buyers. Third, trust becomes a challenge if data buyers are not trustful and they may misuse the data. Data owners may be reluctant to sell their data if the buyers are not trustful [11]. Thus, a key question is how to impede and impose penalties on untruthful buyers. Therefore, one significant aspect that needs to be taken into consideration is ensuring data owners have control of their data and have the autonomy to decide what information is collected, how it is used, and most importantly, how much it is worth. Consequently, designing a trustworthy data market, capable of selling and buying data which incentivizes the participants to maximize their profits under a fair trading mechanism is very critical.

To tackle the first challenge, we integrated the blockchain as a trustworthy and transparent mechanism that preserves the data owner's control over their data. Blockchain technology is difficult to tamper with and transactions are secure as well as transparent to all parties, including the users who generated the data [12]. As such, blockchain presents a solution for developing a transparent and trustful network for data trading and gives the data owners full control of their information, guards their privacy [13]. To tackle the second challenge, we formulate a non-cooperative game in infinite setup between the buyers in which each buyer strategically chooses the bidding price for that specific data to maximize their utilities. In particular, in each one-shot game (stage-game), a limited amount of records is traded. The game is played repeatedly and buyers learn from the outcome of the previous stage and update their bids over the next periods to increase their utilities until the demand is met. To tackle the third challenge, we propose a novel mechanism design based on the trust score. The proposed mechanism design impedes untruthful buyers to obtain the data based on a scoring rule function. Also, if the winner is not fully trusted (i.e., trust score less than 1), we consider a penalty on his/her payment for the current stage of the game. We build the entire scheme in the form of smart contracts on the Hyperledger blockchain to show our system is not using a third party (e.g., auctioneer). The main contributions of this paper are as follows:

- We design a blockchain-based system for data trading using a game-theoretic approach in the IoT ecosystem. In particular, we formulate a non-cooperative infinitely repeated game in which rational buyers are strategically deciding on their bids and learn from the outcome of each one-shot (stage) game and try to adjust their bids to maximize their utility. The non-cooperative nature of the game in the data market is properly modeled in a one-shot game by carefully defining utility functions. Using this one-shot game as a building block, we then proceed to define finite and infinitely repeated games with a discount factor that captures the repeated interactions among rational buyers.
- We show the existence and uniqueness of the Nash equilibrium under the discontinuous utility function setup. Our proposed system achieves Nash equilibrium using pure strategy in a one-shot, discounted finite and infinite repeated game horizon, where no buyers in the market can improve their utility by deviating their bids.
- To ensure data owners' assets are protected and are not being misused within the data trading system, blockchain is used as a means of data transparency and security. Furthermore, we filter out the untruthful buyers based on the scoring function, which is calculated through the trust score. In the payment stage, we consider a penalty for the winner if he/she is not fully trusted. Even with considering a penalty, we ensure that the individual rationality property is set up, which implies the winner buyer receives a non-negative utility.
- We demonstrate and provide a comprehensive theoretical and experimental analysis of the proposed system which satisfies the economic and security properties including, completeness, soundness, computationally efficient, truthful, and individually rational. Moreover, we analyze and evaluate the performance of proposed system using Hyperledger Caliper. Our work measures and analyze transaction throughput, latency, elapsed time, and resource consumption (memory consumption, CPU utilization, and disc read/write operations).

The organization of this paper is as follows: The next section discusses the related work. In Sect. 3, the data market structure is presented, followed by non-cooperative game theoretic approach for data trading in Sect. 4. Section 5 discusses the Nash equilibrium solutions. Section 6 discusses the mechanism design, while Sect. 7 presents system evaluation. Finally, conclusions are drawn and future research directions are discussed in Sect. 8.

2 Related Work

Extensive research has been conducted in order to monetize and trade data [14–18]. Oh et al. [14] proposed a non-cooperative game for data trading with privacy valuation for data consumers in the IoT environment. The paper introduced a method to unify the unit price of data for data brokers as well as an optimization model to maximize data providers' profits. Similarly, in other work, Oh et al. [15] proposed a data trading model between data owners and consumers as two natural logarithmic functions and a data broker who processes data and provides service to the consumers. This model guaranteed that a data broker will find a global maximum point to reach the best probability deal to sell the data. Tian et al. [16] proposed an optimal contract-based model for data trading between data sellers and consumers. This model maximizes the data seller's payoff while satisfying individual rationality and incentive compatibility properties for data consumers. The work in [17] introduced an iterative auction mechanism for data trading to coordinate the selfish agents in an optimal way to prevent direct access to private information. Khokhar et al. [18] proposed an entropy-based trust computation model to verify the correctness of data from untrusted data providers in the data market. This model utilized the Vickrey–Clarke–Groves auction mechanism for the valuation of data providers' attributes for determining truthful pricing strategies.

To build a more transparent data marketplace, blockchain-based data trading systems are studied in [19–23], and [24]. Liu et al. [19] introduced an optimal pricing mechanism for data trading in the IoT environment adopted by the two-stage Stackelberg competition based on the blockchain. The model presented a pricing and purchasing mechanism between the data consumer and the market-agency to maximize the profits of both parties. The work in [20] proposes a decentralized fair data trading system, which guarantees the availability of data and fairness between the sellers and buyers. The model implements homomorphic encryption, double-authentication-preventing signatures, and smart contracts to improve data availability and achieve fairness in data trading between participants. In the work presented in [19], the authors propose a blockchain-data market framework and an optimal pricing mechanism. They designed an optimal pricing mechanism to support efficient data trading in an IoT environment using a two-stage Stackelberg game. Sheng et al. [21] studied a crowdsourcing data trading system based on blockchain. The model implements a smart contract that enables sellers and buyers to conduct credible and truthful data trading while ensuring the copyright and quality of data. The authors also proposed a semantic-similarity-based auction mechanism to guarantee truthful data trading. Similarly, the authors in [22] investigated a blockchain-based data trading ecosystem that filters out dishonest buyers to guarantee the market's truthfulness. The security model in [22] includes a set of trading protocols based on asymmetric cryptography. The work in [23] proposes a trading model based on Ethereum smart contracts. It incorporates machine learning to guarantee fairness in data trading. All the participants in the blockchain network achieve a consensus on an authentication task, and any potential threats can be identified.

Table 1 Comparison between our work and existing studies

Research studies	Decentralization	Smart contracts	Reputation computation	Utility maximization	Performance evaluation
[14, 15]	✗	✗	✗	✓	✗
[16, 17]	✗	✗	✗	✓	✗
[18]	✗	✗	✓	✓	✗
[19, 20]	✓	✓	✗	✓	✗
[21, 22]	✓	✓	✓	✗	✗
[23, 24]	✓	✓	✗	✗	✗
Our work	✓	✓	✓	✓	✓

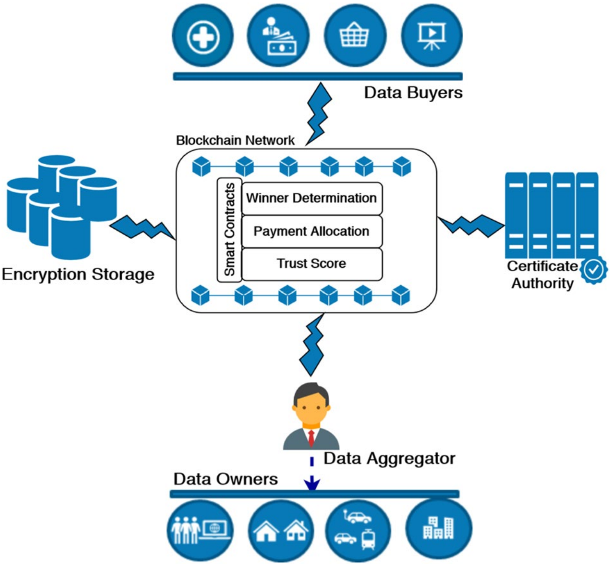

Fig. 1 High-level architecture of the proposed system

Truong et al. [24] proposed a blockchain-based for sharing IoT data, in which data owners can sell their private data. In this framework, smart contracts evaluate access control requests to off-chain encrypted data. Table 1 summarizes the comparison of our work and previous studies.

3 Data Market Structure

Figure 1 shows the high-level architecture of the proposed data market. At a high level, data buyers and a data aggregator (DA) register themselves to the certificate authority (CA) to obtain a legal identity. The CA issues certificates (digital identities contained in X.509 digital certificates) to each entity. In this market, continuous data records are generated through IoT devices and made for sale by data owners (DOs). The latter grant access permission to the DA to aggregate, package, and sell data records on their behalf according to a smart contract-based agreement. The DA informs all the buyers about the packages available for sale through the blockchain network. Buyers simultaneously reply with their bids, which include the bidding price and required data records from a specific package. Buyers will compete with each other to obtain desired records and learn from the outcome. Afterward, through a trustful auctioning process, data records will be awarded to one winner at a given time. Later, DA can leave a review score for a winner buyer for the current transaction. All the transactions will be added to the ledger. Our auction mechanism (winner determination and payment allocation) and review score are implemented in the form of smart contracts on the blockchain network.

3.1 Assumptions

Before describing the detailed process of the proposed system, the following assumptions are made:

(1) Infinite data records: we assume that DO $w \in W = \{1, 2, \dots, m\}$ produces infinite data records r_w^t from IoT devices, such as wearable devices and smart appliances at time t. This is reasonable since 41.6 billion IoT devices will be connected to Internet by 2025 [2].

(2) *Data aggregator* We assume that DA is a trusted entity, acting on behalf of DOs, involves in technical and operational tasks, such as deriving data records value based on DOs' privacy risk, data encryption, coding a smart contract, and blockchain operations. This is reasonable because performing these technical tasks for senior citizens equipped with IoT devices would be extremely difficult.

(3) *Certificate authority* We assume the CA to be fully trusted. This is reasonable since the CA is a government agency responsible for managing the identities and credentials of a data aggregator and buyers.

(4) *Data buyers* We assume that the data buyers do not share their bidding values among each other, and their behavior is non-cooperative with the goal of maximizing their benefit.

3.2 Process Details of the Proposed System

Figure 2 describes the sequence diagram of the interaction between DA and buyers. Once DA and buyers obtained their certificates, the DA creates

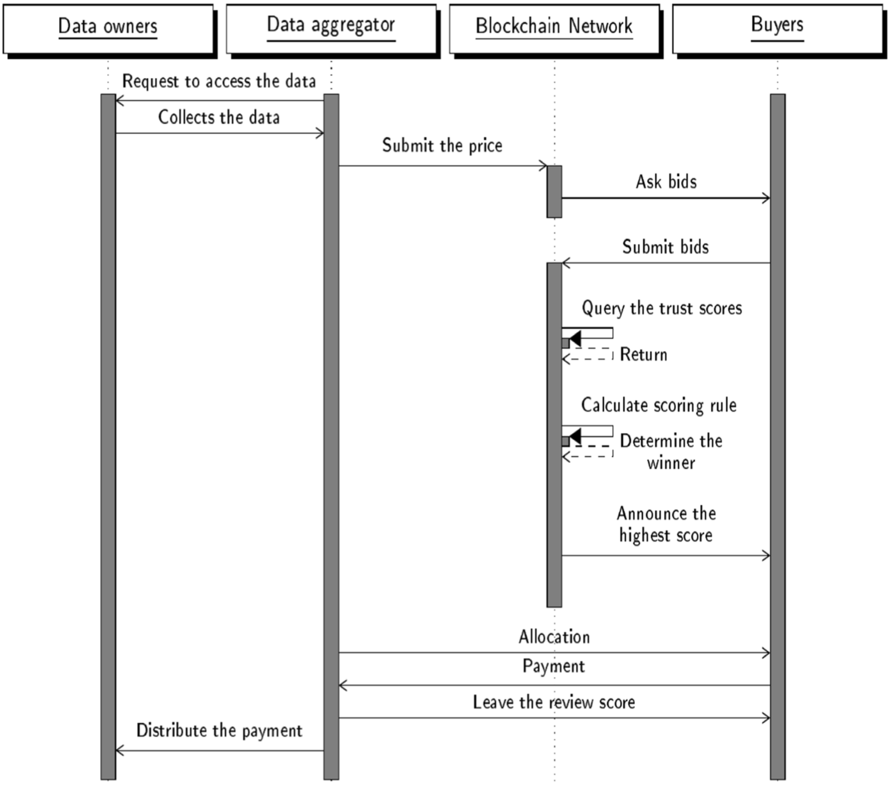

Fig. 2 Sequence diagram describes the interaction between DA and buyers

different packages based on data types received from DOs. For example, a package $\mathcal{D} = \{r_1^t, r_2^t, \dots, r_m^t\}$ may contain smart TV records or energy records. The DA encrypts and stores data records in a secure indexed database, and generates a decryption key. Once this is completed, the DA sends the index of the records to the blockchain. Afterward, the DA publicizes the packages to the blockchain network. Data buyers are the end-users who purchase the data. Let *B = {1, 2, ..., b}* be the set of data buyers in our system. Each data buyer is indexed by $i \in B$. Each data buyer i submits its bid $\beta_i^t = \left(g_i, v_i^t(x)\right)$ to the blockchain network, where g_i and v_i^t are total required quantity and reserved value, respectively. We denote x as the traded amount of records from package $\mathcal{D}$ in the market at time t. In this model, a limited number of records will be traded at each time t, until buyers fulfill their total demand g_i. The traded amount of records x can be defined by the CA or can be based on an agreement between players inside the market. For example, assume that a package consists of 10 million energy records about TV usage. Utility companies (i.e, buyers) are usually interested in different quantities, maybe one company is interested in one thousand records, while another company is interested in one million records, and in each auction period a hundred number of records are going to be sold. Thus, companies are going to keep competing with each other and biding simultaneously at each period (stage) until obtaining the desired quantity. Finally, after receiving the asking price for data records to be traded and bids from buyers, the smart contracts run as follow:

(1) In the first sub-stage, the winner determination smart contract retrieves the trust score of buyers from blockchain. The trust score is determined by DA and buyer i's previous trading experiences. Then, the smart contract will remove the bids which are less than the trust threshold. This is done to ensure untrustworthy buyers will not have a chance to get the data. Then, smart contract will run the scoring rule and announce the winner.

(2) In the second sub-stage, the payment allocation smart contract will run to determine the payment. In the payment stage, we impose a penalty on the payment of the winning buyer with respect to his/her trust score. The winner i receives data records and a decryption key. Simultaneously, the DA receives a payment amount p_i.

Our proposed systems are composed of following polynomial algorithms [25, 26]:

(1) **KeyGenSetup**$(1^{\lambda}) \longrightarrow \mathcal{SK}$: It is run by DA j and takes security parameter λ. It outputs the encryption key $\mathcal{SK}_j$.

(2) **Encrypt**$(\mathcal{SK} \Leftrightarrow \mathcal{D}) \longrightarrow \mathcal{C}$: It is run by the DA j to encrypt data. Given the encryption key of DA $\mathcal{SK}_j$, and data package $\mathcal{D}$, it outputs a ciphertext for the data package $\mathcal{C}$.

(3) **CreateIndex**$(\mathcal{SK} \Leftrightarrow \mathcal{D} \Leftrightarrow \mathcal{C}) \longrightarrow \mathcal{I}$: It is run by the DA j to create the index $\mathcal{I}$. Given the encryption key of DA $\mathcal{SK}_j$, data package $\mathcal{D}$, ciphertext $\mathcal{C}$, and it outputs the searchable index $\mathcal{I}$.

(4) **Trapdoor**$(\mathcal{SK} \Leftrightarrow \mathcal{Q}) \longrightarrow \mathcal{T}_Q$: It is run by the DA j to create trapdoor for the authorized buyer i. Given the encryption key of DA $\mathcal{SK}_j$, and query (e.g., keyword) $\mathcal{Q}$, it outputs the trapdoor $\mathcal{T}_Q$.

(5) **Search**$(\mathcal{I} \Leftrightarrow \mathcal{T}_Q) \longrightarrow \mathcal{SR}$: It is run by buyer i and evaluated by smart contract (1 or 0). Given the search index $\mathcal{I}$, and trapdoor $\mathcal{T}_Q$, it outputs the search result $\mathcal{SR}$, including ciphertext $\mathcal{C}$.

(6) **Eval**$(\mathcal{I} \Leftrightarrow \mathcal{T}_Q) \longrightarrow \pi$: Given index $\mathcal{I}$, and trapdoor $\mathcal{T}_Q$, it evaluates the search function and outputs the correctness proof π.

(7) **Verify**$(\mathcal{T}_Q, \mathcal{SR}, \pi) \longrightarrow$ *True* or *False*: Given trapdoor $\mathcal{T}_Q$, search $\mathcal{SR}$, and proof π, it outputs the (inculding ciphertext $\mathcal{C}$), and correctness proof π. It outputs *True* if the result is valid (correct) and *False* otherwise.

(8) **Decrypt**$(\mathcal{SK} \Leftrightarrow \mathcal{C}) \longrightarrow \mathcal{D}$: Given the encryption key $\mathcal{SK}$, and ciphertext $\mathcal{C}$, it outputs the decrypted data package $\mathcal{D}$ to buyer i.

3.3 Data Value

The value of data will be derived based on DOs' privacy risk. DOs may have different privacy attitudes, and as a result, they may set different values for their data records. For instance, some DOs may be concerned about their privacy and would allow a user to access a small portion of data in exchange for a few dollars, whereas others may not be concerned about privacy and they ask for a higher price. The DA derives the privacy risk $\Omega_w(r_w^t)$ of a DO w as follows [27]:

$$\Omega_w(r_w^t) = PC(r_w^t) \times SL(r_w^t) \quad \forall PC, SL \in [0, 1] \tag{1}$$

where $PC(r_w^t)$ denotes privacy concern of DO w, and $SL(r_w^t)$ denotes the sensitivity level of data [27]. The DA derives the privacy risk values $\Omega_{w,q}(r_w^t)$ of each w. Each DO w is described by a privacy risk value $\Omega_w(r_w^t)$ as well as a value of data $\mathcal{V}_w(r_w^t)$. Therefore, there is mapping $\mathcal{Z}$ between the privacy risk and the value of data that $\mathcal{Z} = [\Omega_w(r_w^t) \rightarrow \mathcal{V}_w(r_w^t)]$. The data values may vary for each DO. In order to find the final value of data records for each data type, we calculate the average value of data records as follows:

$$\bar{\mathcal{V}^t} = \left(\frac{\sum_{w=1}^{m} \mathcal{V}_w(r_w^t)}{m} \right) \tag{2}$$

where m is the total number of DOs which participate in selling data for a specific data type. Once the data aggregator announces the final value of the data to DOs, they can either accept or reject it $\langle Accept, Reject \rangle$. If the DO w decides to accept the final value then DA collects the data for further processing.

3.4 Data Aggregator Utility

For DA j, we define a cost function $C_j(r_w^t)$ representing the total cost incurring from operation, maintenance and electricity bill for the data records r_w^t at period t. It can be noticed that such cost increases with the size of data records, yielding an increasing and strictly convex cost function. We choose a quadratic function to model the cost function as follows [28]:

$$C_j(r_w^t) = a(r_w^t)^2 + b(r_w^t) + c \tag{3}$$

where $a, b, c \geq 0$ are constants. These parameters are dependent on the type of operation, maintenance, and electricity bill incurred to the DA. The utility function of DA is modeled by revenue of selling data records minus the cost:

$$U_j = \sum_{t=1}^{T} R_j(r_w^t) - C_j(r_w^t) \tag{4}$$

$$\text{subject to } x_w^t \leq r_w^t \tag{5}$$

where $R_j(r_w^t)$ is revenue function that is equal to the number of record sold at time t and its corresponding price.

Table 2 Notations

Notation	Description
w	Data owner (DO) $w \in W = \{1, 2, \dots, m\}$
j	Data aggregator (DA)
r_w^t	Infinite data records r_w^t of Do w at time t
i	Buyer $i \in B = \{1, 2, \dots, b\}$
x	Trading amount of records x
$\Omega_w(r_w^t)$	Privacy risk
$\mathcal{V}_w(r_w^t)$	Value of data
$\bar{\mathcal{V}}^t$	Average value of data records
$C_j(r_w^t)$	Cost function
U_j	Utility function for the DA
$u_i(\beta_i^t, \beta_{-i}^t)$	Utility function for buyer i for one-shot game
$v_i^t(x(\beta_i^t, \beta_{-i}^t))$	The buyers' valuation functions v_i^t
U_i	Overall utility of buyer i
G	One-shot game
$\mathcal{S}(\beta_i)$	Scoring rule funtion for buyer i bid
G_ψ^T	Finite repeated game with discount factor ψ
$Tr_n^t(j, i)$	Total trust DA has about a given buyer i
$T_{indirect}^{yi}$	Indirect trust
$p_i(\bar{\beta}_i)$	Payment of the winner buyer i

$$R_j(r_w^t) = \max \sum_t^T r_w^t \times p_{ji}^t, \tag{6}$$

$$\text{subject to } p_{ji}^t > C_j(r_w^t) + \bar{\mathcal{V}}^t \tag{7}$$

Equation (4) ensures that the DA trades no more than the agreed upon amount of records. Significant notation is summarized in Table 2 for the clarity of readers.

4 A Repeated Game Theoretic Approach for Data Trading Between Buyers

In this section, we present a non-cooperative game for data buyers in the infinite repeated horizon. A repeated game is one where the buyers repeatedly play the same one-shot game in each time period (called a stage game) in which they play simultaneously [29]. We first formulate the utility function for a one-shot game G. Then, using the one-shot game definition as a building block, we then proceed to define finitely G^T and infinitely repeated games G^∞ that capture repeated interactions among the different buyers. We consider a data market setting for one-shot game $G = \langle B, A_i, u_i \rangle$, where B is set of buyers. Each buyer i has an action set A_i. An action profile $\beta = (\beta_i, \beta_{-i})$ consists of the bid of buyer i and bids of other buyers, denoted by

$\beta_{-i} = (\beta_1, ..., \beta_{i-1}, \beta_{i+1}, ..., \beta_b) \in A_{-i}$. In addition, each buyer i has a real-valued, one-shot game utility function $u_i : A_i \rightarrow \mathbb{R}$, which maps every action profile $\beta \in A$ into a utility for buyer i, where A denotes the cartesian product of the action spaces A_i, written as $A = \prod_{i=1}^{B} A_i$.

4.1 Data Buyers Utility

We assume a buyer i, who needs a number of records from package $\mathcal{D}$ of a specific type, knows his own valuation of the current traded amount of records, but not those of his opponents. On receiving the required amount of records, the buyers pay the price $p_i^t\big(x(\beta_i^t, \beta_{-i}^t)\big)$, conditional on winning records, given the other buyers bid β_{-i}^t. If the game G is played only once, the utility function for the buyer i is the difference between valuation for traded amount of records and payment. The utility function u_i of buyer i for one-shot game is:

$$u_i(\beta_i^t, \beta_{-i}^t) = \sum_t v_i^t\big(x(\beta_i^t, \beta_{-i}^t)\big) - p_i^t\big(x(\beta_i^t, \beta_{-i}^t)\big) \tag{8}$$

where v_i^t is buyer i's valuation for the trading amount of records x. It represents how much the requested records are worth to the buyer i. The buyer i hopes to pay a smaller price p_i then his estimated value v_i. The buyers' valuation functions v_i^t are drawn independently from the following equation:

$$v_i^t(x(\beta_i^t, \beta_{-i}^t)) = v_i^t\left(x\left(1 + \log\big(\beta_i^t(\varphi), \beta_{-i}^t(\varphi)\big)\right)\right) \tag{9}$$

where φ denotes the satisfaction rate of buyer i $(0 \leq \varphi \leq 1)$. This means that if buyer i is not satisfied with the quality of obtained records at stage t, the valuation of the buyer i in next stage decreases as show in in Fig. 3. We assume that the satisfaction rate of buyers $i \in B$ is 1 at the beginning. After receiving records at stage t, the

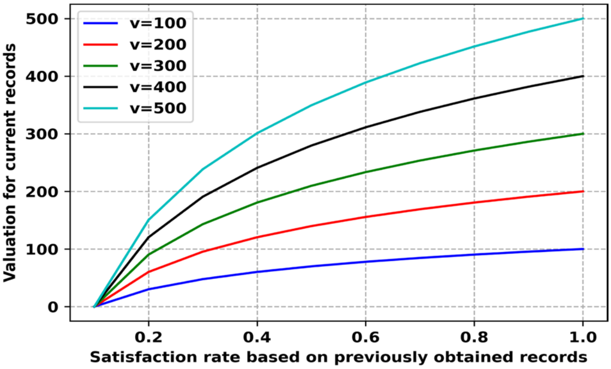

Fig. 3 Valuation function. This example shows that the valuation v_i of buyers B decreases on the basis of the satisfaction rate φ. Note that this example does not take into account the average of all satisfaction rates

buyer measures the quality of the records and updates the satisfaction rate. The buyers use the average of satisfaction rates as their valuation for the next stage. Figure 2 describes they way valuations are affected by the satisfaction rates.

In the next stage, the game G structure stage does not change. Buyers will continue bidding until they obtain the total quantity they needed. The overall utility of buyer i in the repeated game G_{ψ}^{T} is:

$$U_i = (1-\psi)\sum_{t=1}^{T}\psi^{t-1}u_i(\beta_i^t, \beta_{-i}^t) \tag{10}$$

where ψ is a discounted factor and β_{-i}^t denotes the set of bids submitted by the buyers other than i at stage t. We assume that future utilities are discounted proportionally at some rate ($0 \le \psi \le 1$). $\psi = \frac{1}{1+r}$, where r is the interest rate. We used fictitious play as type of learning for the buyers [30, 31]. Each buyer i starts with some belief about what are the bids of other buyers. Each buyer i updates his/her beliefs based on what he/she observed in the iteration of G_{ψ}^{T}. More formally, let $\eta_i^t(\beta_{-i})$ denotes the number of times buyer i has observed (β_{-i}) in the previous stages. So, buyer i assesses other buyers bid using fictitious learning as follow [30, 31]:

$$\sigma_i^t(\beta_i) = \frac{\eta_i^t(\beta_{-i})}{\sum_{\beta-i\in A-i}\eta_i^t(\beta_{-i})} \tag{11}$$

where $\sigma_i^t(\beta_i)$ is the probability that is proportional to the time it was played in the past. This means that buyer i forecasts buyer $-i$'s bid at time t to be the empirical frequency distribution of past G. Given buyer i's belief about other buyers play, he/she chooses the bid at time t to maximize his/her utility [30, 31]:

$$\beta_i^t \in \arg\max_{\beta_i\in A_{-i}}(\beta_i, \sigma_i^t) \tag{12}$$

5 Nash Equilibrium Solutions

The Nash equilibrium (NE) of a game is an action profile (list of actions—one for each buyer) with the property that no player can increase his utility to achieve higher benefits by choosing a different action given the other buyers' actions. To maximize the utilities, the buyers adjust their bids to reach the equilibrium. This means that if a NE exists for the game, then all buyers $i \in B$ are expected to converge to the state represented by the equilibrium. So, each buyer i aims to choose the strategy or action that maximizes its utility function to determine the best outcome. In addition, the players in the one-shot game choose their own bids independently and simultaneously and try to maximize their expected utility. There are two types of strategies or actions available for players: pure strategies and mixed strategies. Pure strategy defines an action that a player wants to take with positive probability from a given set of strategies in the game. In contrast, a mixed strategy for a player is a probability distribution over his/her pure-strategy choices. In our model, we will prove that the pure strategy equilibrium exists for the proposed one-shot game. The objective function of the players is to maximize their utilities. Before finding the NE of our one-shot game *G*, we first define formally the best response and NE. For the sake of clarity, we are dropping *t* notations, referring to the time, since we are dealing with a one-shot game.

Definition 1 (Best response [32]) Assuming all the buyers $i \in B$ are rational, a buyer *i* played his/her bid $\left(\beta_i^*\right)$ as best response to the other buyers' β_{-i} played action $\left(\beta_{-i}^*\right)$ such that:

$$\beta_i^* \in BR\Big(\beta_{-i}\Big) \text{ iff } \forall \beta_i \in A_i, u_i\Big(\beta_i^*, \beta_{-i}\Big) \geq u_i\Big(\beta_i, \beta_{-i}\Big) \tag{13}$$

Definition 2 (Nash equilibrium [32]) The NE is a profile of actions, one for each buyer, such that each action is the best response to the other buyers actions. Specifically, an action profile β is said to be NE, if:

$$\beta^* = \langle \beta_1^*, \beta_2^*, ..., \beta_n^* \rangle \text{ is a NE iff } \forall i,\ \beta_i^* \in BR\Big(\beta_{-i}^*\Big) \tag{14}$$

We will first define the existence of NE for the finite repeated game, which can be viewed as a generalization of the equilibrium concept for the one-shot game. We should point out here that we won't be able to construct subgame perfect nash equilibrium (SPNE) i.e., induced normal form - backward induction, which is the standard solution for finding NE. SPNE works only when the utility function is continuous and only applies to finite games. However, in our model, the utility function of buyers in (8) introduces a discontinuity in utilities. This means that the u_i could be zero at some stage *t* for buyer *i*, or it could have non-zero value. Hence, we will use the following approach to finding the NE in every one-shot game with a discontinuous utility setting. Next, we will leverage the results by using the Folk theorem in the infinitely horizon setup to find NE.

Theorem 1 *A Nash equilibrium exists in the proposed non-cooperative game* $G = \langle B, A_i, u_i \rangle$.

Proof The Nash equilibrium exists only when the following conditions are satisfied [33]:

(1) $A_i \subseteq \mathbb{R}^m, (i = 1, ..., b)$ is a non-empty, compact and convex subset of Euclidean space.
(2) $u_i = A_i \to \mathbb{R}^b$ is upper semi-continuous in β and quasi-concave in $\beta_i\ \forall i$.

□

Obviously, the first condition can be satisfied since A_i is defined by a set of bidding vectors in which all the values are between zero and the maximum bidding of buyers. So, it is a nonempty, compact and convex subset of the Euclidean space R^b. To show that $u_i = A_i \rightarrow \mathbb{R}^b$ is upper semi-continuous, we first define the following property [34]:

Definition 3 $u_i(\beta_i, \beta_{-i})$ is upper semi-continuous at β_{i0} if $\exists \beta_i$ as a neighborhood such that:

$$\lim_{\beta_i \rightarrow \beta_{i0}} \sup u_i(\beta_i, \beta_{-i}) \leq u_i(\beta_{i0}, \beta_{-i}) \tag{15}$$

For a jump point of u_i in a given range $\Delta\beta$, we define [34]: $\beta_{i0} = \beta_i + \Delta\beta$ such that $p_1 \leq p_2$, $p_t(\beta_i, \beta_{-i}) = p_1$, and $p_t(\beta_{i0}, \beta_{-i}) = p_2$. This means that u_i function is upper semi-continuous because rational buyers $i \in B$ attempt for a higher utility around the discontinuity point. Only the quasi-concave property remains to be proved. Taking the derivatives of (8) with respect to β_i, we get:

$$\frac{\partial u_i}{\partial \beta_i} = \frac{v\big(x(\beta_{-i})\big)}{ln(10)\beta_i} - 1 \tag{16}$$

$$\frac{\partial^2 u_i}{\partial \beta_i^2} = -\frac{v\big(x(\beta_{-i})\big)}{ln(10)\beta_i^2} \tag{17}$$

Since $\frac{\partial u_i}{\partial \beta_i} = \frac{v\left(x(\beta_{-i})\right)}{ln(10)\beta_i} - 1 > 0$ and $\frac{\partial^2 u_i}{\partial \beta_i^2} = -\frac{v\left(x(\beta_{-i})\right)}{ln(10)\beta_i^2} < 0$, the utility function u_i is concave with respect to β_i, hence it is quasi-concave in β_i [28], thus we get:

$$u_i((1-\lambda)\beta_i^x + \lambda\beta_i^y, \beta_{-i}) \geq \min\{u_i(\beta_i^x), u_i(\beta_i^y), \beta_{-i}\} \tag{18}$$

where β_i^x and β_i^y belong to the buyer i action set A_i. Therefore, u_i is a quasi-concave in β_i $\forall i$. Thus, we have proved the existence of the NE.

Theorem 2 *The NE of game* $G = \langle B, A_i, u_i \rangle$ *is unique.*

Proof The uniqueness proof is to show that the best response function of each buyer β_i^* is a standard function. Based on best response Definition 13 and using Eq. (16), the best-response is achieved when the first derivative of u_i is equal to 0, thus we have:

$$\frac{\partial u_i}{\partial \beta_i} = \frac{v\big(x(\beta_{-i})\big)}{ln(10)\beta_i} - 1 = 0 \tag{19}$$

and we obtain:

$$\beta_i^* = f(\beta) = \frac{v\big(x(\beta_{-i})\big)}{ln(10)} \tag{20}$$

A function $f(\beta)$ is a standard function [35], if the following properties are satisfied:

(1) *Positivity* $f(\beta) \geq 0$;
(2) *Monotonicity* For all β and $\hat{\beta}$, if $\beta \geq \hat{\beta}$, then $f(\beta) \geq f(\hat{\beta})$;
(3) *Scalability* For all $\mu > 1$, $\mu f(\beta) \geq f(\mu\beta)$;

$f(\beta)$ satisfies the three above properties of a standard function.

Positivity: The best-response function in (19) is always positive, so $f(\beta) \geq 0$ positivity property is set up.

Monotonicity: Assuming $\beta \geq \hat{\beta}$, then

$$f(\beta) - f(\hat{\beta}) = \frac{v\big(x(\beta - \hat{\beta})\big)}{ln(10)} \geq 0 \tag{21}$$

we have $f(\beta) - f(\hat{\beta}) \geq 0$, in which $f(\beta)$ is monotonically increasing function.

Scalability: For all $\mu > 1$ we have,

$$f(\beta) = \mu\frac{v\big(x(\beta_{-i})\big)}{ln(10)} \quad f(\beta) = \frac{\mu v\big(x(\beta_{-i})\big)}{ln(10)} \tag{22}$$

So, for all $\mu > 1$, $\mu f(\beta) \geq f(\mu\beta)$ thus, scalability property holds. Therefore, there exists a unique NE in the above one-shot game G, which can be viewed as the finite repeated game G_{ψ}^{T}. □

If the stage-game of a finitely repeated game has a unique NE, then we can consider that constant action for each buyer i, always play the stage-game best response irrespective of the past history. The infinitely repeated games requires different setup than finitely repeated games since it dose not have a terminal point. Before finding the NE of infinitely repeated game G_{ψ}^{∞}, we need to formally define the minmax value, enforceable and feasible utility as follows:

Definition 4 (Minmax value [32]) Considering stage-game $G = \langle B, A_i, u_i \rangle$, the minmax value v_i for each buyer i is:

$$v_i = \min_{\beta_{-i}} \max_{\beta_i} u_i(\beta_i, \beta_{-i}) \tag{23}$$

It represents the amount of utility buyer i receives when the other buyers play minmax strategies and buyer i plays the best response.

Definition 5 (Feasible [32]) Given a set of utility vector $U = (u_1, u_2, ..., u_n)$, U is said to be feasible if the convex hull of U is expressed as:

$$\mathcal{H} = Conv\left\{ u \in \mathbb{R}_+ \mid \exists \beta \in \mathbb{R}_+, U_i = \sum_{t=1}^{\infty} u_i \right\} \tag{24}$$

First, we need to apply Definition 5 to the set of utilities in a stage game G. Then, the convex hull of U will be determined by the convex combination between all utility vectors. Note that convex hull $\mathcal{H}$ of the vector utilities is achievable with pure strategies. In other words, a utility profile is feasible if it is a convex, and convex combination of the outcomes in G.

Definition 6 (Enforceable [32]) A utility vector U is said to be enforceable, if:

$$\mathcal{U} = \{u_i \geq v_i, \forall i \in B\} \tag{25}$$

The set of feasible and enforceable utilities is $\mathcal{E} = \mathcal{H} \cap \mathcal{U}$. Therefore, any set of feasible and enforceable utilities in $\mathcal{E}_\infty$ (Infinitely repeated game), $\mathcal{E}_T$ (Finitely repeated game), and $\mathcal{E}_\psi$ (Discounted repeated game) are always included in $\mathcal{E}$.

Theorem 3 *A Nash equilibrium exists in infinitely repeated game* $G^\infty_\psi = \langle B, A_i, U_i \rangle$, *if U is enforceable and feasible in* $\mathcal{E}$, *such that for each buyer i, we have* $u_i \geq v_i$ [36]. *Then* $\mathcal{E}_\psi \xrightarrow[\psi \to 0]{} \mathcal{E}$.

Proof According to [37], there exists NE in discounted infinitely repeated game. There can be many NE in the infinitely repeated games G^∞ even if the stage game only has a unique NE. □

6 Mechanism Design

In this section, we design a truthful mechanism to determine the winner allocation and corresponding payment for the proposed one-shot game G (possibly $G \rightarrow G^T$) and implement it in form of smart contracts. The process of developing a mechanism design faces two primary challenges. One is how to determine the winner buyer and allocation. The other is how much the winner buyer should pay for the records. This section addresses these two issues by using a scoring function based on the trust score to evaluate the buyer's bid and announce the winner. Furthermore, we consider a penalty for the winner if he/she is not fully trusted with respect to his/her trust score. Given the reservation price $p_j(x)$ and the submitted bids, the smart contracts will return the winner allocation and payment rules.

6.1 Winner Allocation Stage

In winner allocation stage, each data buyer $i \in B$ submits his/her bid $\beta_i = \left(g_i, v_i(x)\right)$ simultaneously to the blockchain at stage t. The valuation $v_i(x)$ for the traded amount of records offered in all stages is unknown to the DA. In this model the winner allocation stage includes two steps. In the first step, after receiving bids from buyers, the smart contract collects the trust score of buyers who participated in the bidding process and eliminates buyers whose trust score is less than a threshold $\mathcal{T}$. The $\mathcal{T}$ is determined based on the average of data sensitivity $SL(r^t_w)$ $\forall w \in W$, which is obtained through Eq. (1). In the second step, the scoring function $\mathcal{S}(\beta_i)$ is calculated for each buyer $i \in B$ according to the following scoring rule:

$$\mathcal{S}(\beta_i) = \beta_i \times Tr^t_n(j, i) \tag{26}$$

$$\text{subject to } \beta_i \geq p_i \tag{27}$$

where $Tr^t_n(j, i)$ measures the total trust DA has about a given buyer i, which is computed using the current satisfaction Tr_c and previous trust score $Tr^t_{n-1}(j, i)$ as shown in Eq. (28). In case DA does not have a prior trust for buyer i, we take indirect trust $T^{yi}_{indirect}$ into account. The total trust function is defined as follows:

$$Tr^t_n(j, i) = \begin{cases} \alpha \times Tr_c + (1 - \alpha) \times Tr^t_{n-1}(j, i), & \text{if } Tr(.) > 0 \\ \alpha \times Tr_c + (1 - \alpha) \times T^{yi}_{indirect}, & \text{if } Tr(.) = 0 \end{cases} \tag{28}$$

Here α is a relative weight that changes based on the accumulated deviation defined in Eqs. (32, 33 and 34). The Tr_c function measures how much DA j is satisfied about data buyer i. It represents the satisfaction score for the most recent transaction between j and i $(0 \leq Tr_c \leq 1)$.

$$Tr_c = Sat_c \times \frac{(1 - e^{-\lambda Val^v_n(j,i)})}{1 + \xi^v_n(j, i)} \tag{29}$$

Here Sat_c is a feedback-based factor (e.g., review score) for the current transaction n reflecting the way DA j rates data buyer i [38].

$$Sat_c = \begin{cases} 0, & \text{if } j \text{ is totally unstatisied with } i, \\ 1, & \text{if } j \text{ is totally statisied with } i, \\ \in (0, 1), & \text{otherwise.} \end{cases}$$

$Val^v_n(j, i)$ is a recent value fluctuation between the previous and current value Val_c. Here, λ is the decay constant and it controls the trust value. The Tr_c value reaches 1.0 with larger $Val^v_n(j, i)$ and decreases slowly with smaller $Val^v_n(j, i)$. For example, if the transaction's value is insignificant and current satisfaction is high, this will have little effect on overall trust. On the other hand, if the value of the transaction is high, and current satisfaction is high, the overall trust will be increased significantly.

$$Val^v_n(j, i) = |Val^v_{n-1}(j, i) - Val_c| \tag{30}$$

$$\xi^v_n(j, i) = \mathcal{K} \times Val^v_n(j, i) + (1 - \mathcal{K}) \times \xi^v_{n-1}(j, i) \tag{31}$$

Here $\xi^v_n(j, i)$ represents the accumulated value deviation for the history of all transactions. The α is relative weight which gives higher weight to the recent n [38]. The weight of α changes based on the accumulated deviation $\xi^n_t(j, i)$ [38].

$$\alpha = threshold + \mathcal{K} \times \frac{\delta_n^t(j,i)}{1 + \xi_n^t(j,i)} \tag{32}$$

$$\delta_n^t(j,i) = |Tr_{n-1}^t(j,i) - Tr_c| \tag{33}$$

$$\xi_n^t(j,i) = \mathcal{K} \times \delta_n^t(j,i) + (1 - \mathcal{K}) \times \xi_n^t(j,i) \tag{34}$$

Here $\mathcal{K}$ is some user-defined constant factor which controls to what extent we will react to the recent error $\delta_n^t(j,i)$ [38]. So, if we increase the value of $\mathcal{K}$, then we give more significance to the recent deviation than accumulated deviation [38]. The *threshold* is used to prevent α from saturating to a constant value. $T_{indirect}^{yi}$ value is computed when DA j does not have a prior trust relationship and experience with buyer i. The DA requests other entities' $y \in Y$ to provide their rating about the target buyer i. So, the DA will have the capability and experience to truthfully judge the data buyer for the first transaction. The indirect trust function is:

$$T_{indirect}^{yi} = \sum_{y=1}^{Y} \frac{\mathcal{P}_{yi}}{\mathcal{P}_{yi} + \mathcal{N}_{yi}} \tag{35}$$

where, $\mathcal{P}_{yi}$ denotes positive feedback of entity y ($0.5 \leq \mathcal{P}_{yi} < 1$), and $\mathcal{N}_{yi}$ termed as negative feedback ($0 \leq \mathcal{N}_{yi} \leq 0.5$). So, $T_{indirect}^{yi}$ represents the total number of positive and negative feedbacks for buyer i. Based on Eq. (26), the buyer with the highest score wins the game at stage t. If the buyers have an equally high trust score, we randomly selected the winner. After that, allocation rule will apply $\mathcal{X} : \mathbb{R}_+ \rightarrow [0,1]$, meaning that with the score bid profile $\mathcal{S}(\beta_i)$, buyer i gets the records with probability $\mathcal{X}\big(\mathcal{S}(\beta_i)\big)$ and makes a payment of $p_i(\beta) \in \mathbb{R}$, which indicates the amount that buyer i must pay. Furthermore, allocation rules have to satisfy the feasibility constraint as follows:

$$\sum_{i \in B}^{t} \mathcal{X}\big(\mathcal{S}(\beta_i)\big) \leq x \quad \forall \beta \tag{36}$$

Equation (36) restricts the allocation of records for the winner not to be more than the traded amount at stage t. Other buyers will modify their bids accordingly for the next stage of the game. Buyer $i \in B$ will continue bidding until obtaining the total quantity he/she requested.

$$g_i \geq \sum_{t}^{\infty} x \tag{37}$$

6.2 Payment Stage

In the payment stage, we consider a penalty for the winner if he/she is not fully trusted i.e., a trust score less than 1. If the winner is fully trusted which implies ($Tr_n^t(j,i) = 1$), he/she will not be punished. The trust is calculated based on Eq. (28). The payment rule of winning buyer $\bar{\beta}_i$ is:

$$p_i(\bar{\beta}_i) = \bar{\beta}_i - u_i\Bigg(1 + \log\left(Tr_n^t(j,i)\right)\Bigg) \tag{38}$$

The above equation provides assurances that buyer i is punished only on its stage utility u_i and will not be charged more than its bid. Algorithm 1 describes the winner allocation as well as the payment stage.

```
Algorithm 1 Winner allocation and payment stages
Input: Submitted bids $\beta_i$ and reservation price $p_j$ for stage $t$
Output: Winner allocation $\mathcal{X}\big(\mathcal{S}(\bar{\beta}_i)\big)$, payment $p_i(\bar{\beta}_i)$, current satisfaction score $Sat_c$
            Stage 1: Winner allocation stage
for each $\beta_i(g_i, v_i(x))$ and $p_j$ do
    Calculate the $Tr_n^t(j,i)$
    if $Tr_n^t(j,i) \geq \mathcal{T}$ then
        Calculate the $\mathcal{S}(\beta_i) = \beta_i \times Tr_n^t$
    else
        Remove $\beta_i$ Eliminating bids less than thershold $\mathcal{T}$
    end if
end for
$\mathcal{S}(\bar{\beta}_i) = \max\big(\mathcal{S}(\beta_i), i = 1 \text{ to } b\big)$
$\mathcal{X}\big(\mathcal{S}(\bar{\beta}_i)\big) = 1$ Allocation for the highest score
$\mathcal{X}\big(\mathcal{S}(\beta_i)\big) = 0, i = 1$ to $b$
            Stage 2: Payment stage

Calculate the stage utility $u_i$ for winner $i$
$p_i(\bar{\beta}_i) \leftarrow \bar{\beta}_i - u_i\Bigg(1 + \log\big(Tr_n^t(j,i)\big)\Bigg)$
Leave satisfaction score $Sat_c$ for winner $i$
Return $\Bigg(\mathcal{X}\big(\mathcal{S}(\bar{\beta}_i)\big), p_i(\bar{\beta}_i), Sat_c\Bigg)$
```

7 Evaluation of Results

In this section, we first present our security analysis of the proposed data market, which satisfies the properties of completeness and soundness. Then, we evaluate the model and analyze the results using different properties such as computational efficiency, bidding learning process, truthfulness, individual rationality, and budget balance. Then, we evaluate the performance of the blockchain network using different metrics, such as transaction latency, transaction throughput, and resource consumption, under varied scenarios and parameters using Hyperledger Caliper. All the results were conducted using a Ubuntu Linux Intel Core(TM) i7-3610QM CPU @2.30GHz with 6 GB RAM in our experiments.

7.1 Parameters and Experiment Settings

For the evaluation of the model, since the record price is decided by the number of DOs $w \in W$ based on their privacy risk, we choose reasonable values for our experiments. We assigned a privacy risk value that is uniformly distributed between 0 and 1 to reflect the privacy attitude of the different DOs. Then, we calculate the average value of the data. We assumed that the data records cost varies between 0 and 1, which is incurred by DA. We vary the number of buyers for evaluating the performance of our proposed data market. For the blockchain implementation, we deployed two organizations (Org0, Org1), each consisting of one peer (peer0.org1.example.com, and peer0.org2.example.com), each consisting of one database (couchdb.org1.example.com, and couchdb.org2.example.com), each consisting of one certificate authority (ca.org1.example.com, ca.org2.example.com) and one orderer node (orderer.example.com) hosted inside a Docker container in Hyperledger Fabric. We choose the Solo consensus mechanism as an ordering service for our implementation. We run the Hyperledger Caliper benchmark framework [39] on top of the Fabric to analyze our performance. We test the system performance under open workload (i.e., opening accounts and testing the writing performance of the ledger), query workload (i.e., querying accounts and testing the reading performance of the ledger), and transfer workload (i.e., data trading between accounts and testing the transaction performance of the ledger). The workload reflects the actual production usage. It is proportional to the amount of time and computational power required to execute a specific task (e.g., amount of time and computing resources used to create a new transaction on the blockchain). Table 3

Table 3 Summary of performance with 500 transactions

Name	Succ	Fail	Send rate (TPS)	Max latency (s)	Min latency (s)	Avg latency (s)	Throughput (TPS)
Open	500	0	50	9.11	5.41	7.16	39
Query	500	0	50	2.42	1.11	1.27	41
Transfer	500	0	50	1.69	1.02	1.21	48

summarize the performance of the blockchain network with 500 transactions using 50 transactions per second (tps).

7.2 Security Analysis

Our system satisfies the soundness and completeness properties. The soundness property guarantees dishonest entity does not perform any attacks such as eavesdropping and malleability attacks. The proposed system is said to be soundness, if the following properties are satisfied [40]:

$$Pr\left[\begin{array}{l} \mathbf{Verify}\left(\mathcal{T}_Q, \overline{S}, \overline{\pi}\right) = 1 \ : \\ S\mathcal{K} \leftarrow \mathbf{KeyGenSetup}\left(1^{\lambda}\right) \wedge \\ \overline{S} \not\leftarrow \mathbf{Search}\left(\mathcal{I}, \mathcal{T}_Q\right) \wedge \\ \overline{\pi} \longleftarrow \mathbf{Adv}\left(\mathcal{I}, \mathcal{T}_Q\right) \end{array}\right] \tag{39}$$

where **Adv**, is an adversary algorithm to forge $\overline{\mathcal{S}}$ and $\overline{\pi}$ variables. According to [41], Hyperledger Fabric satisfies the soundness property. The completeness property guarantees the search operations are carried out correctly and faithfully. The proposed system is said to be completeness, if the following properties are satisfied [40]:

$$Pr\left[\begin{array}{c} \mathbf{Verify}(\mathcal{T}_Q, \mathcal{S}, \pi) = 1 \ : \\ \mathcal{S} \longleftarrow \mathbf{Search}(\mathcal{I} \Leftrightarrow \mathcal{T}_Q) \wedge \\ \pi \longleftarrow \mathbf{Eval}(\mathcal{I} \Leftrightarrow \mathcal{T}_Q) \end{array}\right] = 1 \tag{40}$$

The search function is translated to a quadratic arithmetic program with sets of polynomials. The evaluation of the search function with input and output parameters is equal to the divisibility check of the target polynomials [42]. Thus, the completeness property is set up.

7.3 Computational Efficiency Analysis

We evaluate the property of computational efficiency property, which means that winning determination and the payment stages in Algorithm 1 must be solved within a polynomial time. The computation complexity of the Algorithm 1 is $O(n)$, where n is the number of bidders. In the winning allocation stage, the for loop runs for all submitted bids and then calculates the trust score and scoring rule. The computational complexity of the for loop takes $O(n)$. The max operation will take $O(n)$, and allocations will take $O(1)$. In the payment stage, each statement will take $O(1)$ to finish. Therefore, the computational complexity of the proposed system is bounded by $O(n)$ time complexity at the most. We select 10, 50, 100, 150, 200, 250 and 300 buyers for the experiment, respectively. Figure 4 shows the running time of Algorithm 1 under various numbers of buyers. These results indicate that Algorithm 1 completes

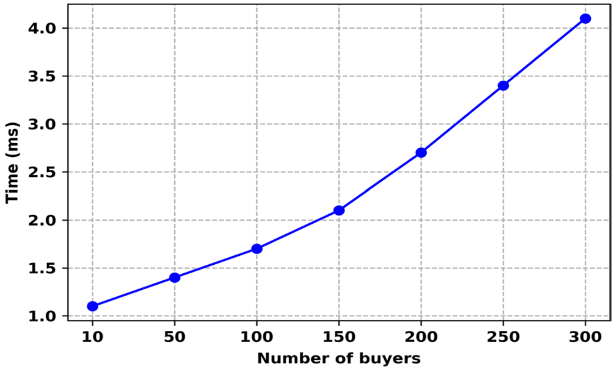

Fig. 4 Computationally efficient property

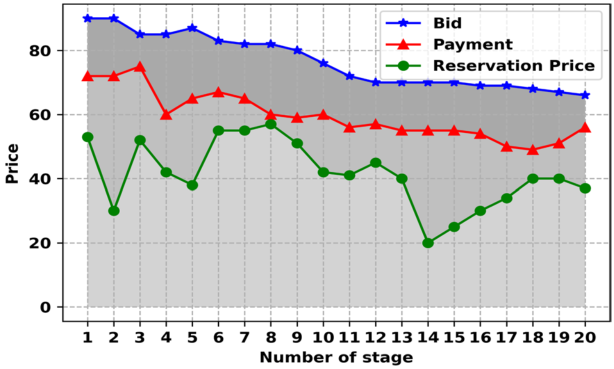

Fig. 5 Budget balance property

the computation in almost linear time, as demonstrated in the time complexity analysis given above. Therefore the computational efficiency property is satisfied.

7.4 Budget Balance Analysis

We verify the budget balance property. Budget balance means at each stage t, the buyer's payment is higher than the reservation price (asked-price) of the data aggregator. As shown in the payment stage described in Algorithm 1, the buying price for the winner, taking into account the penalty, is greater than the selling price $\beta_i \geq p_i$. To verify the property of budget balance, we repeatedly run the game until stage 20 as shown in Fig. 5. We can see that the curve line representing the buyer's payment is higher than of the asking price. Since we are imposing a penalty in the payment stage, definitely we satisfy the property of budget balance.

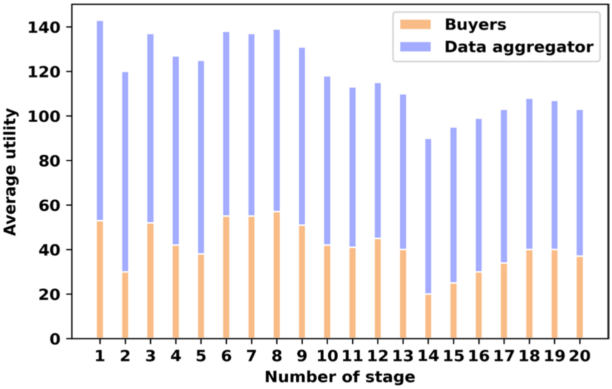

Fig. 6 Individual rationality property

7.5 Individual Rationality Analysis

We evaluate the property of individual rationality in which each winning buyer i must receive a non-negative stage utility $u_i \geq 0$. Similarly, DA utility must be non-negative $U_j \geq 0$. The stage utility is $u_i = 0$ for buyers who are not selected in the winning determination stage. Even by imposing a penalty on the winning buyer in the payment stage, the winning buyer has non-negative u_i. For example, in the worst-case scenario, let's assume that the buyer trust score is 0.1. The payment would be the same as the bidding price. So, the buyer's utility is non-negative. Figure 6 shows the average utility for the buyers and DA. We can observe that the winning buyer receives non-negative utility considering penalty in each stage. It can also be observed that DA has a non-negative utility. From the above, we can verify the individual rationality property of our proposed system.

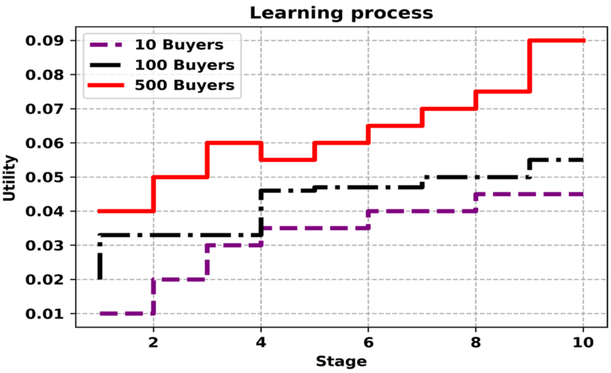

Fig. 7 Learning process

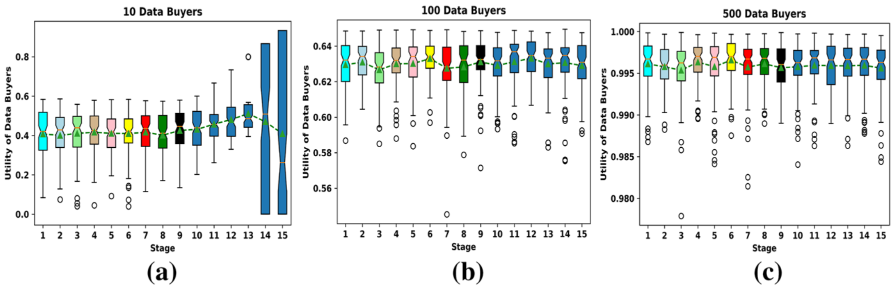

Fig. 8 Boxplot presenting the bid learning processing using fictitious play

7.6 Bidding Learning Analysis

As mentioned in Sect. 4, we used fictitious play for the bidding learning process. After some arbitrary initial bidding at the first stage of the game, the buyers myopically choose their best responses against the empirical action distribution of other buyers' bids at every subsequent stage. Buyers hope that such a bidding learning process will converge and lead to a NE to increase their utility. Figure 7 shows the bidding learning process among 10, 100, and 500 buyers without considering trust. After each stage t, buyers update their actions (bids) based on the outcome of previous stages and observation of other bidders. We can notice that by increasing the number of buyers, the expected utilities increase as well. This will lead to the point where buyers can learn bidding strategies swiftly and converge to a NE to maximize their expected utilities. From Fig. 7, we can see that buyers are learning from the outcome and observation of other buyers in the previous stage and are increasing their stage utility. Furthermore, by increasing the number of buyers, we can see that buyers converge to their NE profile.

We conducted a box-plot presentation to show the learning process of different data buyers using fictitious play. Figure 8 shows the utility with 10, 100, and 500 data buyers competing with each other repeatedly. For each box plot, the central mark indicates a median of utilities. The upper whiskers show the highest utility and the lowest whiskers show the lowest utility at each stage. The outliers are indicated by a (∘) symbol.

7.7 Truthfulness Analysis

We evaluate the property of truthfulness in which each winning buyer i must bid their true valuation. Let's assume the following two cases:

(1) We define $\tilde{\beta}_i$ as overbid from the valuation, and β_i^* as the best response, and $\tilde{u}_i$ and $\bar{u}_i$ as their stage utilities, respectively. Buyer i is the winner when submitting either $\tilde{\beta}_i$ or β_i^* at stage t. However, overbidding creates extreme penalties with respect to trust scores in the payment stage in our model, which leads the

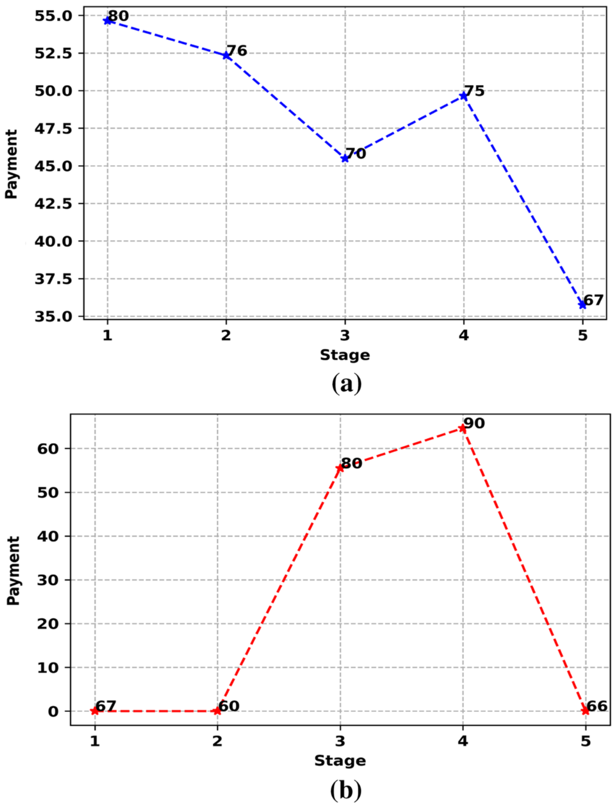

Fig. 9 Truthfulness property

winning buyer to pay more and gain less utility than if he/she plays their best response. The best response scheme given by Eq. 13 is incentive-compatible, when buyer i is repeatedly best-responding, in which case other buyers are incentivized to do the same to maximize their utilities. In other words, a buyer $i \in B$ cannot increase its utility by overbidding, since he/she will be punished severely.

(2) Buyer i will lose the game G if he/she bids lower than their valuation (under-bidding), otherwise would win if he/she reported the true valuation and played their best response.

Therefore, buyer i cannot increase its utility by providing untruthful bidding (overbidding and under-bidding), no matter what the other buyers' bid. For the experiment, we select two buyers; the first buyer bids truthfully while the other bids untruthfully. To provide a consistent environment for comparison, we set the trust score for both buyers to 0.7 and ask price to 50, 48, 41, 40, and 30, respectively. Figure 9a is the result when winner buyer i is biding his/her true valuation and pays the price p_i at stage t. Figure 9b is the result when buyer i bids untruthfully. We can see that buyer i receives zero payment when he/she is underbidding,

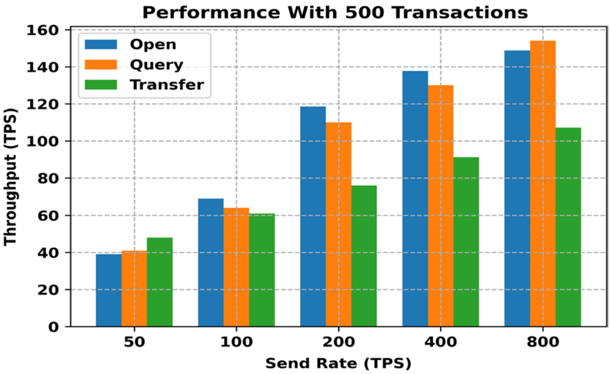

Fig. 10 Transaction throughput under open, query, and transfer workloads with 50, 100, 200, 400, and 800 sending rates

which means that he/she receives zero utility. At stages 3 and 4, he/she is overbidding, and he/she is the winner. However, we can see that the winner is paying much more than if he bids truthfully. Truthfulness property provides the best possible utility for the buyers and ensures there is no incentive for a buyer to bid untruthfully.

7.8 Transaction Throughput Metric

Transaction throughput is the average number of transactions per second that can be written on the ledger. The transaction throughput of the blockchain network calculated as follows [43]:

$$Transaction\ throughput = \frac{Total\ transcations}{Total\ time\ in\ seconds} \tag{41}$$

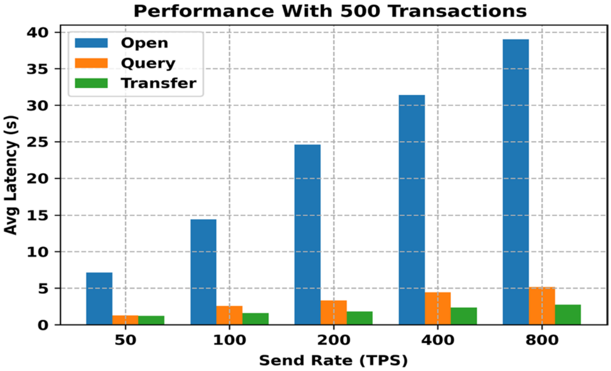

Fig. 11 Transaction throughput under open, query, and transfer workloads with 50, 100, 200, 400, and 800 sending rates

Table 4 Creating the data in CouchDB

Data size (MB)	Max latency (s)	Avg latency (s)	Throughput
3.2MB	12.88	7.50	803.3
5.4MB	19.87	11.13	890.9
7.4MB	29.95	17.05	997.5

Table 5 Retrieval the data from CouchDB

Data size (MB)	Max latency (s)	Avg latency (s)	Throughput
3.2MB	11.11	6.41	792.33
5.4MB	14.31	10.14	819.91
7.4MB	25.41	15.01	935.62

Figure 10 shows the transaction throughput with 50, 100, 200, 400, and 800 sending rates. From the results in Fig. 10, we observe that by increasing the sending rate, the throughput of the transfer workload increased linearly, and the throughput of the open and transfer workload increased slightly too.

7.9 Transaction Metric

Transaction latency is the amount of time taken from the time when a transaction is submitted till the time when it is confirmed and available on the blockchain. Figure 11 indicates the average latency for 500 different sending rates of 50, 100, 200, 400, 800 tps. From Fig. 11, we can see that by increasing the sending rate, the average latency of the open workload increases with the sending rate. Sending a large number of transactions with higher sending rates would cause a failure in the network. Many other things can also cause transaction failures in the blockchain network, such as chaincode (smart contract) logic, version errors, peer resources, policy failures, network resources, consensus errors, and repeated transactions, to name a few. For query and transfer workload, we can see that when the sending rate increases, the average latency increases slightly.

7.10 Elapsed Time Metric

Elapsed time is the amount of time that each buyer needs to interact with the blockchain network to query data from smart contract. This metric measures how much time it takes for the buyer to get query result from database. By default, the underlying data structure of a blockchain does not support an effective method of querying the stored data. To overcome this limitation, we modeled the data in JSON format through CouchDB and deployed indexes. Tables 4 and 5 show the benchmark results of creating and retrieval data within CouchDB, respectively. We can see the average latency increases with the size of the data. We ran the test queries on data to

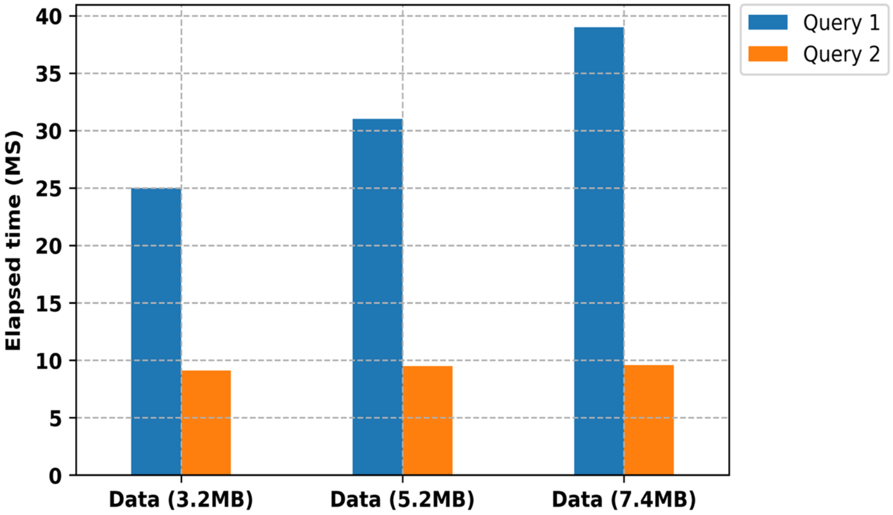

Fig. 12 Running times of test queries

calculate the elapsed time. The elapsed time begins when the query is executed, and ends when the query is returned. Figure 12 shows running times of test queries without index and with index. From Fig. 12, we can observe that querying with index takes less elapsed time compared with querying without index. In general, queries without index will have a longer elapsed time. Indexes enable a database to be queried faster and more efficiently.

7.11 Resource Consumption Metric

Resource consumption metric measures the computing resources consumed by the blockchain network through different operations. Finally, table 6 shows the resource consumption for open, query, and transfer workloads with 500 transactions. Memory displays the amount of memory used by the docker container on each test round. Memory(MAX) measures the maximum resources spent on a transaction, and memory(AVG) measures the average resources spent on all transactions. CPU displays the amount of CPU used by the docker containers during the test round. CPU(MAX) measures the maximum resources spent on a transaction, and CPU(AVG) measures the average resources spent on all transactions. For both 500 transactions, the resource consumption for open and query workloads reveals that CouchDB consumes the most memory and CPU, followed by the peers. The network usage is conducted based on the traffic input and traffic output parameters. The majority of the network traffic is occupied by the orderer since it seeks the consensus in the network in an open workload. During our experiments, we observed that disk read is zero bytes, as there is no need to perform read operations on the ledger in open, and transfer workloads. The query transaction reads the data from the CouchDB in query workload. The performance analysis

Table 6 Resource consumption for open, query, and transfer workloads with 500 transactions

Type	Name	Memory (Max)	Memory (Avg)	CPU% (Max)	CPU% (Avg)	Traffic input	Traffic out	Disc read/write
Open workload								
Docker	peer0.org2.example.com	60.7MB	59.3MB	4.41	3.48	3.1MB	2.4MB	0B & 8.8MB
Docker	peer0.org1.example.com	30.7MB	25.1MB	4.5	2.65	2.9MB	2.2MB	0B & 8.7MB
Docker	orderer.example.com	7.5MB	6.7MB	2.46	1.04	2.9M	4.6M	0B & 5.5MB
Docker	couchdb.org2.example.com	79.1MB	71.5MB	39.1	28.7	3.9MB	3.5MB	0B & 4.1MB
Docker	couchdb.org1.example.com	72.5MB	67.1MB	34.5	24.4	3.5MB	3.2MB	0B & 3.8MB
Docker	ca.org2.example.com	32.1MB	29.9MB	2.7	2.5	1.4MB	1.1MB	0B & 1.2MB
Docker	ca.org1.example.com	29.5MB	24.7MB	2.1	1.7	1.2MB	1MB	0B & 1.0MB
Query workload								
Docker	peer0.org2.example.com	37.1MB	35.1MB	2.22	1.48	1.5MB	1.1MB	0B
Docker	peer0.org1.example.com	19.1MB	10.1MB	1.9	1.08	1.0MB	0.952B	0B
Docker	orderer.example.com	4.9MB	4.3MB	1.21	0.53	1.3MB	0.592B	0B
Docker	couchdb.org2.example.com	93.4MB	87.3MB	63.2	40.5	5.9MB	5.7MB	1.4M & 0B
Docker	couchdb.org1.example.com	90.1MB	84.4MB	51.8	34.3	5.6MB	5.5MB	1.2M & 0B
Docker	ca.org2.example.com	26.3MB	22.9MB	1.7	1.5	1.0MB	606B	0B
Docker	ca.org1.example.com	18.1MB	10.7MB	1.5	1.1	876B	512B	0B
Transfer workload								
Docker	peer0.org2.example.com	33.2MB	29.4MB	2.12	1.01	1.2MB	1.0MB	0B & 4.3MB
Docker	peer0.org1.example.com	15.1MB	10.4MB	1.4	0.78	1.0MB	0.702B	0B & 4.2MB
Docker	orderer.example.com	11MB	8.2MB	1.64	1.04	914B	553B	0B & 4.5MB
Docker	couchdb.org2.example.com	60.6MB	58.7MB	25.3	17.0	2.0MB	1.7MB	0B & 2.8MB
Docker	couchdb.org1.example.com	54.1MB	48.3MB	21.1	13.8	1.6MB	1.5MB	0B & 2.5MB
Docker	ca.org2.example.com	24.7MB	20.4MB	1.2	1.1	1.1MB	796B	0B & 1.4MB
Docker	ca.org1.example.com	17.4MB	11.5MB	1.1	1.0	776B	552B	0B & 1.1MB

shows considerably low memory and CPU consumption. The peer node consumes an average of 59.3MB for memory and 4.41% for CPU in 500 transactions. This depicts that this blockchain network can be easily deployed in real-world applications with low-cost hardware.

8 Conclusion and Future Work

While the economic value of IoT data is increasing, it is not very well known how these data can be conceptualized, measured, and monetized in IoT data markets that enable data owners to trade their data. Unfortunately, the existing IoT data markets are insufficient for capitalizing on the full value of the data in a trusted and transparent way. To address these challenges, we proposed a trustful data trading framework using the game theory approach in an infinitely repeated horizon to enable secure and efficient data trading between buyers and sellers. To model the data market, this paper proposed a non-cooperative infinity repeated game model between rational data buyers. In each stage of the game, buyers hold a bid for a traded amount of records and seek to maximize their expected utility through learning from the outcome of previous stages considering discounted rates for the future utility. We proved NE and the uniqueness of our model, which is derived theoretically for the one-shot game, finite, and infinite horizon games, respectively. Besides, this model imposes a penalty on those buyers who do not have a good reputation and decreases their chance of winning to preserve the data owner's privacy. Through theoretical and security analysis, the paper showed that the proposed system is computationally efficient, soundness, completeness, truthful, budget balance, and individually rational. We implemented our blockchain network using Hyperledger Fabric. We measure the system performance using different metrics. Currently, there is a limitation that buyers can not increase their trust score. We completely understand that this might lead to the starvation of other buyers who are eager to win deals but unable to reach the highest score. We intend to propose a new mechanism for buyers to maximize their trust scores in future work. We also plan to add more data aggregators in the system, in which they can compete with each other to collect IoT data from data owners and sell them to buyers.

Acknowledgements We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Bebortta, S., Singh, A.K., Pati, B., Senapati, D.: A robust energy optimization and data reduction scheme for iot based indoor environments using local processing framework. J. Netw. Syst. Manage. **29**(1), 1–28 (2021)
2. International Data Corporation. The growth in connected iot devices is expected to generate 79.4zb of data in 2025, according to a new idc forecast. https://www.idc.com/getdoc.jsp?containerId=prUS45213219, 2020. Accessed 14 Dec 2020.

3. Hassan, M.U., Rehmani, M.H., Chen, J.: Privacy preservation in blockchain based iot systems: Integration issues, prospects, challenges, and future research directions. Fut. Gener. Comput. Syst. **97**, 512–529 (2019)
4. Yassine, A., Shirmohammadi, S.: Privacy and the market for private data: a negotiation model to capitalize on private data. In: 2008 IEEE/ACS International Conference on Computer Systems and Applications, pp. 669–678. IEEE (2008)
5. Opher, A., Chou, A., Onda, A., Sounderrajan, K.: The Rise of the Data Economy: Driving Value Through Internet of Things Data Monetization. IBM Corporation, Somers, NY (2016)
6. Rashid, Z., Noor, U., Altmann, J.: Economic model for evaluating the value creation through information sharing within the cybersecurity information sharing ecosystem. Fut. Gener. Comput. Syst. **124**, 436 (2021)
7. Rodrigues, B., Scheid, E., Killer, C., Franco, M., Stiller, B.: Blockchain signaling system (bloss): cooperative signaling of distributed denial-of-service attacks. J. Netw. Syst. Manage. **28**(4), 953–989 (2020)
8. Fernandez, R.C., Subramaniam, P., Franklin, M.J.: Data market platforms: trading data assets to solve data problems [vision paper]. arXiv, pp. arXiv–2002 (2020)
9. Lopez, D., Farooq, B.: A multi-layered blockchain framework for smart mobility data-markets. Transp. Res. Part C: Emerg. Technol. **111**, 588–615 (2020)
10. Xiao, Y., Zhang, N., Li, J., Lou, W., Hou, Y.T.: Privacyguard: enforcing private data usage control with blockchain and attested off-chain contract execution. In: European Symposium on Research in Computer Security, pp. 610–629. Springer (2020)
11. An, B., Xiao, M., Liu, A., Xu, Y., Zhang, X., Li, Q.: Secure crowdsensed data trading based on blockchain. IEEE Trans. Mobile Comput. (2021). https://doi.org/10.1109/TMC.2021.3107187
12. Si, H., Sun, C., Li, Y., Qiao, H., Shi, L.: Iot information sharing security mechanism based on blockchain technology. Fut. Gener. Comput. Syst. **101**, 1028–1040 (2019)
13. Khezr, S., Benlamri, R., Yassine, A.: Blockchain-based model for sharing activities of daily living in healthcare applications. In: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp. 627–633. IEEE (2020)
14. Oh, H., Park, S., Lee, G.M., Choi, J.K., Noh, S.: Competitive data trading model with privacy valuation for multiple stakeholders in iot data markets. IEEE Internet Things J. **7**(4), 3623–3639 (2020)
15. Oh, H., Park, S., Lee, G.M., Heo, H., Choi, J.K.: Personal data trading scheme for data brokers in iot data marketplaces. IEEE Access **7**, 40120–40132 (2019)
16. Tian, L., Li, J., Li, W., Ramesh, B., Cai, Z.: Optimal contract-based mechanisms for online data trading markets. IEEE Internet Things J. **6**(5), 7800–7810 (2019)
17. Cao, X., Chen, Y., Liu, K.J.R.: Data trading with multiple owners, collectors, and users: an iterative auction mechanism. IEEE Trans. Signal Inf. Process. Netw. **3**(2), 268–281 (2017)
18. Khokhar, R.H., Iqbal, F., Fung, B.C.M., Bentahar, J.: Enabling secure trustworthiness assessment and privacy protection in integrating data for trading person-specific information. IEEE Trans. Eng. Manage. **68**, 149 (2020)
19. Liu, K., Qiu, X., Wuhui Chen, X., Chen, Z.Z.: Optimal pricing mechanism for data market in blockchain-enhanced internet of things. IEEE Internet Things J. **6**(6), 9748–9761 (2019)
20. Li, Y., Li, L., Zhao, Y., Guizani, N., Yu, Y., Du, X.: Toward decentralized fair data trading based on blockchain. IEEE Netw. **35**, 304 (2020)
21. Sheng, D., Xiao, M., Liu, A., Zou, X., An, B., Zhang, S.: Cpchain: a copyright-preserving crowdsourcing data trading framework based on blockchain. In: 2020 29th International Conference on Computer Communications and Networks (ICCCN), pp. 1–9. IEEE (2020)
22. Dai, W., Dai, C., Choo, K.K.R., Cui, C., Zou, D., Jin, H.: Sdte: A secure blockchain-based data trading ecosystem. IEEE Trans. Inf. Forensics Secur. **15**, 725–737 (2019)
23. Xiong, W., Xiong, L.: Smart contract based data trading mode using blockchain and machine learning. IEEE Access **7**, 102331–102344 (2019)
24. Thu, T., Hien, T., Almeida, M., Karame, G., Soriente, C.: Towards secure and decentralized sharing of iot data. In: 2019 IEEE International Conference on Blockchain (Blockchain), pp. 176–183. IEEE (2019)
25. Parno, B., Howell, J., Gentry, C., Raykova, M.: Pinocchio: nearly practical verifiable computation. In: 2013 IEEE Symposium on Security and Privacy, pp. 238–252. IEEE (2013)

26. Zhang, Y., Deng, R.H., Shu, J., Yang, K., Zheng, D.: Tkse: Trustworthy keyword search over encrypted data with two-side verifiability via blockchain. IEEE Access **6**, 31077–31087 (2018)
27. Yassine, A., Shirehjini, A.A.N., Shirmohammadi, S.: Smart meters big data: Game theoretic model for fair data sharing in deregulated smart grids. IEEE Access **3**, 2743–2754 (2015)
28. Boyd, S., Boyd S.P., Vandenberghe L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
29. Hoang, D.T., Lu, X., Niyato, D., Wang, P., Kim, D.I., Han, Z.: Applications of repeated games in wireless networks: a survey. IEEE Commun. Surv. Tutor. **17**(4), 2102–2135 (2015)
30. Levine, D.K.: Learning in games (2001)
31. Fudenberg, D., Levine, D.: Learning in games. Eur. Econ. Rev. **42**(3–5), 631–639 (1998)
32. Leyton-Brown, K., Shoham, Y.: Essentials of game theory: a concise multidisciplinary introduction. Synth. Lect. Artif. Intell. Mach. Learn. **2**(1), 1–88 (2008)
33. Dasgupta, P., Maskin, E.: The existence of equilibrium in discontinuous economic games, i: theory. Rev. Econ. Stud. **53**(1), 1–26 (1986)
34. Wang, Y., Saad, W., Han, Z., Poor, H.V., Başar, T.: A game-theoretic approach to energy trading in the smart grid. IEEE Trans. Smart Grid **5**(3), 1439–1450 (2014)
35. Yates, R.D.: A framework for uplink power control in cellular radio systems. IEEE J. Sel. Areas Commun. **13**(7), 1341–1347 (1995)
36. Sorin, S.: On repeated games with complete information. Math. Oper. Res. **11**(1), 147–160 (1986)
37. Laraki, R., Renault, J., Sorin, S.: Mathematical Foundations of Game Theory. Springer, Berlin (2019)
38. Das, A., Islam, M.M.: Securedtrust: a dynamic trust computation model for secured communication in multiagent systems. IEEE Trans. Depend. Secur. Comput. **9**(2), 261–274 (2011)
39. Hyperledger.org. Hyperledger caliper: a blockchain performance benchmark framework. https://github.com/hyperledger/caliper (2021). Accessed 26 Aug 2021
40. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct nizks without pcps. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 626–645. Springer (2013)
41. Graf, M., Küsters, R., Rausch, D.: Accountability in a permissioned blockchain: formal analysis of hyperledger fabric. In: 2020 IEEE European Symposium on Security and Privacy (EuroS &P), pp. 236–255. IEEE (2020)
42. Liu, D., Ni, J., Huang, C., Lin, X., Shen, X.S.: Secure and efficient distributed network provenance for iot: A blockchain-based approach. IEEE Internet Things J. **7**(8), 7564–7574 (2020)
43. Hyperledger Performance, Scale Working Group, et al.: Hyperledger blockchain performance metrics. *Hyperledger. org*, pp. 1–17 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Seyednima Khezr received his B.Sc. and M.Sc. degree in Software Engineering from Islamic Azad University, Iran, in 2013 and 2016, respectively. He received his Ph.D. degree in Electrical and Computer Engineering from Lakehead University, Thunder Bay, Ontario, Canada. His research interests include blockchain technology, reputation systems, IoT data trading, software security, and game theory.

Abdulsalam Yassine received his B.Sc. in electrical engineering from Beirut Arab University, Lebanon, in 1993, and his M.Sc. and Ph.D. in electrical and computer engineering from the University of Ottawa, Canada, in 2004 and 2010, respectively. He is a Associate Professor in the Software Engineering Department, Lakehead University, Canada. His research interests are energy informatics, smart cities, and blockchain applications.

Rachid Benlamri is a Professor of Software Engineering and Vice President Academic at the University of Doha for Science and Technology, Qatar. He received his Master's degree and Ph.D. in Computer Science from the University of Manchester - UK. He served as keynote speaker and general chair for many international conferences. His research interests are in AI, Semantic Web, Blockchain, Data Science and Knowledge Engineering.

Authors and Affiliations

Seyednima Khezr[1] · **Abdulsalam Yassine**[2] · **Rachid Benlamri**[2,3]

Abdulsalam Yassine
ayassine@lakeheadu.ca

Rachid Benlamri
rachid.benlamri@udst.edu.qa

[1] Department of Electrical and Computer Engineering, Lakehead University, 955 Oliver Road, Thunder Bay, ON P7B 5E1, Canada

[2] Department of Software Engineering, Lakehead University, 955 Oliver Road, Thunder Bay, ON P7B 5E1, Canada

[3] University of Doha for Science and Technology, 24449 Arab League St, Doha, Qatar