# Secure Crowdsensed Data Trading Based on Blockchain

Baoyi An, Mingjun Xiao, *Member, IEEE,* An Liu, *Member, IEEE,* Yun Xu, *Member, IEEE,* Xiangliang Zhang, *Senior Member, IEEE,* and Qing Li, *Senior Member, IEEE*

**Abstract**—Crowdsensed Data Trading (CDT) is a novel data trading paradigm, in which each data consumer can publicize its data demand as some crowdsensing tasks, and some mobile users (*i.e.*, data sellers) can compete for these tasks, collect the corresponding data, and sell the results to the consumers. Existing CDT systems generally depend on a data trading broker, which will inevitably cause data consumers' concerns on the trustworthiness of the systems and truthfulness of the data. To address this problem, we propose a Blockchain-based Crowdsensed Data Trading (BCDT) system, mainly containing a smart contract, called BCDToken. First, we replace the data trading broker with blockchain to guarantee the trustworthiness of the data trading. Meanwhile, BCDToken adopts Blockchain-based Reverse Auction (BRA) to assign sensing tasks to data sellers. The BRA mechansim holds truthfulness and individual rationality, which can ensure the data sellers to report data collection costs honestly and prevent sellers to manipulate the auction. Moreover, we implement a Secure Truth Discovery and reliability Rating (STDR) mechanism in BCDToken based on homomorphic cryptography, which can incentivize sellers to upload the truthful data and consumers to rate truthfully the reliabilities of sellers based on the collected data without revealing any privacy of data. Additionally, we also deploy BCDToken on an Ethereum test network to demonstrate its practicability and significant performances.

**Index Terms**—Blockchain, Crowdsourcing, Data trading, Mobile crowdsensing, Privacy, Reverse auction, Truth discovery.

## 1 Introduction

Many online data trading systems [2] have emerged in recent years due to the huge potential economic value of data resources, such as CitizenMe, DataExchange, Datacoup, Factual, and Terbine, etc., whereby data consumers can search and purchase data they are interested in. However, most data in the real world are preserved by few research institutions or companies only for their own analysis purposes rather than sharing them with others who have data needs but cannot afford to collect data by themselves, leading to limited volumes of data in trading systems. It is hard to acquire appropriate data from these trading systems, which has significantly suppressed the users' willingness to use these systems. To tackle this problem, a novel data trading paradigm, called Crowdsensed Data Trading (CDT), is proposed, in which the mobile crowdsensing technology is adopted to provide data resources for trading, *i.e.*, a large crowd of mobile users are leveraged to collect data with their smart phones [3]–[5].

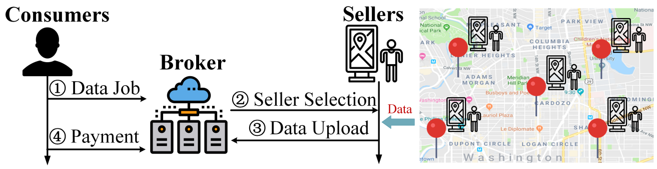


Fig. 1: An Example of the CDT System

Basically, a typical CDT system (*e.g.*, Thingful [6], Thingspeak [7]) consists of a data trading broker, some data consumers, and data sellers (a.k.a., crowdsensing workers). As shown in Fig. 1, the consumer publishes data collection job via broker to employ a large amount of sellers to collect data with their mobile phones according to some requirements (*e.g.*, collecting scopes). Then, sellers sell the sensed data to the consumer via broker. So far, there have been a few works focusing on the CDT system design. For example, [8] introduces a framework HORAE to trade private data with temporal correlations from the perspective of a data broker, which can guarantee balance and avoid arbitrage. A profit-driven data collection framework is proposed in [9] for crowdsensed data markets, in which a data procurement auction is adopted to determine the minimum payment for each data collection. The sellers and consumers propose their selling and buying quantities, respectively, to match the market supply and demand in the brokerage-based data market [10]. However, these CDT systems have to rely on a third-party as the data broker, which causes data consumers'

- *B. An, M. Xiao and Y. Xu are with the School of Computer Science and Technology / Suzhou Institute for Advanced Research, University of Science and Technology of China, Hefei, China.*
  *Email: xiaomj@ustc.edu.cn (Corresponding Author)*
- *A. Liu is with the Department of Computer Science and Technology, Soochow University, Suzhou, China.*
  *Email: anliu@suda.edu.cn*
- *X. Zhang is with the Division of Computer, Electrical and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology, Jeddah, Saudi Arabia.*
  *Email: xiangliang.zhang@kaust.edu.sa*
- *Q. Li is with the Department of Computing, the Hong Kong Polytechnic University, Hong Kong, China.*
  *Email: csqli@comp.polyu.edu.hk*

*This paper is an extended version of the conference paper [1] published in DASFAA 2019. This research was supported in part by the National Natural Science Foundation of China (NSFC) (Grant No. 61872330, 61572457, 61572336, 61632016, U1709217, 61379132), the National Key R&D Program of China (Grant No. 2018AAA0101204), the Natural Science Foundation of Jiangsu Province in China (Grant No. BK20191194, BK20131174, BK2009150), the Natural Science Research Project of Jiangsu Higher Education Institution (No. 18KJA520010), and Anhui Initiative in Quantum Information Technologies (Grant No. AHY150300).*

concerns on the security of the whole data trading, and may prevent them from using the systems.

On the other hand, blockchain [11], a newly-emerging decentralized transaction recording technology, shows a glimpse of solutions to fairness and transparency issues. Data exchanges or transactions among mutually distrusted users can be securely conducted on blockchains with no need of a centralized trusted intermediary, which can avoid high legal and transactional costs [12]. There exist some special complex programs deployed on blockchains, called smart contracts [13], which can automatically execute operations according to trading treaty conditions and enforce the participants to fulfill their obligations. Hence, smart contract can be introduced into CDT systems as a trusted broker to conduct the data trading between sellers and consumers. For example, the CDT framework in [14] sells the aggregated results to consumers and pays for sellers by their weight shares of results on the blockchain, which protects data privacy by storing data separately in two semi-honest cloud severs. Even though the blockchain technology used in CDT systems can eliminate the influence of the third-party data broker to some extent, there still are some challenges for implementing a secure CDT system.

**First, untrusted participant**. From the system perspective, malicious participants (i.e, consumers or sellers) may arbitrarily drop out or break the contracts halfway, causing losses to others. Therefore, we devise some modifiers, a kind of smart contract specific technology, and set life span limitations for each procedure of the data trading to ensure all participants to comply with the prescribed trading rules.

**Second, untruthful cost**. Untruthful sellers might report fake data collection costs to achieve more rewards. As we know, many auction mechanisms can ensure sellers to report their costs honestly in crowdsensing systems [15]–[18]. So we design a Blockchain-based Reverse Auction (BRA) mechanism to guarantee the truthfulness of costs.

**Third, untruthful data**. The consumers can use truth discovery technique to derive the truth of data from noisy data collected by different sellers [19], where the data truth is a true data value that should be sensed to each collection task, regarded as an unknown stochastic variable. But the sellers might be untruthful to report inaccurate or low quality data. We define a reliability metric for each seller to indicate its collected data quality and rate each seller to prompt him to report high quality data.

**Fourth, untruthful rating**. The rating scores from consumers imply the overall data quality of sellers and are used to iteratively update sellers' reliabilities. However, the consumers may not rate truthfully the sellers, such that we design a Secure Truth Discovery and reliability Rating (STDR) mechanism to force consumers to give truthful rating scores based on the data. Additionally, the privacy-preserving of the sensed data is also critical in STDR.

Based on the above ideas, we propose a Blockchain-based Crowdsensed Data Trading (BCDT) system in this paper, which mainly contains a smart contract, called BCDToken. BCDToken as the broker will execute the BRA mechanism for seller selection and the STDR mechanism over any numerical data for truth discovery and privacy-preserving truthful rating. As shown in Fig. 1, any consumer can start a CDT by issuing the data demand via BCDToken (*e.g.*, report traffic conditions of multiple locations), and the sellers who have registered in BCDToken can bid for their interested tasks. BCDToken runs automatically to determine the winners and payments. After the sellers collect and upload data, the consumer can find the truth of data, rate the sellers via STDR, and pay rewards to the sellers via BCDToken.

Overall, the major contributions are as follows:

1) We design a BCDT system, which mainly contains a smart contract embedded with the BRA and STDR mechanisms, *i.e.*, BCDToken. By employing BCDToken as the data trading broker, BCDT can ensure the trustworthiness of trading process, the privacy-preserving of sensed data and the derived truth, as well as the truthfulness of sensed data, the sensing costs, and the rating scores on sellers' reliability. To the best of our knowledge, BCDT is the first system that can obtain these security properties simultaneously.
2) We propose a BRA mechanism to select sellers and determine payments for data trading by letting blockchain serve as the reverse auctioneer and adopting a two-step bidding strategy. BRA can ensure that all sellers follow the workflow of auction and report their costs truthfully. Moreover, no one can manipulate and benefit from the reverse auction by eavesdropping others' bids during the trading.
3) We propose a STDR mechanism to provide the functionalities of privacy-preserving truth discovery and reliability rating for data trading based on the homomorphic encryption and data hiding techniques. STDR can incentivize sellers to report the truthful sensed data and force the consumer to rate the reliability truthfully based on the data, during which the data privacy can be protected from leakage.
4) We implement a prototype of BCDT and deploy BCDToken to an official Ethereum test network. Extensive simulations are conducted to demonstrate the significant performances and the practicability of BCDToken.

The paper is organized as follows. We introduce some preliminaries and design goals in Sec. 2, and present a system model of BCDT in Sec. 3. The BRA and STDR mechanisms are elaborated in Sec. 4 and Sec. 5, respectively. System analyses are carried on in Sec. 6. We present simulations and evaluations in Sec. 7. Finally, we review the related works in Sec. 8 and conclude in Sec. 9.

## 2 Preliminaries and Design Goal

### 2.1 Smart Contacts

In this paper, we deploy a smart contact on the blockchain to act as the broker of data trading. Essentially, smart contact is a kind of special program, which can be automatically executed on the blockchain, so as to enforce all participants to fulfill their obligations. Moreover, the consensus protocols of blockchains can guarantee the execution correctness. Each blockchain user, identified by its account (blockchain address), can send transactions to interact with smart contracts. Each transaction refers to a signed data package, called *msg* in smart contracts, and has a recipient. It also has a *VALUE* field which is the amount of **wei** to be transferred from sender to recipient. **ether** is a digital currency in blockchain: 1 **ether** $=10^{18}$ **wei**. When a smart contract receives a transaction (*msg*), it can obtain two parameters: 1) **msg.sender**: the sender's account. 2) **msg.value**: the amount of **wei**. We also specify three features of smart contracts:

- **Timing.** A smart contract has a time clock which is modeled as a continuously increasing variable $now$. $now$ is an alias of the timestamp on the blockchain.
- **Function Modifier.** Modifiers are inheritable properties of smart contracts which are used to automatically check a prior condition to executing the function.
  - *"payable"* is a reserved keyword in smart contracts and a kind of modifier. Functions with "payable" are able to receive ether while being called.
  - *"require"* can check for conditions and throw an exception if the condition is not met, such as user inputs, the responses from contracts, and the state conditions prior to state changing operations.
- **Event.** Events facilitate communication between smart contracts and their user interfaces. In traditional web development, a server response is provided in a callback to the frontend. In blockchains, events can be generally considered as asynchronous triggers with data. When a contract wants to trigger the frontend, the contract emits an event. As the frontend is watching for events, it can take actions, display messages, etc.

## 2.2 Homomorphic Encryption

The homomorphic encryption technique is used in this paper to realize the trustworthy evaluation on each seller's reliability and protect the data privacy, defined as follows.

**Definition 1 (Homomorphic Encryption [20]).** *A homomorphic encryption scheme is a public-key cryptosystem with such a homomorphic property that the "addition" operation can be applied to the encrypted data without decrypting them. Let $\mathcal{Z}_q$ be a prime field, $\otimes$ and $\oplus$ be the multiplication and addition operations in this field,* i.e., $x\otimes y\overset{def}{=}xy \bmod q$ *and* $x\oplus y\overset{def}{=}x+y \bmod q$ *for* $\forall x,y\in\mathcal{Z}_q$. *Then, the homomorphic encryption scheme satisfies:*

$$E[m_1]\otimes E[m_2]=E[m_1\oplus m_2], \tag{1}$$

*where $m_1,m_2\in\mathcal{Z}_q$ are two plaintexts, and $E[\cdot]$ is the homomorphic encryption operation.*

*Here, we adopt the well-known Paillier cryptosystem [21] for this encryption scheme. For a set of values $M$, we let $E[M]=\{E[m]|m\in M\}$.*

## 2.3 The Security Model

In this paper, we consider such a security model that both of the consumer and sellers are rational participants. That is to say, each participant will follow the whole data trading process if it can benefit from this process; no one is willing to bear a penalty, so as to maliciously destroy the entire data trading. Here, the security is an extended concept in the system level, mainly including three aspects. The first one is individual rationality, defined as follows.

**Definition 2.** ***(Individual Rationality).*** *If each participant can obtain a non-negative profit after completing the data trading, we say the whole system is individually rational.*

Under our security model, if a data trading system meets the above individual rationality property, all sellers and the consumer will follow the whole data trading rule so as to obtain their profits. Despite this, it does not mean that these participants are trustworthy since they still might submit untruthful data, report untruthful data collection cost, or evaluate the data with untruthful scores during data trading. Therefore, our security model also includes the truthfulness.

**Definition 3.** ***(Truthfulness).*** *Let $X_i$ be an input of the $i$-th participant in the data trading, where $0\le i\le n$, $i=0$ represents the consumer, and $1\le i\le n$ represents $n$ sellers. Assume that $x_i$ is the truthful value of $X_i$, but the participant can manipulate its input by submitting another value $x'_i$ for $X_i$. The corresponding profits are denoted by $\mathcal{P}rofit_i(X_i=x_i)$ and $\mathcal{P}rofit_i(X_i=x'_i)$, respectively. Then, if for any $X_i$,*

$$\mathcal{P}rofit_i(X_i=x_i)\ge\mathcal{P}rofit_i(X_i=x'_i), \tag{2}$$

*we say that the data trading system is truthful. Here, $X_i$ might be the data for trading, the cost of collecting data, or the score on evaluating the seller's reliability. Accordingly, the profit is also not limited to economic reward.*

Additionally, the security model also takes the data privacy into consideration. We extend the concept of privacy in [22] and define it as follows.

**Definition 4 (Privacy-Preserving).** *Denote the data trading as a functionality computed by a consumer and $n$ sellers jointly,* i.e., $\mathcal{F}(X_0,X_1,\cdots,X_n)=(\mathcal{F}_0,\mathcal{F}_1,\cdots,\mathcal{F}_n)$, *where $X_i$ and $\mathcal{F}_i$ are the input and output of the $i$-th party ($0\le i\le n$, where $i=0$ represents the consumer and $1\le i\le n$ represents $n$ sellers), both belonging to a prime field $\mathcal{Z}_q$. For $\mathcal{I}=\{i_1,\cdots,i_\kappa\}\subset\{0,\cdots,n\}$, we let $\mathcal{F}_\mathcal{I}$ denote the subsequence $\mathcal{F}_{i_1},\cdots,\mathcal{F}_{i_\kappa}$. Consider a data trading protocol for computing $\mathcal{F}$. Define all messages that the $i$-th party can observe during the execution of the protocol as the view of this party and denote it as $VIEW_i$. Let $VIEW_\mathcal{I}\overset{def}{=}(\mathcal{I},VIEW_{i_1},\cdots,VIEW_{i_\kappa})$. Then, we say that the protocol privately computes $\mathcal{F}$ if there exists a polynomial-time algorithm, denoted as $\mathcal{A}$, such that for every $\mathcal{I}$ above*

$$\mathcal{A}(\mathcal{I},(X_{i_1},\cdots,X_{i_\kappa},\mathcal{F}_\mathcal{I}))\overset{C}{\equiv}VIEW_\mathcal{I}, \tag{3}$$

*where $\overset{C}{\equiv}$ denotes computational indistinguishability.*

*Eq.* (3) *asserts that the view of each party in $\mathcal{I}$ can be efficiently simulated based solely on its inputs and outputs. This implies that no extra information can be derived by others during the execution of the protocol, and thus the privacy can be preserved.*

Now, we define the security model as follows:

**Definition 5.** ***(The Security Model).*** *If a data trading system meets the individual rationality, the collected data, the data collection cost, and the rating score satisfy the truthfulness, and meanwhile the data for trading and the truth derived from data also meet the privacy-preserving property, we say that the system is secure.*

## 2.4 Design Goal

Consider a data trading scenario where a consumer wants to collect sensed data from some Points of Interest (PoIs) during a particular time. Moreover, the consumer has some

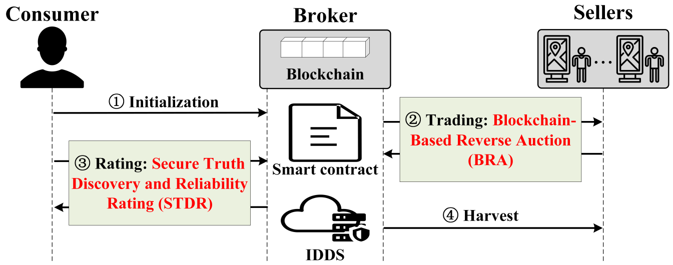

Consumer
Broker
Sellers
Blockchain
① Initialization
② Trading: Blockchain-Based Reverse Auction (BRA)
③ Rating: Secure Truth Discovery and Reliability Rating (STDR)
Smart contract
④ Harvest
IDDS

Fig. 2: The Framework of BCDT

quality requirements for the data collected from each PoI. It is not a cost-effective way to deploy data collection sensors to all PoIs, and the consumer also cannot finish this work by himself manually and individually. So it intends to recruit a crowd of mobile users to collect sensed data and purchase the data from them. Then, our goal is to design a data trading system for them based on blockchain under the above security model. In such a system, the consumer and sellers are constrained to follow the whole data trading process, submit truthful data collection cost, make truthful evaluation, upload truthful data and protect the privacy of trading data and derived truth.

## 3 The BCDT System

In this section, we propose a secure Blockchain-based CDT system, *i.e.*, the BCDT system. This system mainly includes a consumer, some sellers, a smart contract deployed on the blockchain, *i.e.*, BCDToken, and an IPFS-based Distributed Data Storage under the blockchain (IDDS), in which BCDToken acts as a broker to negotiate the data trading between a consumer and some sellers. These components are illustrated in Fig. 2 and also can be defined as follows.

**Definition 6 (Consumer, Job, Tasks and Requirement).** *The consumer is a data service requester who has a data requirement and wants to buy the desired data through the BCDT system. The data requirement can be represented as a data collection job, composed of a series of sensing tasks,* i.e., $Job \overset{def}{=} \langle \mathcal{T}, \boldsymbol{\epsilon}, \tau_{start}, \tau_{end} \rangle$. *Here, $\mathcal{T}$ includes $l$ location-related sensing tasks, each of which corresponds to a PoI, denoted as $\mathcal{T}=\{t_1, t_2, \cdots, t_l\}$. The sensed data need to be collected from the $l$ PoIs between $\tau_{start}$ and $\tau_{end}$, where $\tau_{start}$ and $\tau_{end}$ are the earliest start-time and the latest finish-time of the data collection respectively. $\boldsymbol{\epsilon}=\{\epsilon_j | t_j \in \mathcal{T}\}$ are the minimum quality requirements of the sensed data. That is, the quality of task $t_j$'s sensed data is no less than $\epsilon_j$.*

**Definition 7 (Seller, Cost, and Reliability).** *Sellers are a crowd of mobile workers who participate in the job, denoted by $\mathcal{W}=\{w_1, w_2, \cdots, w_n\}$, where $n$ is the number of sellers. Each seller $w_i$ has an interested tasks subset $\mathcal{T}_i$ ($\subseteq \mathcal{T}$). Performing the sensing tasks in $\mathcal{T}_i$ will produce a cost, denoted by $c_i$. Moreover, the seller $w_i$ has a reliability value $r_i$ which can be seen as the quality of its sensed data. Here, we also denote the sellers who execute $t_j$ by $\mathcal{W}_j$ ($\subseteq \mathcal{W}$).*

**Definition 8 (IPFS-based Distributed Data Storage, IDDS).** *IDDS is a distributed data storage system based on the Inter-Planetary File System (IPFS) protocol, which can use content-addressing to uniquely identify each data file in a peer-to-peer distributed data sharing network. In BCDT, sellers will encrypt their collected data and store the encrypted data on IDDS, denoted as $EnData=\{EnData_i | w_i \in \mathcal{W}\}$. IDDS provides a persistent and reliable storage for these encrypted data. The consumer can download these data according to the returned addresses.*

TABLE 1: Description of major notations

| Variable | Description |
|---|---|
| $w_i$, $t_j$ | the $i$-th seller, the $j$-th task. |
| $l$, $n$ | the number of tasks and the number of sellers. |
| $\mathcal{T}$, $\mathcal{T}_i$ | the set of all tasks and the set of tasks that $w_i$ deals with. |
| $\mathcal{W}$, $\mathcal{W}_j$ | the set of all sellers and the set of sellers who execute $t_j$. |
| $\tau_{start}$, $\tau_{end}$ | the start-time and the finish-time of data collection. |
| $\boldsymbol{\epsilon}$ | the minimum reliability requirements of all tasks. |
| $\beta_i$, $en\beta_i$ | the bid value of $w_i$ and the encrypted bid value of $w_i$. |
| $\mathcal{B}_i$, $en\mathcal{B}_i$ | the bid profile of $w_i$ and the encrypted bid profile of $w_i$. |
| $r_i$, $\mathcal{R}$ | the reliability of $w_i$ and the reliabilities of all sellers. |
| $\mathcal{S}$ | the set of all winners (*i.e.*, selected sellers). |
| $p_i$, $\mathcal{P}$ | the payment of $w_i$ and the payments of all winners. |
| $Data_i$, $Data$ | the sensed data of $\mathcal{T}_i$ and the set of all sensed data. |
| $EnData_i$,<br>$EnData$ | the encrypted data of $\mathcal{T}_i$ and the set of all encrypted data. |
| $o_i$, $O$ | the rating score of $w_i$ and the rating scores of all winners. |
| $pk$, $sk$ | public key and secret key. |
| $E_{pk}[\cdot]$, $D_{sk}[\cdot]$ | homomorphic encryption and decryption functions. |
| $data_{ij}$ | the observed data of $t_j$ from $w_i$. |
| $\mu_j$, $\hat{\mu}_j$ | the truth value of $t_j$ and the estimate truth value of $t_j$. |
| $q_{ij}$, $Q_i$ | the quality of $t_j$ from $w_i$ and the overall quality of $w_i$. |
| $\boldsymbol{\lambda}$, $\boldsymbol{\eta}$ | two matrices to verify the truthfulness of rating scores. |

**Definition 9 (BCDToken).** *BCDToken is a smart contract deployed on the blockchain, acting as a broker of the data trading between the consumer and sellers. It maintains a registry $\mathcal{R}$, a dictionary data type, to record sellers' reliability values,* i.e., *$\mathcal{R}(w_i)=r_i$, which is also a similar metric as in [23]. Besides, it includes two major functionality modules: trading and rating. The first module selects some sellers for the data trading according to the requirements from the consumer and determines the payments for the sellers who complete the sensing tasks. The second module helps the consumer discover the truth values of the data collected by sellers and make the truthful rating for each seller's reliability. $\langle \tau_{bid}, \tau_{reveal}, \tau_{transfer} \rangle$ are three time constraints in BCDToken, where $\tau_{bid}$ and $\tau_{reveal}$ are the given time durations that allow sellers to participate in the trading process, and $\tau_{transfer}$ is the given time duration for sellers to transfer data in the rating process.*

The workflow of BCDT can be roughly divided into four phases: initialization, trading, rating, and harvest, as shown in Fig. 2.

**Phase 1: Initialization.** All participants register in BCDToken, each of whom first offers some ethers no less than $\$deposit$ to BCDToken, where $\$deposit$ is the minimum margin requirement to enter the data trading. When a consumer wants to start a CDT, it will send a transaction (*msg*) which contains the data collection job description (*i.e.*, *Job*) to invoke a function of *Initiate()* in BCDToken. The detailed function is given in Fig. 3. First, it checks whether the deposit offered by the consumer (*msg.value*) is no less than the deposit threshold $\$deposit$, *i.e.*, $msg.value \geq \$deposit$. Then, the function records the account of the sender as the consumer ($msg.sender$), the sensing tasks $\mathcal{T}$, the data quality requirements $\boldsymbol{\epsilon}$, and the consumer's public key $pk$. Next, it sets some parameters of time constraints for the subsequent phases. Finally, the function emits $Notify$ event to inform all registered sellers of the data collection job.

**Phase 2: Trading.** After receiving the notification, sellers

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3107187, IEEE Transactions on Mobile Computing

```
Initiate($\mathcal{T}$, $\epsilon$, $\tau_{start}$, $\tau_{end}$, $pk$) payable:
 1. Require $msg.value \geq \$deposit$.
 2. Set $consumer{=}msg.sender$ and store $\mathcal{T}$, $\epsilon$, $pk$.
 3. Set $t_{bid}{=}now{+}\tau_{bid}$, $t_{reveal}{=}t_{bid}{+}\tau_{reveal}$, $t_{trasfer}{=}\tau_{end}{+}\tau_{trasfer}$.
 4. Trigger $Notify$ event to inform the registered sellers.
```

Fig. 3: The Initiate Function in BCDToken

```
Refund():
 1. Require $msg.sender{=}consumer$.
 2. Update $\mathcal{R}$ with $O$.
 3. Compute the remaining rewards:
      $\$rewards{=}\$rewards - \sum_{w_i \in \mathcal{S}} \mathcal{P}(w_i)$.
 4. Transfer $rewards to $msg.sender$.

Payment():
 1. Require $Bids(msg.sender).\beta_i \neq NULL$.
      ▷ If it is NULL, it is untruthful according to step 4 in Fig. 5.
 2. If $msg.sender \in \mathcal{S}$, transfer $\mathcal{P}(msg.sender)$ to $msg.sender$.
```

Fig. 4: The Refund and Payment Functions in BCDToken

first submit the bids for their desired sensing tasks according to the corresponding costs. Then, BCDToken selects some sellers to conduct the tasks and determines payments. Next, the selected sellers are informed to collect data only after the consumer transfers some ethers as the rewards to BCDToken, which are no less than the total payments. Finally, sellers will encrypt their collected data using the consumer's public key, upload them onto IDDS, and return the corresponding addresses to BCDToken. Here, we propose a Blockchain-based Reverse Auction (BRA) mechanism with two-step bidding to select the sellers and compute the corresponding payments for the data trading, which can ensure sellers to report their data collection costs truthfully. The selected sellers are called the winners of auction, denoted by $\mathcal{S}(\subseteq \mathcal{W})$, and their payments are denoted as $\mathcal{P}{=}\{p_i | w_i {\in} \mathcal{S}\}$. The detailed BRA mechanism is presented in Sec. 4.

**Phase 3: Rating.** The consumer first receives the addresses of the data stored on IDDS from BCDToken. Then, according to the addresses, the consumer downloads the data from IDDS, decrypts them to get the plaintext data using its private key, and further derives the corresponding truth values. Next, based on these truth values and the plaintext data, the consumer gives some rating scores to evaluate sellers' data collection qualities, which are used to update their reliability values. In this phase, we propose a Secure Truth Discovery and reliability Rating (STDR) mechanism on the blockchain, by which the consumer can determine the truth value of each task and give the truthful rating scores, while protecting the privacy of collected data from being revealed. The rating scores are denoted by $O{=}\{o_i | w_i {\in} \mathcal{S}\}$. The detailed STDR mechanism is presented in Sec. 5.

**Phase 4: Harvest**. Finally, the consumer retrieves the remaining rewards and sellers acquire their payments through BCDToken to complete the whole data trading. This is conducted through two functions $Refund()$ and $Payment()$, as shown in Fig. 4. As for the consumer, 1) $Refund()$ requires that the invoker ($msg.sender$) is the consumer and 2) updates reliabilities $\mathcal{R}$ with $O$. 3) $Refund()$ computes the remaining rewards, *i.e.*, $\$rewards{=}\$rewards{-}\sum_{w_i \in \mathcal{S}} \mathcal{P}(w_i)$ and 4) transfers the remaining rewards to the consumer. As for each seller $w_i$, 1) $Payment()$ checks whether the bid $\beta_i$ of invoker $w_i$ ($msg.sender$) is 0, because an untruthful seller's bid is 0 due to the checks in $RevealBid()$. 2) $Payment()$ transfers the payment to $w_i$ if it is verified to be a winner.

Additionally, for ease of reference, we list the major notations in Table 2.

TABLE 2: Description of major notations for BCDToken

| **Variable** | **Description** |
|---|---|
| ether, wei | the virtual currency units on blockchain. |
| $msg$ | the alias of transaction in smart contracts. |
| $msg.sender$ | the sender of $msg$. |
| $msg.value$ | equals to the amount of ethers attached in $msg$. |
| $now$ | the timestamp of blockchain. |
| $payable$ | the specific keyword for mandatory payment. |
| $require$ | the specific keyword for the implementation of modifiers. |
| $\$deposit$ | the refundable deposit for participating in a CDT. |
| $\$rewards$ | the minimum total rewards for sellers. |
| $t_{bid}$, $t_{reveal}$, $t_{transfer}$ | the latest-finish time points of bid commitment, bid reveal and data transfer. |
| $Bids$,$EnBids$ | the bid profiles and the encrypted bid profiles in BCDToken. |

## 4 The BRA mechanism

In this section, we propose the BRA mechanism for BCDT to realize secure data trading. First, BRA adopts the truthful reverse auction mechanism to select sellers and determine the corresponding payments, whereby sellers will report their sensing costs honestly during the data trading. Second, BRA employs the BCDToken smart contract as the reverse auctioneer, which enforces all sellers to follow the workflow of auction, making the trading process trustworthy. Third, BRA utilizes a two-step bidding strategy to prevent sellers from manipulating the reverse auction by eavesdropping others' bids. Here, the BRA mechanism only takes account of the single-minded auction [24], but it can be extended to support other truthful auction mechanisms: multi-minded auction [25], double auction [26], online auction [27], etc. The problem formulation and detailed design of the BRA mechanism are presented as follows.

### 4.1 Problem Formulation

In the trading phase, each seller $w_i \in \mathcal{W}$ will submit a bid profile $\mathcal{B}_i = (\beta_i, \mathcal{T}_i)$ to compete for the sensing tasks, where the bid $\beta_i$ actually is the reward claimed by $w_i$ to compensate for the cost $c_i$ of completing sensing tasks in $\mathcal{T}_i$. In general, the bid $\beta_i$ is not necessarily equal to the cost $c_i$, since the seller $w_i$ might manipulate the claimed cost to obtain more rewards. However, when the auction mechanism is truthful, claiming false costs will not bring any extra reward, and thus the seller will report its cost honestly, *i.e.*, $\beta_i = c_i$. After receiving the bids from sellers, BCDToken will select some sellers to conduct the sensing tasks and determine the corresponding payments through the BRA mechanism. The goal is to minimize the total cost, while ensuring the data quality contributed by the selected sellers no less than the threshold $\epsilon$. More specifically, the seller selection problem can be formulated as follows:

$$Minimize: \quad C(S){=}\textstyle\sum_{w_i \in \mathcal{S}} \beta_i \tag{4}$$

$$Subject\ to: \quad \mathcal{S} \subseteq \mathcal{W} \tag{5}$$

$$\sigma_j^{\mathcal{S}}{=}\textstyle\sum_{w_i \in \mathcal{S} \cap \mathcal{W}_j} r_i \geq \epsilon_j, \ 1 \leq j \leq l \tag{6}$$

Here, Eq. (4) is the optimization objective. In the following subsections, we will show that the BRA mechanism is truthful. Thus, we can directly set $c_i = \beta_i$ in Eq. (4). In

```
CommitBid($\mathcal{T}_i, en\beta_i$) payable:
 1. Require now $\leq t_{bid}$ and $msg.value \geq \$deposit$.
 2. Set $EnBids(msg.sender)$=($\mathcal{T}_i, en\beta_i, msg.value$).
     $\triangleright$ $w_i = msg.sender$, the deposit of $w_i$=$msg.value$.
RevealBid($\mathcal{T}_i, \beta_i, nonce_i$):
 1. Require $t_{bid} \leq$ now $\leq t_{reveal}$.
 2. Require $EnBids(msg.sender).en\beta_i$=sha3($\mathcal{T}_i, \beta_i, nonce_i$).
 3. Require $EnBids(msg.sender).\mathcal{T}_i$=$\mathcal{T}_i$.
 4. Set $Bids(msg.sender) = (\mathcal{T}_i, \beta_i)$.
```

Fig. 5: The Two-Step Bidding Strategy of BRA

Eq. (6), $\sigma_j^{\mathcal{S}}$ is the total reliability of selected sellers (a.k.a., auction winners), which indicates the total data quality contributed by these sellers. Additionally, the above seller selection problem is NP-hard because it can be seen as a complex weighted set coverage problem [28].

## 4.2 The Detailed BRA Mechanism

The design goal of BRA is to make the whole reverse auction process secure. However, traditional auction mechanisms cannot ensure the auctioneer to be trustworthy. Moreover, sellers might not follow the trading rules, *e.g.*, the seller maliciously exits the auction early at will, or the seller manipulates the auction by postponing its own bid and eavesdropping others' bids. Thus, BRA lets BCDToken act as the auctioneer and adopts a two-step bidding strategy to ensure that the whole auction is conducted securely. More specifically, BRA consists of the following three parts:

### 4.2.1 Two-step Bidding

At the beginning of BRA, each seller submits its bid to BCDToken. Due to the transparent characteristic of blockchain, the bid value is publicly visible. To prevent potential adversaries to eavesdrop the bid value and manipulate the auction, we divide the bid commitment into two steps and use two functions to implement this process in Fig. 5.

**Step 1: Commit Encrypted Bid.** The first step bidding is to submit encrypted bid profiles. First, each seller $w_i$ computes an encrypted bid $en\beta_i$ using Secure Hash Algorithm-3 (SHA-3) [29]. SHA-3 takes as input its account $w_i$, its bid $\beta_i$, and a randomly selected $nonce_i$, *i.e.*, $en\beta_i = sha3(w_i, \beta_i, nonce_i)$. Then, the seller sends its encrypted bid profile $en\mathcal{B}_i = (\mathcal{T}_i, en\beta_i)$ to BCDToken. BCDToken uses a function *CommitBid()* to check whether the current time ($now$) upon receiving bid commitment is no larger than the specified latest bid finish time ($t_{bid}$), *i.e.*, $now \leq t_{bid}$. Meanwhile, it also checks whether the seller offers adequate deposit to take part in the auction, *i.e.*, $msg.value \geq \$deposit$. Finally, *CommitBid()* uses the dictionary-type storage $EnBids$, indexed by the seller's account, to record $en\mathcal{B}_i$ and its deposit.

**Step 2: Reveal Real Bid.** The second step bidding is to submit unencrypted bid profiles and verify them using the encrypted versions. First, BCDToken is invoked by the seller $w_i$ to send $\mathcal{T}_i$, $\beta_i$, and $nonce_i$. Then, a function *RevealBid()* is used to check whether the invocation time meets time limits, *i.e.*, $t_{bid} \leq now \leq t_{reveal}$. Also, *RevealBid()* checks the legality of the bid profile, which requires that the revealed bid value $\beta_i$ and tasks subset $\mathcal{T}_i$ are same with the original recorded ones. If $w_i$ passes the time and bid checks, *RevealBid()* will use the dictionary-type storage $Bids$ to record the revealed bid. Otherwise, $w_i$ is untruthful and its deposit will be forfeited.

```
SellerSelection():
 1. Require now $\geq t_{reveal}$ and $msg.sender$ = $consumer$.
 2. Repeat until $G(\mathcal{S}) = \sum_{j=1}^{l} \epsilon_j$:
     1) for $\forall B_i$, compute $\rho_i = \frac{v_i(\mathcal{S})}{\beta_i}$;
     2) Record the index of the maximum $\rho_i$ as $i^*$;
     3) Add $w_{i^*}$ to $\mathcal{S}$, set $\mathcal{C} = \mathcal{C} + \beta_{i^*}$.
 3. Store ($\mathcal{S}, \mathcal{C}$).
Pricing($w_i$):
 1. Require $msg.sender$ = $consumer$ and now $> t_{transfer}$.
 2. Create a empty winner set $\mathcal{S}'$.
 3. Record $(\mathcal{T}_i, \beta_i) = Bids(w_i)$ and set $Bids(w_i) = NULL$.
 4. Repeat until $G(\mathcal{S}') = \sum_{j=1}^{l} \epsilon_j$:
     1) Compute $\rho_k = \frac{v_k(\mathcal{S}')}{\beta_k}$, for $\forall w_k \notin \mathcal{S}'$ and $Bids(w_k) \neq NULL$;
     2) Record the index of the maximum $\rho_k$ as $k^*$;
     3) if $\mathcal{P}(w_i) < \frac{\beta_{k^*} v_i(\mathcal{S}')}{v_{k^*}(\mathcal{S}')}$,  Set $\mathcal{P}(w_i) = \frac{\beta_{k^*} v_i(\mathcal{S}')}{v_{k^*}(\mathcal{S}')}$;
     4) Add $w_{k^*}$ to $\mathcal{S}'$.
 5. Set $Bids(w_i) = (\mathcal{T}_i, \beta_i)$, delete $\mathcal{S}'$.
 6. Trigger $AuctionEnd$ event to inform all winners ($\mathcal{S}$).
```

Fig. 6: The reverse auction of BRA

### 4.2.2 Seller Selection

Since the seller selection problem is NP-hard, there is no optimal solution in polynomial time. Thus, we adopt a greedy strategy to select sellers, which can achieve an approximate optimal solution [28]. The criterion is that the seller who has the largest reliability to execute the most tasks with the least cost will be selected first as an auction winner.

First, we define a reliability truncation function $G(\mathcal{S})$, which indicates the total reliability contributed by the winners in $\mathcal{S}$ until it reaches the threshold $\boldsymbol{\epsilon}$.

$$G(S) = \sum\nolimits_{j=1}^{l} \min\{\sigma_j^{\mathcal{S}}, \epsilon_j\} \tag{7}$$

The corresponding marginal function is denoted as $v_i(\mathcal{S})$:

$$v_i(\mathcal{S}) = G(\mathcal{S} \cup \{w_i\}) - G(\mathcal{S}), \text{ where } w_i \in \mathcal{W} - \mathcal{S}. \tag{8}$$

Based on Eq. (8), BCDToken uses a functionality $SellerSelection()$ to determine the auction winners, as shown in Fig. 6. When the consumer invokes the functionality $SellerSelection()$, BCDToken will check whether the invocation time and invoker are legal. Then, it starts to select sellers by initializing an empty set $\mathcal{S}$. The process will be conducted iteratively. In each iteration, BCDToken computes the weight $\rho_i = \frac{v_i(\mathcal{S})}{\beta_i}$ for each seller and records the seller $w_{i^*}$ who has the maximum weight as the winner in the current iteration. Meanwhile, BCDToken updates the total cost and the winner set $\mathcal{S}$. The algorithm terminates when $G(\mathcal{S}) = \sum_{j=1}^{l} \epsilon_j$, which means that the winners can meet the quality requirements of all sensing tasks. The computation overhead is $O(n^2 l)$, where $n$ is the number of sellers and $l$ is the number of tasks.

### 4.2.3 Payment Computation

After selecting the sellers, BCDToken uses a functionality *Pricing()* to determine the payment for each winner $w_i \in \mathcal{S}$. To compute the payment $p_i$ for each auction winner $w_i \in \mathcal{S}$, we define four notations. Let $\mathcal{B}_{-i}$ denote all bids except $\mathcal{B}_i$, and $\mathcal{S}'$ be the winner set which is iteratively produced by our greedy selection strategy after we remove $w_i$ from $\mathcal{W}$. Moreover, the corresponding reliability truncation function is $G(\mathcal{S}')$ and the marginal function is $v_i(\mathcal{S}')$. Then, the payment $p_i$ can be computed as follows:

$$p_i = \max\Big\{\frac{\beta_k\, v_i(\mathcal{S}')}{v_k(\mathcal{S}')}|k=1,2,\cdots\Big\}, \tag{9}$$

where $k$ is the number of iteration and $\beta_k$ is the corresponding winning bid value.

The detailed payment computation process is shown in Fig. 6. First, the functionality *Pricing()* checks the legality of the invoker and invocation time in Step 1. Then, it initializes the set $\mathcal{S}'$ and creates a new storage to record the bid profile $Bids(w_i)$ of the current winner $w_i$ in Steps 2-3. Next, the payment $p_i$ is calculated according to Eq. (9) in Step 4. Finally, BCDToken deletes the temporary set $\mathcal{S}'$ to free the space owing to the expensive storage cost on the blockchain. The computation overhead of the whole process is $O(n^3 l)$, where $n$ is the number of sellers and $l$ is the number of tasks.

## 5 The STDR mechanism

In the BCDT system, sellers are untrustworthy so that they might submit forged data for trading to evade the corresponding sensing costs. To stimulate sellers to submit truthful sensed data, we let the BCDToken smart contract record sellers' reliabilities, based on which the truth values of sensed data are derived by applying truth discovery techniques. Moreover, we let the consumer evaluate sellers after each data trading by giving a certain rating scores according to their sensed data. BCDToken will maintain and update sellers' reliabilities using these rating scores. Owing to the intrinsic characteristic of blockchain, the reliability becomes a public, persistent, and tamper-resistant reputation metric of each seller. Sellers will try their best to improve reliability values for obtaining the better reputation and rewards. Such motivation will incentivize them to submit truthful sensed data. Additionally, we also need to protect sellers' sensed data from being revealed to any others (except the data owner and the consumer) during the truth discovery and rating process. Based on these considerations, we propose the STDR mechanism for BCDT to ensure the truthfulness of trading data in this section.

The crucial design of STDR is to make the consumer give truthful rating scores for sellers. However, this is challenging because the consumer is also untrustworthy and it might maliciously evaluate some sellers by giving decreased rating scores. To tackle this problem, we let the BCDToken smart contract act as a trustworthy supervisor to verify the truthfulness of rating scores. More specifically, the consumer first derives the truth of sensed data and calculates the rating scores according to the differences between seller's data and the truth. Likewise, BCDToken computes the encrypted rating scores by applying the homomorphic encryption techniques. Note that BCDToken can only get to know the encrypted sensed data. Thus, it has to conduct the verification on encrypted rating scores, in which it needs to infer whether two encrypted scores are equivalent. Since two same rating scores might produce different homomorphically encrypted data, we cannot verify the equivalence by directly comparing two encrypted rating scores. By using data hiding techniques, we propose a novel light-weight comparison protocol to determine the equivalence of two encrypted data. Based on this idea, STDR can efficiently verify the truthfulness of rating scores while protecting the privacy of sensed data from leakage. Untruthful rating scores will incur a certain monetary punishment to the consumer. To the best of our knowledge, STDR is the first privacy-preserving truth discovery mechanism with truthful rating. The detailed STDR mechanism mainly includes the privacy-preserving truth discovery and truthful reliability rating, described as follows.

### 5.1 Privacy-preserving Truth Discovery

To protect the data privacy, sellers are required to upload the data ciphertext $EnData$ to IDDS rather than the data plaintext $Data$, where $EnData$ is encrypted with the consumer's homomorphic public key $pk$, *i.e.*, $EnData = E_{pk}[Data]$. Then, the consumer can download $EnData$ from IDDS and decrypt them with its private key $sk$ to obtain $Data$. To prompt the sellers to report truthful data, the consumer needs to conduct the truth discovery and reliability rating on $Data$. Hence, the consumer will first execute truth discovery to find the truth of data out of the data plaintext $Data$ (*i.e. a variety of reports from winners*) which is unknown a priori, denoted as $\boldsymbol{\mu}$=$\{\mu_1,\mu_2,\cdots,\mu_l\}$. Note that STDR can be applied to any numerical data, and we use binary data here to illustrate the truth discovery and rating processes for a better understanding. That is, we consider 'true' or 'false' task, such as whether there exists traffic jams, so the observed data of each task $t_j$ from the seller $w_i$ is denoted by a binary value $data_{ij}$=$\{-1,1\}(\in Data)$. $data_{ij}$=1 means that $w_i$ reports traffic jam happens in PoI $j$ (*i.e.*, $t_j$) and $data_{ij}$=$-1$ otherwise. However, a seller $w_i$ may not process $t_j$, so we set $data_{ij}$ as 0 for convenience of calculations.

$$Data = \begin{bmatrix} 1 & -1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \cdots & 1 \end{bmatrix} \tag{10}$$

where $Data_i$=$\{data_{ij}|t_j \in \mathcal{T}_i\}$ indicates the data from $w_i$.

As mentioned before, the larger the reliability of the seller $w_i$, the higher quality of its sensed data. The reliability $r_i$ of the seller $w_i$ will be regarded as the weight of its sensed data to compute the truth. According to the observed data matrix $Data$ and the sellers' reliabilities. We can calculate the weighted average of a task $t_j$'s sensed data as the estimate truth value $\hat{\mu}_j$:

$$\hat{\mu}_j = \frac{\sum_{w_i \in \mathcal{W}_j \cap \mathcal{S}} data_{ij} \cdot r_i}{\sum_{w_i \in \mathcal{W}_j \cap \mathcal{S}} r_i} \tag{11}$$

Then, we derive that if $\hat{\mu}_j \geq 0$, $\mu_j$=1; Otherwise, $\mu_j$=$-1$. Here, since no one knows the truth value, for $\forall \mu_j \in \boldsymbol{\mu}$, we use the data of $t_j$ from multiple sellers to estimate the truth $\mu_j$. The accuracy of the estimated truth value $\hat{\mu}_j$ depends on each participated seller's reliability. Hence, we leverage the reliability as the weight to estimate the truth value. To truthfully evaluate the reliability of sellers, we then in turn use truthful rating to calculate the reliability of sellers.

### 5.2 Truthful Rating

In this subsection, we first calculate the rating scores on evaluating the sellers' reliabilities, and then give the truth-

**Protocol 1** The STDR Mechanism

```
Input: BCDToken: $\mathcal{S}, \mathcal{T}, \mathcal{R}$; Sellers: $Data$; Consumer: $(pk, sk)$
Output: BCDToken: $\mathcal{R}^*$; Consumer: $Q$
1: The consumer creates a pair of public and private keys of homomorphic encryption, i.e., $pk, sk$, and sends $pk$ via $Initiate()$ to BCDToken at the beginning of data trading.
2: After receiving $pk$ from BCDToken, each winner $w_i\in\mathcal{S}$ encrypts its data with $pk$, i.e., $EnData_i = E_{pk}[Data_i]$, and sends $EnData_i$ to IDDS and the data address to BCDToken.
3: After receiving the data address from BCDToken, the consumer downloads each piece $EnData_i \in EnData$ from IDDS and decrypts $EnData_i$ with $sk$, i.e., $Data_i = D_{sk}[EnData_i]$. Then the consumer computes qualities $Q$ of data according to Eq. (12) and sends $Q$ to BCDToken.
4: After receiving $Q$ from consumer, BCDToken computes $E_{pk}[q'_{ij}]\in E_{pk}[Q']$ according to Eq. (18) and obtains $E_{pk}[q_{ij}]\in E_{pk}[Q]$ by encrypting $q_{ij}$. Then BCDToken creates two matrices $\boldsymbol{\lambda}$, $\boldsymbol{\eta}$ and stores the SHA-3 encrypted $\boldsymbol{\eta}$, i.e., sha3($\boldsymbol{\eta}$). BCDToken calculates $E_{pk}[\lambda_{ij}(q_{ij}–q'_{ij})+\eta_{ij}]\in E_{pk}[\boldsymbol{\lambda}(Q–Q')+\boldsymbol{\eta}]$ for each $q_{ij}$, and sends $E_{pk}[\boldsymbol{\lambda}(Q–Q')+\boldsymbol{\eta}]$ to the consumer.
5: After receiving $E_{pk}[\boldsymbol{\lambda}(Q–Q')+\boldsymbol{\eta}]$ from BCDToken, the consumer decrypts them to get $\boldsymbol{\eta}'$, and sends $\boldsymbol{\eta}'$ to BCDToken.
6: After receiving $\boldsymbol{\eta}'$ from consumer, BCDToken encrypts $\boldsymbol{\eta}'$ with SHA-3, and compares sha3($\boldsymbol{\eta}'$) with sha3($\boldsymbol{\eta}$) to verify the truthfulness of the consumer.
7: After the truthfulness verification, BCDToken rates each seller according to Eq. (14) to obtain the scores $O$, and updates the sellers' reliabilities $\mathcal{R}$ according to Eq. (15).
```

fulness verification for rating scores.

### 5.2.1 Reliability Rating

In order to rate these sellers, we first need to measure the qualities $Q=\{q_{ij}|w_i\in\mathcal{S}, t_j\in\mathcal{T}_i\}$ of their sensed data. The quality $q_{ij}$ of task $t_j$'s data collected by the seller $w_i$ ($\in\mathcal{S}$) is the difference between the estimate truth value $\hat{\mu}_j$ and the observed value $data_{ij}$. So we determine the overall quality $Q_i$ of the seller $w_i$ as follows:

$$q_{ij} = data_{ij} - \hat{\mu}_j \tag{12}$$

$$Q_i = \big(\sum\nolimits_{t_j\in\mathcal{T}_i} |q_{ij}|\big)/|\mathcal{T}_i| \tag{13}$$

where the lower value of $Q_i$, the better quality of the data.

BCDToken can compute the average sensing quality of sellers and get the rating scores $O$=$\{o_i|w_i\in\mathcal{S}\}$ as follows:

$$o_i = \big(\sum\nolimits_{w_i\in\mathcal{S}} Q_i\big)/|\mathcal{S}| - Q_i \tag{14}$$

If $Q_i$ is less than the average quality, $o_i<0$, whereas $o_i>0$.

After that, BCDToken updates the reliability of each seller according to the rating scores $O$ as follows:

$$r_i = r_i + \alpha\cdot o_i \tag{15}$$

where $\alpha$ is the impact factor that rating scores influence $r_i$.

Then BCDToken will normalize all reliabilities of data sellers so as to select sellers in the trading phase of next CDT. The normalization process is as follow:

$$r_i^* = (r_i - r_{min})/(r_{max} - r_{min}) \tag{16}$$

where $r_{min}=\min\{r_i|w_i\in\mathcal{W}\}$ and $r_{max}=\max\{r_i|w_i\in\mathcal{W}\}$.

### 5.2.2 Truthfulness Verification

The qualities $Q$ can be merely calculated by the data consumer, because the real sensed data required for calculation only can be obtained by the data consumer. However, the consumer might be reluctant to spend time rating these sellers and fabricate the quality values, since it just needs the truth values. In order to prevent the quality values manipulation from happening, BCDToken needs to check that whether the consumer indeed accomplishes to measure the sensed data and returns the truthful qualities $Q$.

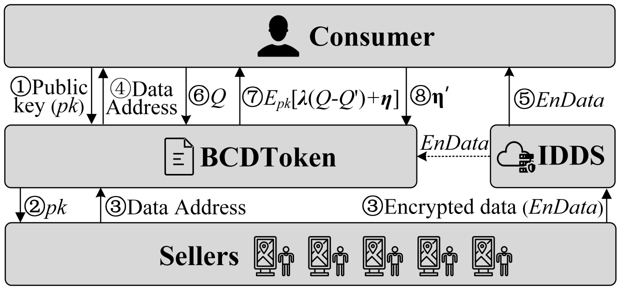


Fig. 7: The workflow of STDR

**First**, BCDToken asks for all qualities $Q$ from consumer and encrypts $Q$ again to get $E_{pk}[Q]=\{E_{pk}[q_{ij}]|q_{ij}\in Q\}$. Then BCDToken computes $E_{pk}[Q']=\{E_{pk}[q'_{ij}]|q'_{ij}\in Q'\}$ directly according to the encrypted data $E_{pk}[Data]=\{E_{pk}[data_{ij}]|data_{ij}\in Data\}$. We can deduce $E_{pk}[q'_{ij}]$ from $E_{pk}[Data]$ as follows:

$$q_{ij}\overset{\text{def}}{=}data_{ij} - \hat{\mu}_j \tag{17}$$

$$\begin{aligned} E_{pk}[q'_{ij}] &= E_{pk}[data_{ij}]\otimes E_{pk}[\hat{\mu}_j]^{-1} \\ &= E_{pk}[data_{ij}]\otimes E_{pk}[\frac{\sum_{w_i\in\mathcal{S}_j} data_{ij}\cdot r_i}{\mathcal{R}_{\mathcal{S}_j}}]^{-1} \\ &= E_{pk}[data_{ij}]\otimes E[data_{1j}]^{\frac{-r_1}{\mathcal{R}_{\mathcal{S}_j}}}\otimes\cdots\otimes E[data_{Kj}]^{\frac{-r_K}{\mathcal{R}_{\mathcal{S}_j}}} \end{aligned} \tag{18}$$

where $\mathcal{S}_j=\mathcal{W}_j\cap\mathcal{S}$, $\mathcal{R}_{\mathcal{S}_j}=\sum_{w_i\in\mathcal{S}_j} r_i$, and $K=|\mathcal{S}_j|$.

**Second**, to verify whether each $q_{ij}\in Q$ is truthful (*i.e.*, $q_{ij}=q'_{ij}$) in case of only having $E_{pk}[Q']$, BCDToken needs to compare $E_{pk}[q_{ij}]$ and $E_{pk}[q'_{ij}]$. But we may get two unequal encrypted results when using the same homomorphic encryption for two equal numbers, *e.g.*, if $e=1$ and $e'=1$, $E_{pk}[e]\neq E_{pk}[e']$. BCDToken cannot determine that whether $q_{ij}=q'_{ij}$ by comparing $E_{pk}[q_{ij}]$ and $E_{pk}[q'_{ij}]$ for equality directly. So we design a light-weight truthfulness verification trick of qualities as below.

**Third**, BCDToken randomly creates two matrices $\boldsymbol{\lambda}=\{\lambda_{ij}|data_{ij}\in Data\}$ and $\boldsymbol{\eta}=\{\eta_{ij}|data_{ij}\in Data\}$, and computes each $E_{pk}[\lambda_{ij}(q_{ij}-q'_{ij})+\eta_{ij}]\in E_{pk}[\boldsymbol{\lambda}(Q-Q')+\boldsymbol{\eta}]$ as follows:

$$\begin{aligned} & E_{pk}[\lambda_{ij}(q_{ij}-q'_{ij})+\eta_{ij}] \\ =\ & E_{pk}[q_{ij}-q'_{ij}]^{\lambda_{ij}}\otimes E_{pk}[\eta_{ij}] \\ =\ & E_{pk}[q_{ij}]^{\lambda_{ij}}\otimes E_{pk}[q'_{ij}]^{-\lambda_{ij}}\otimes E_{pk}[\eta_{ij}] \end{aligned} \tag{19}$$

**Fourth**, BCDToken sends $E_{pk}[\boldsymbol{\lambda}(Q-Q')+\boldsymbol{\eta}]$ to the consumer. The consumer is required to decrypt $E_{pk}[\boldsymbol{\lambda}(Q–Q')+\boldsymbol{\eta}]$ and send the decrypted values $D_{sk}[E_{pk}[\boldsymbol{\lambda}(Q-Q')+\boldsymbol{\eta}]]=\boldsymbol{\eta}'=\{\eta'_{ij}|data_{ij}\in Data\}$ to BCDToken.

For $\forall i,j$, if $\eta'_{ij}$=$\eta_{ij}$, we say $\boldsymbol{\eta}'$=$\boldsymbol{\eta}$. So the rating is proved to be truthful in Theorem 4. It should be noticed that data on the blockchain are public visible, so we record the SHA-3 value of $\eta_{ij}$, *i.e.*, sha3($\eta_{ij}$), in BCDToken rather than the original $\eta_{ij}$. Then we encrypt $\eta'_{ij}$ with SHA-3 algorithm, and compare sha3($\eta_{ij}$) with sha3($\eta'_{ij}$) instead.

### 5.3 The Detailed STDR mechanism

The interactions among the consumer, the sellers, BCDToken, and IDDS in STDR are presented in Protocol 1 and are also illustrated in Fig. 7. We design STDR based on the privacy-preserving truth discovery and truthful rating, which correspond to Steps 2-3 and Steps 4-6 in Protocol 1, respectively. In STDR, we introduce IDDS to provide data storage and use BCDToken to provide quality verification service. Steps 4 and 5-6 are completed via two subfunctions $CreateVer()$ and $Verify()$ in $Rating()$ of BCDToken.

## 6 The Security Analysis of BCDT

In this section, we will analyze the security of the BCDT system in the following three parts.

### 6.1 Analysis of the BRA mechanism

In this subsection, we prove that the BRA mechanism can achieve truthfulness and individual rationality.

**Lemma 1 (Bid monotonicity).** *Each seller $w_i$ who wins by bidding $(\mathcal{T}_i, \beta_i)$ will still win by biding any $\beta'_i < \beta_i$ and any $\mathcal{T}'_i \supset \mathcal{T}_i$ when other bids are fixed.*

*Proof.* Let $v_i(\mathcal{S})$ denote the marginal reliability of seller $w_i$ who bids $(\mathcal{T}_i, \beta_i)$, and $\rho_i{=}\frac{v_i(\mathcal{S})}{\beta_i}$. Consider the first case that the seller $w_i$ bids for sensing tasks $\mathcal{T}_i$ with a lower price $\beta'_i$, *i.e.*, $(\mathcal{T}_i, \beta'_i)$. We have $\rho_i = \frac{v_i(\mathcal{S})}{\beta_i} \leq \frac{v_i(\mathcal{S})}{\beta'_i} = \rho'_i$. Next, we consider the second case, where $w_i$ bids for more sensing tasks, *i.e.*, $(\mathcal{T}'_i, \beta_i)$. We can get $\rho_i{=}\frac{v_i(\mathcal{S})}{\beta_i} \leq \frac{v_i(\mathcal{S})'}{\beta_i}{=}\rho'_i$. That is to say, $\rho_i \leq \rho'_i$ holds in the two cases. According to the greedy seller selection strategy, $\beta'_i$ will always be selected before $\beta_i$. Thus, $w_i$ can still win the auction by biding any $\beta'_i{<}\beta_i$ and any $\mathcal{T}'_i{\supset}\mathcal{T}_i$. Above all, the lemma holds. □

**Lemma 2 (Critical payment).** *Each seller $w_i$ is paid a critical value $p_i$.*

*Proof.* Let $\mathcal{S}$ and $\mathcal{S}'$ denote the winner sets produced by the seller selection algorithm and the pricing algorithm in Fig. 6, respectively. Moreover, $\mathcal{S}_k$ is the winner set until the $k$-th iteration. If $w_i$ is the winner selected in the $k$-th iteration, $\mathcal{S}_k{=}\mathcal{S}_{k-1} \cup \{w_i\}$. We assume $w_i$ reports a bid $\beta'_i$ instead of $\beta_i$. We need to prove that $w_i$ will lose the auction if $\beta'_i{>}p_i$, otherwise it will win when $\beta'_i{\leq}p_i$. Then, we consider these two cases in the $k$-th iteration:

**Case 1:** $\beta'_i > p_i$**.** According to Fig. 6, we can derive that $\frac{v_i(\mathcal{S}_{k-1})}{\beta'_i}{=}\frac{v_i(\mathcal{S}'_{k-1})}{\beta'_i}{<}\frac{v_i(\mathcal{S}'_{k-1})}{p_i}{\leq}\frac{v_k(\mathcal{S}'_{k-1})}{\beta_k}$, where $w_k$ is a winner, so that $\mathcal{S}'_k{=}\mathcal{S}'_{k-1}{\cup}\{w_k\}$. The first equation holds because $\mathcal{S}_{k-1}{=}\mathcal{S}'_{k-1}$, and the last inequation makes sense for $p_i{\geq}\frac{\beta_k v_i(\mathcal{S}'_{k-1})}{v_k(\mathcal{S}'_{k-1})}$ according to Eq. (9). Hence, $w_k$ is selected as a winner instead of $w_i$ in the $k$-th iteration. So, $\mathcal{S}_k{=}\mathcal{S}_{k-1}{\cup}\{w_k\}{=}\mathcal{S}'_k$. Based on the above analysis, we can conclude that $w_i$ will fail in all iterations of the seller selection.

**Case 2:** $\beta'_i{\leq}p_i$**.** Assume that the seller selection runs over $\mathcal{B}_{-i}$, which is the process for pricing $w_i$. According to Eq. (9), we assume that $p_i{=}\frac{\beta_k v_i(\mathcal{S}_{k'-1})}{v_k(\mathcal{S}_{k'-1})}$, where $w_k$ is the winner in the $k'$-th iteration. Now, we run the seller selection again with the input set $\mathcal{B}$. We discuss two subcases:

1) $w_i$ wins before the $k'$-th iteration;

2) $w_i$ does not win before the $k'$-th iteration. In the $k'$-th iteration: $\frac{v_i(\mathcal{S}_{k'-1})}{\beta'_i}{\geq}\frac{v_i(\mathcal{S}_{k'-1})}{p_i}{\geq}\frac{v_k(\mathcal{S}_{k'-1})}{\beta_k}$. Therefore, $w_i$ wins in this iteration.

Synthesizing both subcases, $w_i$ wins when $\beta'_i \leq p_i$.

In summary, all payments for winners are critical. □

**Theorem 1.** *The BRA mechanism is truthful. Both of the auctioneer and sellers will follow the whole reverse auction rules to complete the data trading, during which sellers will report their costs honestly.*

*Proof.* In the BRA mechanism, the BCDToken smart contract works as the auctioneer. The smart contract will be automatically executed on the blockchain to enforce all participants to follow the auction rules. Moreover, according to the two-step bidding strategy, each seller will submit the encrypted bid in the first step and the unencrypted bid in the second step. No sellers can eavesdrop others' bid values. Also, the sellers cannot lie about their bids since they can be verified by the encrypted bid submitted in the first step. Thus, the whole auction is trustworthy. Additionally, Lemmas 1 and 2 prove that the seller selection is monotonic and all payments are critical, respectively. According to [30], sellers will report their costs honestly. Therefore, the whole BRA mechanism is truthful. The theorem holds. □

**Theorem 2.** *The BRA mechanism is individually rational.*

*Proof.* We consider that a seller $w_i$ probably encounters these two situations, $w_i{\in}\mathcal{S}$ and $w_i{\notin}\mathcal{S}$. If $w_i{\notin}\mathcal{S}$, its payment will be zero. Otherwise, it wins the auction and its payment is $p_i$. According to Lemma 2, $w_i$ will always be paid with the critical value $p_i$ when it bids any $\beta_i \leq p_i$. Each seller bids its truthful cost due to the truthfulness in Theorem 1. Apparently, $p_i{-}c_i{\geq}0$ holds. □

**Theorem 3.** *According to our previous work [31], we can derive that the seller selection algorithm achieves the $(1{+}\ln\frac{\theta\sum_{j=1}^{l}\epsilon_j}{opt})$-approximation, where opt is the cost of the optimal solution to the minimum weight set cover problem. $\theta{=}\max\{\frac{1}{\theta_1}, \theta_2\}$, where $\theta_1{=}\min\{\frac{v_i(\mathcal{S}_{k-1})}{\beta_i}|i{=}1, \cdots, k\}$, $\mathcal{S}_k$ is the winner set after the k-th iteration of the seller selection, and $\theta_2{=}\frac{C(\mathcal{S})}{\sum_{j=1}^{l}\epsilon_j}$.*

### 6.2 Analysis of the STDR Mechanism

In this subsection, we prove that the STDR mechanism can ensure the truthfulness and privacy-preserving.

**Theorem 4.** *The rating scores are truthful in STDR.*

*Proof.* For $\forall q_{ij} \in Q$, if $q_{ij}{=}q'_{ij}$, then $\lambda_{ij}(q_{ij}-q'_{ij})+\eta_{ij}{=}\eta_{ij}$. The consumer can decrypt $E_{pk}[\lambda_{ij}(q_{ij}-q'_{ij})+\eta_{ij}]$ and get the decrypted value $\eta'_{ij}=\lambda_{ij}(q_{ij}-q'_{ij})+\eta_{ij}$. Even though the consumer knows the values of $q_{ij}$ and $q'_{ij}$ and the construction of the internal expression, it cannot deduce $\eta_{ij}$ from $\lambda_{ij}(q_{ij}-q'_{ij})+\eta_{ij}$ because of the randomly created $\lambda_{ij}$. If the consumer gives a untruthful $q_{ij} \neq q'_{ij}$, it yields $\eta'_{ij} = \lambda_{ij}(q_{ij}-q'_{ij})+\eta_{ij} \neq \eta_{ij}$, which demonstrates the untruthfulness of consumer. Hence, this check can force the

```
Modifiers:
    1) modifier onlyBefore(uint time)  require(now < time);
    2) modifier onlyAfter (uint time)  require(now > time);
    3) modifier onlyTrue (uint flag)  require(flag == true);
    4) modifier onlyFalse (uint flag)  require(flag == false);
```

Fig. 8: Some Modifiers

consumer to provide truthful $q_{ij}$ which equals $q'_{ij}$. So, we say the rating scores are truthful in STDR. □

**Theorem 5.** *STDR can protect the data privacy from being revealed to any other data sellers and BCDToken* (i.e., *blockchain) except the data consumer.*

*Proof.* According to Def. 4, we construct the three simulators $BT$, $DC$, and $DS$ for BCDToken, the data consumer, and an arbitrary seller $w_i$, so that their views can be efficiently simulated by the outputs of the simulators $BT$, $DC$, and $DS$. That is to say, the outputs of the simulators and the views are computational indistinguishable.

Denote the views of BCDToken, the consumer, and seller $w_i$, as $VIEW_B$, $VIEW_C$, and $VIEW_{w_i}$. Then, according to STDR, these views can be represented as follows:

$$VIEW_B = (Q, E_{pk}[Q'], \boldsymbol{\lambda}, \boldsymbol{\eta}, E_{pk}[\boldsymbol{\lambda}(Q - Q') + \boldsymbol{\eta}]) \tag{20}$$

$$VIEW_C = (sk, E_{pk}[Data], Q, E_{pk}[\boldsymbol{\lambda}(Q - Q') + \boldsymbol{\eta}]) \tag{21}$$

$$VIEW_{w_i} = (Data_i, E_{pk}[Data_i]) \tag{22}$$

where the encrypted data $EnData$ stored on IDDS are known for all users because of the data address published in BCDToken, which cannot be used to derive $Data$ without $sk$. $sk$ is the input of data consumer. $\boldsymbol{\lambda}, \boldsymbol{\eta}$ are the internal random matrices of BCDToken. $Data_i$ is the sensed data of seller $w_i$, and the others are the messages received by the three parties during the execution of STDR. Here, we ignore the public message such as $\mathcal{T}$, $\mathcal{S}$, $pk$ for simplicity.

Simulator $BT$ randomly selects numbers to construct three matrices $Data', \boldsymbol{\lambda}', \boldsymbol{\eta}'$ for each element in $Data$, and outputs $Q'$ based on $Data'$. By using the public homomorphic encryption key $pk$, $BT$ creates $E_{pk}[Data']$ to compute $E_{pk}(Q'')$, and outputs $E_{pk}[\boldsymbol{\lambda}'(Q' - Q'') + \boldsymbol{\eta}']$. Since both of $\boldsymbol{\lambda}, \boldsymbol{\eta}$ and $\boldsymbol{\lambda}', \boldsymbol{\eta}'$ are selected at random, and $E_{pk}[Data], E_{pk}[Data']$ are the ciphertexts of the homomorphic encryption $E_{pk}[\cdot]$, the outputs of simulator $BT$ and $VIEW_B$ are computational indistinguishable. Likewise, simulator $DC$ randomly selects the numbers to construct a matrice $Data'$ for $Data$ and a matrice $E_{pk}[\boldsymbol{\lambda}(Q - Q') + \boldsymbol{\eta}]'$ for $E_{pk}[\boldsymbol{\lambda}(Q - Q') + \boldsymbol{\eta}]$, and outputs $(sk, E_{pk}[Data'], E_{pk}[\boldsymbol{\lambda}(Q - Q') + \boldsymbol{\eta}]')$ by using the input $sk$ and homomorphic encryption operations. As a result, the outputs of simulator $DC$ and $VIEW_C$ are also computational indistinguishable. In addition, simulator $DS$ randomly selects numbers of matrice $Data'_i$ for $Data_i$, and directly outputs $(Data'_i, E_{pk}[Data'_i])$. Since $Data_i$ and $Data'_i$ are randomly selected, the outputs of simulator $DS$ and $VIEW_{w_i}$ are computational indistinguishable. □

### 6.3 The Security Analysis

We use some modifiers in Fig. 8 which are introduced at the beginning of Sec. 3 to ensure that BCDToken can run steadily. The keyword *require* can roll back all states without deducting gas when encountering some invalid codes. The security of BCDT is guaranteed by the following points:

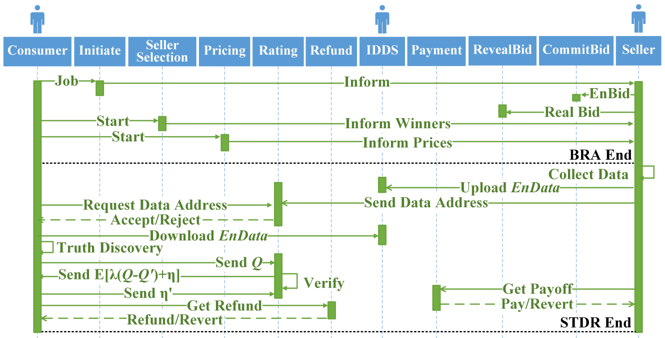


Fig. 9: The Sequence Diagram of BCDT System

1) **Only one job in one round.** Once a consumer invokes BCDToken to launch a job like in Fig. 3, others cannot invoke it until the job ends. If there exists an active job, the flag $jobEnded$ whose default value is $true$ will become to be $false$, which cannot pass the check of $onlyTrue(jobEnded)$ in *Initiate()*, so that other consumers will be rejected.
2) **Each participant should offer deposit.** We set *Initiate()* and *CommitBid()* $payable$, a keyword of smart contract. Invoking the two functions requires the trading initiator (*i.e.*, data consumer) and data sellers to transfer ethers to BCDToken. If an untruthful operation from any participant is detected, its deposit will be fined as a compensation for other truthful participants. The untrusted participation might occur in both of the data consumers and sellers. For example, the untrusted consumer might transfer insufficient rewards, manipulate rating scores, quit the system midway; an untrusted seller might also drop out the data trading halfway, eavesdrop others' bids, misreport its bid, etc.
3) **Each procedure only be executed orderly.** Every function in our BCDToken has an independent entry through separate calls. We also set some time modifiers to ensure the safety. We illustrate the sequence diagram of BCDT in Fig. 9. For example, *RevealBid()* should be invoked after $t_{bid}$ and before $t_{reveal}$; the reverse auction can only be executed after *RevealBid()*. Sufficient rewards, no less than the total payments computed in $Pricing()$, are required to be transferred to BCDToken within a given time duration after $Pricing()$. In addition, we design a reward transfer function for BCDToken, in which some modifiers are used to verify the time of invocation, the identity of invoker, the quantity of rewards, etc. Consequently, only when the consumer who initiates the data trading transfers enough rewards to BCDToken within the given time limitation, the selected sellers will be informed to execute the tasks.

The untrusted participation either will be prevented or will be checked by some special modifiers in BCDToken, so that all participants will follow the workflow of the whole data trading. In addition, we also prove that the whole BCDT system is secure.

**Theorem 6.** *The BCDT system is secure when it employs BCDToken as the data trading broker and meets individual rationality, truthfulness and privacy-preserving.*

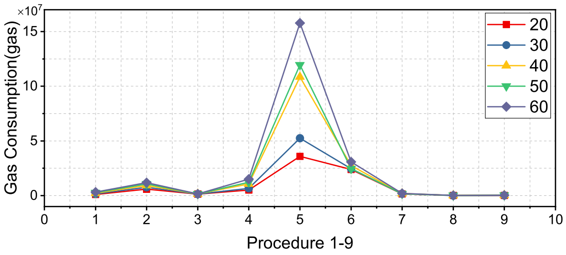


Fig. 10: Gas consumption of each procedure

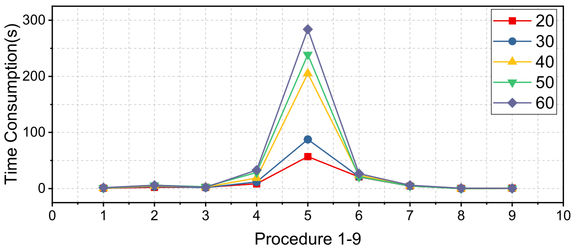


Fig. 11: Time consumption of each procedure

*Proof.* By employing the BCDToken smart contract as the data trading broker, BCDT can guarantee the trustworthiness of data trading workflow according to the above security analysis of BCDToken. Besides, the other three properties also can be ensured: **1) Individual rationality.** According to Theorem 2, each seller's payoff is non-negative when it follows the whole reverse auction rules to complete the data trading. **2) Truthfulness.** According to Eq. (14) of STDR, each seller can get a non-negative score when it reports the truthful data. Moreover, the truthfulness of data collection costs and rating scores on sellers' reliabilities also can be ensured by Theorems 1 and 4, respectively. **3) Privacy-preserving.** The privacy of the sensed data and the derived truth is preserved during the data trading according to Theorem 5. Before this, all sensed data are encrypted and stored on IDDS, which can only be decrypted and accessed by the consumer who pays for the data and owns the private key. Moreover, due to the IPFS technique adopted in IDDS, these sensed data also cannot be tampered by others, even though their addresses are publicly available. Hence, the BCDT system is secure. □

## 7 Implementation and Evaluations

In this section, we evaluate the performance of BCDT through extensive simulations on blockchain. Here, we only need to verify the feasibility of BCDT since the security property of BCDT has been proven through the theoretical analysis in Sec. 6. More specifically, we mainly evaluate the gas consumption and running time of BCDT by changing different parameters. This is because BCDT is the first blockchain-based CDT system which can provide the privacy-preserving, truthful data, truthful cost, and truthful rating simultaneously, so that there are no existing systems that can be used to make a fair performance comparison.

### 7.1 Implementation and Settings

We implement a prototype of BCDT including the BCDToken smart contract, the consumers, and sellers. BCDToken is the core component in the BCDT system, which acts as the data broker and coordinates the consumer and sellers to complete the data trading. Moreover, BCDToken, embedded with the BRA and STDR mechanisms, is realized in the programming language Solidity, and it is deployed to a local simulated network TestRPC using Ethereum development tool Ganache Cli. The simulated network is much like the real Ethereum environment, irrespective of the time-consuming mining process and the complex network circumstances in Ethereum. The consumer's client and seller's client are written in Java, in which the consumer can complete truth discovery and rating, while each seller can finish auction and data encryption. Here, we leverage JavaScript (JS) as the intermediate interactive language for the consumer and sellers to communicate with BCDToken.

Since the frequently used SHA-3 algorithm is different in BCDToken (*i.e.*, Solidity) and clients (*i.e.*, JS), we implement a custom SHA-3 algorithm in JS to make $en\beta_i$ have the identical value with sha3($w_i, \beta_i, nonce_i$) in Solidity. Before the evaluation, we set some major parameters of BCDT. The number of tasks $l$ varies in $[20, 30, 40, 50, 60]$ while the number of sellers $n$ is 20. The reliability of each seller is randomly generated from 0.6 to 1, and the bid ranges from 10 to 20. The reliability requirements range from 1 to 2.

### 7.2 Evaluation of BCDToken on Simulated Network

The BCDToken smart contract coordinates the consumer and sellers to realize the data initialization, trading, rating, and harvest phases based on the nine main functions: **Initiate**, **CommitBid**, **RevealBid**, **SellerSelection**, **Pricing**, **CreatVer**, **Verify**, **Refund**, and **Payment**. To evaluate the unique performance of BCDToken on the simulated network, we use two special metrics: *gas consumption* and *time consumption*. Note that each function might be invoked multiple times by multiple participants. We use **Procedures 1-9** in Fig. 10 and Fig. 11 to indicate the total gas consumption and time consumption after all invocations of each function, respectively. For example, in Fig. 10, the total gas consumed by Procedure 2 is the accumulated gas consumed by $n$ sellers to invoke the function of **CommitBid**.

Since each computational step will be charged some gas, the more complicated the procedure is, the more gas and time it will consume. The operations to create and write storage data are relatively expensive [32]. As we can see, Procedures 2, 4-6 use more gas and time. Procedure 4 needs to execute a nontrivial set of add, subtract, multiply, divide, compare, and write operations, and there is a positive correlation between the number of winners and gas consumption. Procedure 5 is roughly equivalent to execute Procedure 4 $|\mathcal{S}|$ times. On the other hand, we may traverse more iterations than the entire Procedure 4 to find the critical payment for a winner, which uses more gas and time accordingly. Procedures 2 and 6 use much gas because each SHA-3 encrypted value is a 32-bytes hash value which will take up more storage, and Procedure 6 also involves much operations over homomorphic encrypted data. Procedure 8 including the reliability update process in STDR uses less gas since the update operation involves most read operations with smaller data length, as well as Procedures 1, 3, and 9. Notice that here we use gas in **wei**.

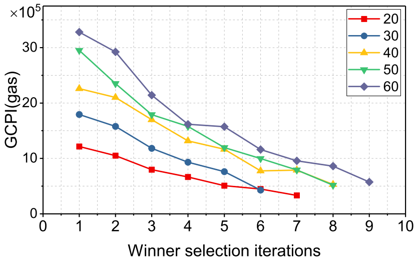


Fig. 12: GCPI vs. iterations

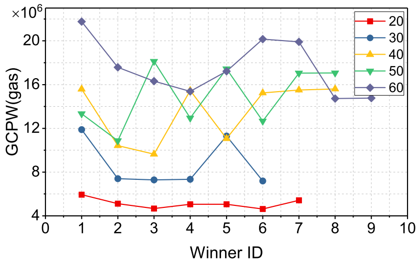


Fig. 13: GCPW vs. ID

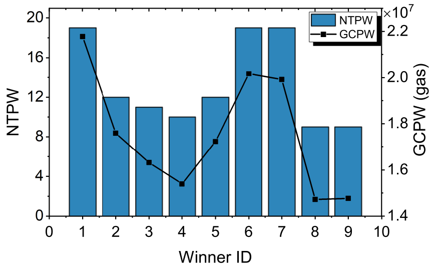


Fig. 14: NTPW vs. ID

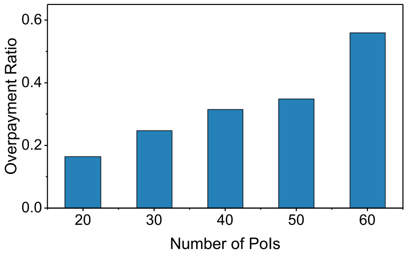


Fig. 15: Overpayment ratio

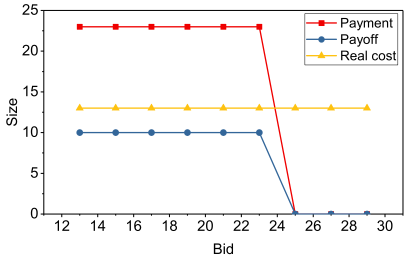


Fig. 16: Payoff of a bid

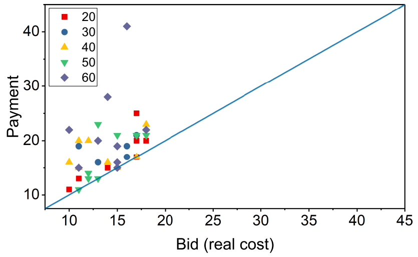


Fig. 17: Payments vs. bids

### 7.3 Evaluations of BRA

Since the core **auction** mechanism of BCDToken includes **SellerSelection** and **Pricing**, we explicitly evaluate the performance of them respectively. Note that there is a *gasLimit* in Ethereum, which restricts the total computational resources that can be used every time to invoke a function. To avoid exceeding gasLimit, we divide the seller selection and pricing procedures into multiple iterations and repeatedly send a transaction to BCDToken to select a winner and price the winner.

**SellerSelection.** We give an example in Fig. 12 to compare the *gas consumption per iteration* (GCPI) under different number of tasks from 20 to 60. We notice a gradual decline of GCPI, because of the constant cost of loading past mined blocks from storage into memory before each selection [33].

**Pricing.** We use *gas consumption per winner* (GCPW) and *number of transactions per winner* (NTPW) as two metrics in Figs. 13 and 14 respectively. We figure out that determining the payment for each winner will consume how much gas and need how many transactions. Fig. 13 shows that the total gas consumption increases as the increasing number of tasks from an overall perspective. The GCPW has nothing to do with the iteration sequence which is only related to the number of tasks and traverse times to obtain its critical payment as shown in Fig. 14. The NTPW represents traverse times needed to price a winner and the corresponding GCPW shows that more gas will be used if more transactions are needed when the number of tasks is 60.

Beyond evaluating the performance on the blockchain, we use the three metrics: overpayment ratio, truthfulness, and individual rationality, to illustrate the properties of our auction mechanism. The overpayment ratio is defined as:

$$\lambda = (\mathcal{P} - \mathcal{C}(\mathcal{S}))/\mathcal{C}(\mathcal{S}) \tag{23}$$

where $\mathcal{P}$ is the total payment and $\mathcal{C}(\mathcal{S})$ is the total cost. It measures the cost paid by the consumer to induce the truthfulness overall. Ensuring truthfulness means that no sellers can improve its payment by committing a different bid from the real one. Individual rationality ensures that each payoff is non-negative.

**Overpayment ratio:** Fig. 15 plots the overpayment ratio $\lambda$ when $l$ changes from 20 to 60. The results show that $\lambda$ is always less than 0.6, which means that the consumer does not have to pay much extra money to induce truthfulness. $\lambda$ increases monotonously with the increasing number of tasks because more sellers will be selected and the increments of the payments are greater than those of the costs.

**Truthfulness:** We randomly pick a winner and change its claimed bid, then recalculate the payment as well as the payoff. The results illustrated in Fig. 16 show that when the truthful bid (real cost) is 13, the critical payment is 23, and the payoff is 10. The payoff remains unchanged when the bid is no more than 23. However, if the bid is larger than 23, the payoff becomes zero which means that the winner loses the auction. Hence, BRA can ensure truthfulness of cost.

**Individual rationality:** In Fig. 17, each winner's payment is greater than its bid when the number of tasks varies from 20 to 60, which demonstrates the individual rationality.

### 7.4 Evaluations of STDR

In order to verify the effectiveness of STDR, we evaluate the protocol from the following two aspects:

**Efficiency:** STDR mainly includes data encryption and decryption. Meanwhile, the rating process, which needs to calculate $E_{pk}(Q)$, also involves some computation on the encrypted data. Hence, we measure the overall running time of data encryption, data decryption, and $E_{pk}(Q)$ calculation when the number of tasks $l$ and the number of sellers $n$ change from 20 to 60. The running time of the three operations is exactly proportional to the amount of data, *i.e.*, becoming longer with the increase of $n$ and $l$. As shown in Fig. 18, the top layer is the running time of data decryption, and the middle layer is the time of $E_{pk}(Q)$ calculation, while the bottom layer is the time of data encryption. Moreover, the running time of STDR is also relative to the security level, which depends on the key size of homomorphic cryptosystem. As depicted in Fig. 19, the running time climbs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3107187, IEEE Transactions on Mobile Computing

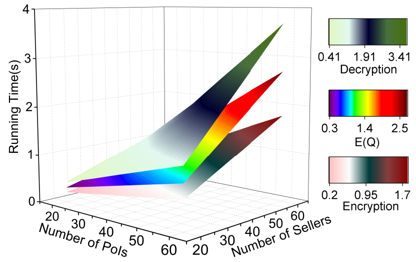


Fig. 18: Running time vs. $l$ and $n$

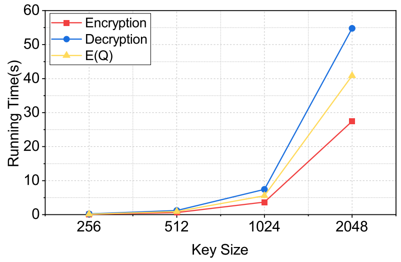


Fig. 19: Running time vs. Key Size

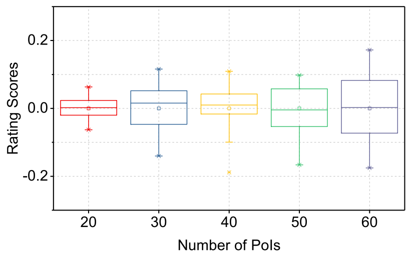


Fig. 20: Rating scores vs. $l$

TABLE 3: Comparisons between BCDT and the existing works

| Systems | Blockchain as Broker | Incentive Mechanism | Truth Discovery | Trusted Participation | Truthful Cost | Truthful Data | Truthful Rating | Cost Privacy | Data Privacy | Truth Privacy |
|---|---|---|---|---|---|---|---|---|---|---|
| [4], [17], [20], [31] | | ✓ | | | ✓ | | | ✓ | | |
| [5], [10], [23] | | ✓ | | | | | | | | |
| [8] | | ✓ | | | | | | | ✓ | |
| [9], [25]–[27], [31] | | ✓ | | | ✓ | | | | | |
| [12] | ✓ | | | ✓ | | | | ✓ | | |
| [14], [40] | ✓ | | ✓ | ✓ | | | | | ✓ | ✓ |
| [15] | ✓ | ✓ | | | ✓ | | | | | |
| [16] | | ✓ | ✓ | | ✓ | | | | ✓ | |
| [18] | | ✓ | | ✓ | ✓ | | | | ✓ | |
| [19], [34], [35] | | | ✓ | | | | | | ✓ | |
| [33], [36] | ✓ | | | ✓ | | | | | ✓ | ✓ |
| [37] | | ✓ | ✓ | | | | | | | |
| [38], [39] | | | ✓ | | | | | | ✓ | ✓ |
| [41] | ✓ | ✓ | | | | | | | | |
| Our system | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

up slowly first, and then increases dramatically with the increase of security level (*i.e.*, key size) when $n$=20, $l$=60. By default, we set the key size as 512 in the simulations.

**Rating scores:** Fig. 20 plots the scores when $l$ changes from 20 to 60, which implies the sensing quality level of each seller compared to the average sensing quality. The scores fluctuations do not exceed the range of 0.2.

## 8 Related Works

Crowdsensed Data Trading (CDT) is a novel paradigm derived from traditional data trading, which takes advantage of the crowds to collect data and tackles the scarcity of data sources for sale. For example, CDT markets in [9] and [37] allow data broker to purchase raw data from sellers and sell consumers the data statistics. In [38], the broker trades private statistics with consumers, which are aggregated over the collected IoT data. However, the raw data in [9], [37], [38] are always same or collected from the fixed locations, which cannot tackle the scarcity of data sources. For this, [42] recruits data sellers to collect data from different locations to enrich the data sources. All of these researches need a trusted third-party data broker and aim to maximize the revenue of broker, while our BCDT system intends to use blockchain as a trusted broker and remove the commission of broker. Moreover, in addition to the untruthful cost in trading, our system also considers some other security issues in data trading compared with these papers, such as untruthful data, untruthful rating, and data privacy.

On the other hand, the newly emerging technology, blockchain, is considered to be used in some data trading systems to eliminate the expense of broker and realize security in some extent. For instance, [36] and [40] allow the consumer to purchase a statistical result calculated by some blockchain nodes over the raw data, which protect data privacy by SGX and additive secret sharing, respectively. But they are not applicable for our scenario, where the truth discovery and the truthfulness verification of rating need to directly execute computation over the encrypted data. [34], [35], [39] adopt the homomorphic encryption which allows direct computation over encrypted data to design the privacy-preserving truth discovery algorithms. However, [39] does not consider the rating for data, and [34], [35] do not involve the privacy of truth and the truthfulness of rating scores. Even though [34], [35], [39] cannot be directly applied to our scenario, they enlighten us to leverage homomorphic cryptography to implement truthful rating with privacy-preserving of data and truth.

Besides, [36] and [40] are traditional blockchain-based data trading systems without incentive mechanisms, which cannot stimulate the participation of sellers and consumers. [34], [35], [39] focus on truth discovery algorithms in mobile crowdsensing system without data trading. The CDT system in [14] also protects data privacy based on additive secret sharing, but does not consider any incentive mechanisms. [41] adopts the incentive Stackelberg game to determine the unit data price and data volume for sale without considering the security issues in blockchain-based data trading. Moreover, the raw data provided by sellers in [14], [36], [40], [41] are existing and fixed, which also cannot relieve the scarcity of data sources. To tackle this problem, [15] designs a reverse auction based CDT system on the blockchain to incentivize sellers to collect various data and report truthful costs, which protects identities' privacy but does not take account of data privacy. Furthermore, all of the above blockchain-based data trading systems do not consider the untruthful data and rating.

We summarize the differences between BCDT and the related works in Table 3. Compared with the related works, BCDT can ensure the trustworthiness of data trading workflow, truthfulness of cost, data, and rating, as well as preserve the privacy of data and truth simultaneously.

## 9 Conclusion

In this paper, the proposed BCDT system is different from existing CDT. We use a meticulous designed smart contract with some blockchain tricks to replace the third-party data broker, which can ensure the security of data trading in system level. We adopt reverse auction in BRA, combined with a two-step bidding strategy, to incentivize sellers to claim truthful bids and prevent sellers to manipulate the auction. Meanwhile, the STDR mechanism based on homomorphic cryptography is designed to realize truth discovery on sensed data and rating of sellers with data privacy-preserving, where the truthfulness verification of rating scores is guaranteed by data hiding techniques. Finally, we theoretically prove the system-level security of the BCDT system and implement a prototype on an Ethereum test network and the evaluations demonstrate its practicability.

## References

[1] B. An, M. Xiao, A. Liu, G. Gao, and H. Zhao, "Truthful crowd-sensed data trading based on reverse auction and blockchain," in *Springer DASFAA*, 2019.

[2] T. Jung, X. Li, W. Huang, J. Qian, L. Chen, J. Han, J. Hou, and C. Su, "Accounttrade: Accountable protocols for big data trading against dishonest consumers," in *IEEE INFOCOM*, 2017.

[3] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *The VLDB Journal*, vol. 29, no. 1, pp. 217–250, 2020.

[4] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE. Trans. Mob. Comput.*, vol. 17, no. 1, pp. 16–28, 2018.

[5] S. He, D. Shin, J. Zhang, J. Chen, and P. Lin, "An exchange market approach to mobile crowdsensing: Pricing, task allocation, and walrasian equilibrium," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 4, pp. 921–934, 2017.

[6] "Thingful," *https://www.thingful.net/*.

[7] "Thingspeak," *https://thingspeak.com/*.

[8] C. Niu, Z. Zheng, S. Tang, X. Gao, and F. Wu, "Making big money from small sensors: Trading time-series data under pufferfish privacy," in *IEEE INFOCOM*, 2019.

[9] Z. Zheng, Y. Peng, F. Wu, S. Tang, and G. Chen, "Trading data in the crowd: Profit-driven data acquisition for mobile crowdsensing," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, pp. 486–501, 2017.

[10] J. Yu, M. H. Cheung, J. Huang, and H. V. Poor, "Mobile data trading: Behavioral economics analysis and algorithm design," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 4, pp. 994–1005, 2017.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Online at: https://bitcoin.org/bitcoin.pdf*.

[12] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *IEEE S&P*, 2016.

[13] V. Buterin, "A next-generation smart contract and decentralized application platform," *White paper*, 2014.

[14] C. Cai, Y. Zheng, and C. Wang, "Leveraging crowdsensed data streams to discover and sell knowledge: A secure and efficient realization," in *IEEE ICDCS*, 2018.

[15] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain," in *IEEE MASS*, 2018.

[16] L. Yang, M. Zhang, S. He, M. Li, and J. Zhang, "Crowd-empowered privacy-preserving data aggregation for mobile crowdsensing," in *ACM Mobihoc*, 2018.

[17] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *IEEE INFOCOM*, 2019.

[18] L. Zhang, Y. Li, X. Xiao, X. Li, J. Wang, A. Zhou, and Q. Li, "Crowdbuy: Privacy-friendly image dataset purchasing via crowdsourcing," in *IEEE INFOCOM*, 2018.

[19] X. Tang, C. Wang, X. Yuan, and Q. Wang, "Non-interactive privacy-preserving truth discovery in crowd sensing applications," in *IEEE INFOCOM*, 2018.

[20] M. Xiao, K. Ma, A. Liu, H. Zhao, Z. Li, K. Zheng, and X. Zhou, "Sra: Secure reverse auction for task assignment in spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 782–796, 2020.

[21] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*, 1999.

[22] O. Goldreich, "Foundations of cryptography: volume 2, basic applications," *Cambridge Uni. Press*, 2009.

[23] Y. Zhang and M. v. d. Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *IEEE INFOCOM*, 2012.

[24] D. Lehmann, L. O'Callaghan, and Y. Shoham, "Truth revelation in approximately efficient combinatorial auctions," *J. ACM*, vol. 49, no. 5, pp. 577–602, 2002.

[25] J. Lin, M. Li, D. Yang, G. Xue, and J. Tang, "Sybil-proof incentive mechanisms for crowdsensing," in *IEEE INFOCOM*, 2017.

[26] H. Zhang, B. Liu, H. Susanto, G. Xue, and T. Sun, "Incentive mechanism for proximity-based mobile crowd service systems," in *IEEE INFOCOM*, 2016.

[27] Y. Wei, Y. Zhu, H. Zhu, Q. Zhang, and G. Xue, "Truthful online double auctions for dynamic mobile crowdsourcing," in *IEEE INFOCOM*, 2015.

[28] P.-J. Wan, D.-Z. Du, P. Pardalos, and W. Wu, "Greedy approximations for minimum submodular cover with submodular cost," *Comput. Optim. Appl.*, vol. 45, no. 2, pp. 463–474, 2010.

[29] J.-P. Aumasson, L. Henzen, W. Meier, and R. C.-W. Phan, "Sha-3 proposal blake," *Submission to NIST*, 2008.

[30] R. B. Myerson, "Optimal auction design," *Math. Oper. Res.*, vol. 6, no. 1, pp. 58–73, 1981.

[31] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE. Trans. Mob. Comput.*, vol. 17, no. 12, pp. 2982–2997, 2018.

[32] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Online at: https://gavwood.com/paper.pdf*, 2014.

[33] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo, and K. Ren, "Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization," in *IEEE INFOCOM*, 2018.

[34] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *ACM SenSys*, 2015.

[35] C. Miao, L. Su, W. Jiang, Y. Li, and M. Tian, "A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems," in *IEEE INFOCOM*, 2017.

[36] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H. Jin, "Sdte: A secure blockchain-based data trading ecosystem," *IEEE Trans. Inf. Forensic Secur.*, vol. 15, pp. 725–737, 2020.

[37] Z. Zheng, Y. Peng, F. Wu, S. Tang, and G. Chen, "An online pricing mechanism for mobile crowdsensing data markets," in *ACM MobiHoc*, 2017.

[38] Z. Cai and Z. He, "Trading private range counting over big iot data," in *IEEE ICDCS*, 2019.

[39] X. Liu, B. Qin, R. H. Deng, and Y. Li, "An efficient privacy-preserving outsourced computation over public data," *IEEE Trans. Serv. Comput.*, vol. 10, no. 5, pp. 756–770, 2017.

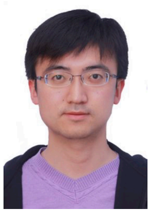
[40] C. Cai, Y. Zheng, Y. Du, Z. Qin, and C. Wang, "Towards private, robust, and verifiable crowdsensing systems via public blockchains," *IEEE Trans. Dependable Secur. Comput.*, pp. 1–1, 2019.

[41] K. Liu, X. Qiu, W. Chen, X. Chen, and Z. Zheng, "Optimal pricing mechanism for data market in blockchain-enhanced internet of things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9748–9761, 2019.

[42] G. Gao, M. Xiao, J. Wu, S. Zhang, L. Huang, and G. Xiao, "Dpdt: A differentially private crowd-sensed data trading mechanism," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 751–762, 2020.

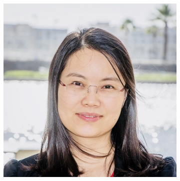
**Baoyi An** received her B.S. degree from the School of Computer Science and Technology at the Nanjing Normal University (NNU), Nanjing, China, in 2017. She is currently a Ph.D. student on computer science and technology with the School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, China. Her research interests include data trading, blockchain, privacy preservation, and incentive mechanism.

**Mingjun Xiao** is a professor in the School of Computer Science and Technology at the University of Science and Technology of China (USTC). He received his Ph.D. from USTC in 2004. His research interests include crowdsourcing, mobile social networks, vehicular ad hoc networks, mobile cloud computing, auction theory, data security and privacy. He has published more over 100 papers in referred journals and conferences, including TMC, TC, TPDS, TKDE, ToN, INFOCOM, ICDE, etc. He served as the TPC member of INFOCOM'21, IJCAI'21, INFOCOM'20, INFOCOM'19, ICDCS'19, INFOCOM'18, etc. He is on the reviewer board of several top journals such as TMC, TON, TPDS, TSC, TVT, TCC, etc.

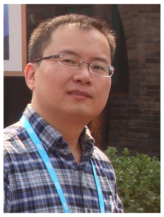


**An Liu** is a professor in the Department of Computer Science and Technology at Soochow University. He received his Ph.D. degree in computer science from both City University of Hong Kong (CityU) and University of Science and Technology of China (USTC) in 2009. His research interests include spatial databases, crowdsourcing, data security and privacy, and cloud/service computing. He has published more than 80 papers in referred journals and conferences, including IEEE TKDE, IEEE TSC, GeoInformatica, KAIS, ICDE, WWW etc. He served as the Workshop Co-Chairs of WISE 2017 and DASFAA 2015. He is on the reviewer board of several top journals such as IEEE TKDE, IEEE TSC, IEEE TII, IEEE TCC, ACM TOIT, JSS, DKE, FGCS, WWWJ, and JCST.

**Yun Xu** received the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, in 2002. He is currently a professor with the School of Computer Science of USTC and a member of National High Performance Computing Center at Hefei. His research interests include bioinformatics, biological sequence analysis and mining and parallel algorithms. He is a member of the IEEE and the ACM.



**Xiangliang Zhang** received the Ph.D. degree from INRIA-University Paris-Sud 11, France, in 2010. She is currently an Associate Professor and directs the Machine Intelligence and kNowledge Engineering (MINE) Laboratory, King Abdullah University of Science and Technology. Her main research interests and experiences are in diverse areas of machine learning and data mining. She has authored over 100 papers in referred journals and conferences, including IEEE TKDE, VLDB J, SIGKDD, VLDB, ICDE, WWW, WSDM, AAAI, IJCAI, ICDM, ECML/PKDD, CIKM, and InfoCom.



**Qing Li** received the B.Eng. degree from Hunan University, Changsha, China, and the M.Sc. and Ph.D. degrees from the University of Southern California, Los Angeles, all in computer science. He is currently a Chair Professor (Data Science) and the Head of the Department of Computing, the Hong Kong Polytechnic University. He is a Fellow of IET/IEE, a Senior Member of IEEE, a member of ACM-SIGMOD and IEEE Technical Committee on Data Engineering. He serves as a Steering Committee member of DASFAA, ER, ICWL, UMEDIA, and WISE Society. His current research interests include Multi-modal data management, Data warehousing and mining, Social media and Web services, and e-Learning Technologies.