2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS) | 978-1-6654-4935-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/MASS52906.2021.00089

Privacy-Enhanced Crowdsourcing Data Trading based on Blockchain and Stackelberg Game

Zhiyuan Huang*, Jun Zheng†, and Mingjun Xiao*✉
*School of Computer Science and Technology / Suzhou Institute for Advanced Study,
University of Science and Technology of China, Hefei, China
†School of Software Engineering / Suzhou Institute for Advanced Study
University of Science and Technology of China, Hefei, China
✉Correspondence to: xiaomj@ustc.edu.cn

***Abstract*—Crowdsourcing Data Trading is a novel paradigm, where a platform can aggregate data collected by a group of mobile users. Traditional crowdsourcing data trading systems rely on trusted data trading brokers, which increase costs and cannot prevent collusion. What's more, they don't take of data quality and truthfulness into consideration concurrently. The emergence of blockchain and smart contracts brings in a decentralized scheme. However, the selection of the most effective providers remains challenging. To tackle these problems, we propose a dynamic-game-with-complete-information-and-Blockchain-based Crowdsourcing data Trading system (CBCT), which mainly includes a smart contracts called CBCToken. Firstly, we replace the data trading broker and adopt the Stackelberg Game, a Dynamic Game with complete information (DGC), to manage the selection of providers.Moreover, homomorphic watermarking technology is applied to protect the data copyright. Lastly, we deploy BCDToken on an Ethereum test network to demonstrate its practicability and significant performances.**

***Index Terms*—Blockchain, Dynamic Game with Complete Information, Stackelberg Game, Homomorphic Watermarking**

I. Introduction

Along with the rapid development of computer applications and Internet technology, which will create a large amount of data every second, we have entered an era of big data. Lots of online data trading systems have appeared these years [1]. Although the total amount of data is very huge, most of it is owned by only a few companies and institutions, which they treat as their private property, and others in need couldn't obtain it. Moreover, the total amount of data in trading systems is very limited [2]. To handle this problem, Crowdsourcing Data Trading (CDT) is put forward, which is a new data trading paradigm combining Mobile Crowdsensing, taking advantage of the crowds to collect data and tackling the scarcity of data sources for sale.

A typical crowdsourcing data trading system mainly includes four entities: a data broker, some data sellers, some data consumers, and data storage as shown in Fig. 1. Yet the traditional crowdsourcing data trading system relies on a Trusted Third Party (TTP), which results in untrustworthy and high cost. Zhao et al. [3] adopts differentially private to ensure safety of crowdsensing system. Xiao et al. [4] designs a reverse auction model to solve unknown worker recruitment problem. Li et al. [5] uses double auction and Bayesian game to solve the problem of cloud resource allocation among multiple users and suppliers. Xiao et al. [6] propose an Average makespan sensitive Online Task Assignment algorithm and a Largest makespan sensitive Online Task Assignment algorithm to solve sensitive task assignment problems.

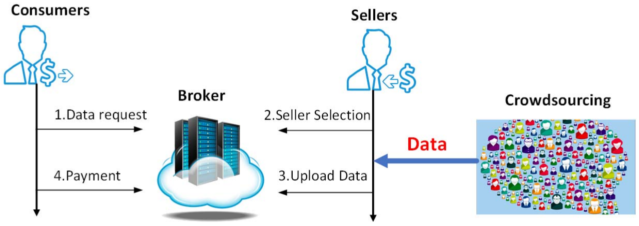

Fig. 1. An Example of a CDT system

As a novel technology, blockchain has attracted wide attention because of its immutable and verifiable properties. By leveraging the blockchain, users who don't trust each other can safely finish data exchange and currency transfer without requiring a TTP, avoiding high legal and transaction cost [7]. [8-9] proposes a traditional data trading system based on blockchain, which allows buyers to buy aggregate results calculated by blockchain nodes over the raw data. Hui et al. [10] designs a crowdsourcing data trading system based on reverse auction on blockchain. The introduction of blockchain technology into data trading system ensure transparency in the trading process. However, simply combining blockchain with crowdsourcing data trading cannot guarantee the truthfulness of individuals, lacks a credible confirmation of identified violations, and still faces challenges in achieving consensus outside the blockchain.

In this paper, we propose a dynamic-game-with-complete-information-and-Blockchain-based Crowdsourcing data Trading system (CBCT). Our CBCT includes a contract called CBCToken corresponding to the DGC, which we model as a Stackelberg Game. And we let the InterPlanetary File System (IPFS) do the job of data storage. The IPFS has a closed relationship with the blockchain and thus is suitable for the data transmission. To be specific, the contributions of the paper are summarized as follows:

978-1-6654-4935-9/21/$31.00 ©2021 IEEE
DOI 10.1109/MASS52906.2021.00089

1) We proposed a CDT system based on DGC and blockchain, it's a decentralized scheme, leveraging smart contract of the blockchain instead of a third-party agent to be the data broker. The data sellers and consumers are anonymous users on the blockchain.
2) We design our CBCT with a smart contract called CBCToken. Firstly, CBCToken adopts the DGC to manage the selection of the cost-effective data sellers to begin the transaction, which we solve with a Stackelberg Game. We disign an incentive mechanism based on the Stackelberg Game and derive an optimal incentive strategy whereby each participant can maximize its profit. Every participant must obey this strategy to protect its profit since the optimal stragegy consists of an unique Stackelberg Equilibrium (SE).
3) We propose a homomorphic watermarking technology based on blockchain to protect the data copyright of the whole data trading. By leveraging homomorphic encryption, our copyright protection scheme is easy for finding the data pirates and can be demonstrated to be quite safe.
4) We implement a prototype of the CBCT system and deploy it to an Ethereum test network. The extensive simulations demonstrate that our CBCT system has low monetary cost and time cost.

II. The CBCT Framework Design

A. Framework Overview

In this section, we propose a dynamic-game-with-Incomplete-information-and-Blockchain-based Crowdsourcing data Trading system, i.e., the CBCT, which mainly includes a consumer, some sellers, a smart contracts deployed on the blockchain, i.e., CBCToken, and an IPFS-based Data Storage under Blockchain(IDSB). These components are illustrated in Fig.2 and can also be defined as follows:

Definition 1 (Consumer, and Job). As a key role of CDT, consumer wants to buy the statistics. $Job \triangleq \langle \mathcal{M}, \tau_{start}, \tau_{end}, \varepsilon, PK_c, FP_c^{enc} \rangle$, where $\mathcal{M} = \{1, 2, \ldots, M\}$ includes M location-related sensing tasks, each corresponding to to a Point of Interest(PoI). ε is the minimum quality requirements for the collected data and aggregated statistics. τ_{start} and τ_{end} are the earliest start-time and the latest finish-time of the data collection,respectively. PK_c is the public key of consumer, DF_c^{enc} is the consumer's digital fingerprint encrypted by PK_c.

Definition 2 (Sellers, Sensing Quality, Participation Level, and benefits). Sellers are a crowd of mobile users who participate in the trade, denoted by $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$. $s_{n,m}$ denotes the sensing quality of seller $i (i \in \mathcal{N})$ completing data collection with sensing task $m (m \in \mathcal{M})$. We assume that the seller will keep the same sensing quality at different POIs, i.e., $s_{i,m} \equiv s_i, \forall m \in \mathcal{M}$. And each seller i will collect data from all M PoIs. Each seller (crowdsourcing worker) has a participation level [11], e.g., sensing data transmission frequency or sensing resolution, denoted by p_i. We let $P \triangleq (p_1, p_2, \ldots, p_N)$ and P_{-i} denote the participation levels of all sellers and all sellers except seller i, respectively. What's more, we let $\Psi(p_i, P_{-i})$ denotes the external benefits p_i gets from social network effects. In a crowdsourcing system, a crowdsourcing worker (i.e. seller) can enjoy some additional benefits from information shared by others [12]. $\mathcal{Q}_i$ represents the seller i's payment.

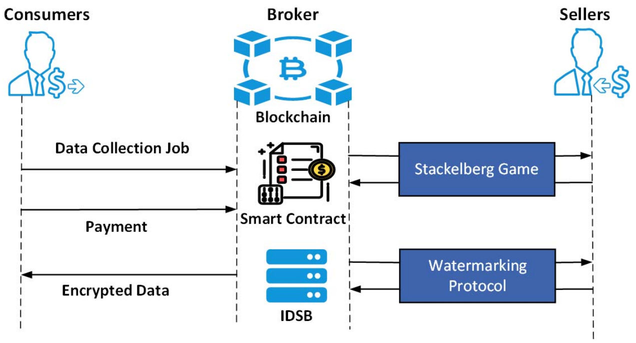

Fig. 2. The Framework of CBCT

Definition 3 (CBCToken). CBCToken is a smart contract deployed on the blockchain, which is also the fundamental smart contract of the system, dealing with all of the issues during the transaction. CBCToken acts as the broker between the consumer and sellers of the CDT system instead of the TTP. CBCToken adopts the DGC, which is a Stackelberg Game and we develop a sophisticated Stackelberg Game framework which is more realistic since it is dynamic.

Definition 4 (IPFS-based Data Storage under Blockchain, IDSB). IDSB is a blockchain-based distributed Data Storage system using Inter-planetary File System (IPFS) protocol, which is a peer-to-peer distributed file system. When a seller uploads his data to the IPFS, it will encrypt the data and return a unique hash address, through which the consumer can acquire data. IDSB provides a persistent and reliable storage for our data.

Definition 5 (Blockchain-based Homomorphic Encryption Fingerprint, BHEF). BHEF is a Blockchain-based Homomorphic Encryption Fingerprint protocol that is used to protect the trading data copyright. It is built on top of homomorphic encryption [13], which is a public-key cryptosystem with a homomorphic property that we can perform computations on its encrypted data without first decrypting it. These rusulting computations are left in an encrypted form which, when decrypted, result in an identical output to that produced had the operations been performed on the unencrypted data. Let $\oplus$ and $\otimes$ represent the addtion and multipulation operation in the prime field Φ_q, i.e., for $\forall x, y \in \Phi_q, x \oplus y \triangleq x + y \mod q$ and $x \otimes y \triangleq x * y \mod q$. Then, the homomorphic encryption

```
Initilaization():
 1. Require $msg.value \geq \$foregift$.
 2. Set $consumer = msg.sender$ and sore $\mathcal{M}, K, L, \varepsilon, PK_c$.
 3. Set $T_{trade} = now + \tau_{trade}, T_{viodet} = T_{trade} + \tau_{viodet}$,
    $T_{delivery} = T_{viodet} + \tau_{delivery}$.
 4. Trigger $Notify$ event to inform the registered sellers.
CaculateRefund():
 1. Require $msg.sender = consumer$.
 2. compute the remaining reward:
    $\$rewards = \$rewards - \sum_{n \in \mathcal{N}} R_i p_i - TF$.
 3. delivery $\$rewards$ to $msg.sender$.
CaculatePayment():
 1. Require $ViolationDetection(msg.sender) = false$.
 2. delivery $\Omega_{msg.sender}$ to $msg.sender$.
```

Fig. 3. The initiate, refund and payment procedure of CBCToken

scheme satisfies

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

where $m_1, m_2 \in \varPhi_q$ are two plaintexts, and $E[\cdot]$ is the homomorphic encryption operation.

B. The Workflow of the CBCT

Phase 1: Initializaiton. All participants of the blockchain first register in the CBCToken. Then every participant should offer $ethers \geq \$foregift$ to CBCToken, where $\$foregift$ is the minimum margin requirement to enter the CDT. When a consumer starts a CDT, it will send a msg(i.e., transaction) to invoke the function $Initialization$ in CBCToken. The detailed function is given in Fig. 3. First of all, it checks if the consumer's foregift (i.e., $msg.value$) is no less than the foregift threshold $foregift$, i.e.,$msg.value \geq \$foregift$. Afterwards, the account of the message sender ($msg.sender$) is recorded by the function as the consumer. The function also stores its sensing tasks $\mathcal{M}$, the data quality requirement ε and consumer's public key PK_c. Next, it sets some time constraints for the subsequent phases, $< \tau_{trade}, \tau_{viodet}, \tau_{delivery} >$ represents the time duration threshold for data trading, violation detection and delivery copyrighted data, respectively. Last but not least, the function emits $Notify$ event to inform all registered data sellers of the CBCT system.

Phase 2: Trading. The trading phase is organized by CBCToken smart contract, which mainly contains a one-leader-multi-followers two-stage Stackelberg Game. The first stage is called leader game, in which consumer (i.e., leader) decides his strategies, denoted as p, to maximize its profit. Then in the second stage, the seller i (i.e., follower) competes with other sellers and decides their strategies, denoted as τ_i, according to the strategies of the leader to maximize their profits. The trading part will be explained in detail in Sect. IV.

Phase 3: Delivery data and Harvest After finishing the trading phase, the seller gets the consumer's fingerprint from the CBCToken. Through the BHEF algorithm, each seller embeds the DF_c and DF_i^s into his data $data_i$ to get watermarked data $data_i^w$. Afterwards, each seller uploads $data_i^w$ to IDSB and obtain the hash storage path $Addr_i$ in return. Then the seller uploads the address $Addr_i$ to the CBCToken. The consumer gets the responding address $Addr_i$ from the CBCToken and then download the data $data_i^w$ from IDSB. The BHEF protol wiill be discussed explicitly in Sect. V.

III. Trading and Stackelberg Game

In this section, we propose the Stackelberg Game for CBCT to realize cost-effective data trading. We formulate this problem into a single-leader-multi-followers two-stage Stackelberg Game, in which the consumer is the single leader and the sellers are the multiple followers.

Definition 6 (Seller's Profit). The profit of the each seller i is defined as follows:

$$\Omega_i(p_i, P_{-i}) = R_i(p_i) - \theta_i(p_i) \tag{2}$$

It contains two parts.The first part $R_i(p_i)$ represents the reward paid to seller i from the consumer. We assume that $R(p_i) = R_i p_i$, i.e., the reward is a linear function to the participation level of seller i. The second part, $\theta_i(p_i)$ is the cost function of seller i, which is assumed to be a monotonically increasing, differentiable and strictly convex function. We adopts a widely used quadratic cost function in this paper, similar to the works in [11], [14] and [15]:

$$\theta_i(p_i) = (a_i p_i^2 + b_i p_i)s \tag{3}$$

$\theta_i(\cdot)$ represents seller i's cost attaching to its effort level, i.e., sensing time in [16][17], and participation level in [18], on data collection with the paraments $a_i \geq 0, b_i \geq 0$, s is a tunable parameter denoting the equivalent monetary worth of the sellers' participation level. Then the cost of seller i can be expressed by:

$$\Omega_i(p_i, P_{-i}) = R_i p_i - (a_i p_i^2 + b_i p_i)s \tag{4}$$

Definition 7 (Consumer's Profit). Our CBCT system works as the crowdsourcing service provider(CSP), which is a blockchain-based system so it has to charge a transaction fee. We assume that the CBCT system only charge for a transaction fee, which means the system is not for profit. The consumer's profit is the CSP's revenue minus the transaction fee. And the revenue of the CSP is given by the payoff from all the sellers minus the total rewards to sellers. Thus the consumer's profit can be described as follows:

$$\Phi = \eta \sum_{i=1}^{N} (c_i p_i - d_i p_i^2) - \sum_{i=1}^{N} R_i p_i - TF \tag{5}$$

The first part is the payoff function from all the sellers. we adopts a linear-quadratic function with the decreasing marginal return property, which transform the sellers' participation level into the revenue of the CSP. η is a tunable parameter denoting the equivalent monetary worth of the sellers' participation level, and $c_i, d_i > 0$ are coefficients characterizing the concavity extent of the funciton. Note that this is a widely adopted function as in the literature [19] and [20].

Definition 8 (two-stage single-leader multi-follower Stackelberg Game). We model this problem as a two-stage single-leader multi-follower Stackelberg Game, where the consumer is the leader and sellers are the followers. Both consumer and sellers tries to maximize their profits by determining an optimal parameter in their strategies(called optimal strategy), satisfying:

Stage I (leader game): The consumer determines the rewards, aiming at highest revenue, i.e.,

$$\mathcal{R}^* = argmax_{\mathcal{R}}\Phi \tag{6}$$

$\mathcal{R} = \{R_1, R_2, ...R_N\}$ denotes the rewards to all sellers from the consumer.

Stage II (follower game): Every seller chooses the participation level p_i. The rewards $\mathcal{R}$, the participation levels of others P_{-i} is already known. The goal of the follower game is to maximize every follower's profit, i.e.,

$$p_i^* = argmax_{x_i}\Omega_i(p_i, P_{-i}, \mathcal{R}) \tag{7}$$

Definition 9 (Stackelberg Equilibrium, SE). An optimical incentive strategy $< P^*, \mathcal{R}^* >$ constitutes a SE if and only if the following set of inequalities are satisfied:

$$\Omega_i(P^*, \mathcal{R}^*) \geq \Omega_i(P, \mathcal{R}) \tag{8}$$

$$\Phi(P^*, \mathcal{R}^*) \geq \Phi(P, \mathcal{R}) \tag{9}$$

Def. 9 shows that no one can improves its own profit by deviating from the optimal strategy in the trading part.

We apply the backward induction to solve the Stackelberg Game. Firstly, we analyze the follower game, finding a unique Nash equilibrium for the Sellers, demonstrated by (7). Then the leader determines the optimal rewards $\mathcal{R}^*$ based on the results of the follower game.

A. Follower Game

Based on the rewards $\mathcal{R}$ announced by the leader, the followers compete with each other to maximize their profits.

Lemma 10. If the following conditions are satisfied, there exists a Nash equilibrium in the game.

- The player set is finite.
- The strategy sets are closed, bounded, and convex.
- The utility functions are continuous and quasi-concave in the strategy space.

Proof. In our system, the number of sellers is limited, so the player set is finite. By deriving the first-order derivatives of each seller's profit function $\Omega_i(p_i, P_{-i})$ in (4) with respect to p_i, we can derive that

$$\frac{\partial\Omega_i(p_i, P_{-i})}{\partial p_i} = R_i - 2a_i p_i s - b_i s \tag{10}$$

Based on the first-order derivative of $\Omega_i(p_i, P_{-i})$, we can derive the second-order derivative of $\Omega_i(p_i, P_{-i})$ with respect to p_i as

$$\frac{\partial^2\Omega_i(p_i, P_{-i})}{\partial p_i^2} = -2a_i s < 0 \tag{11}$$

Therefore, we can derive that there exists a Nash equilibrium in the second stage game. □

Theorem 11. In Stage 2, given the rewards $\mathcal{R}$ announced by the leader , each follower i's optimal strategy P_i can be determined:

$$P_i = \frac{R_i - b_i S}{2a_i s} \tag{12}$$

Proof. We can learn that there must be a Nash equilibrium from Lemma 10 and have proved second-order derivative of $\Omega_i(p_i, P_{-i})$ is positive. Then , we can obtain the unique optimal strategy of each seller i by setting $\dfrac{\partial\Omega_i(p_i, P_{-i})}{\partial p_i} = 0$. □

B. Leader Game

According to the above analysis, consumer i , which is the leader in the Stackelberg game, knows that there exists a unique Nash equilibrium among sellers under any given. Therefore, we can substitute (12) into (5), we have

$$\Phi = \eta\sum_{i=1}^{N}(c_i(\frac{R_i - b_i s}{2a_i s}) - d_i(\frac{R_i - b_i s}{2a_i s})^2) - \sum_{i=1}^{N} R_i(\frac{R_i - b_i s}{2a_i s}) - TF \tag{13}$$

Theorem 12. There exists a unique Stackelberg equilibrium $(P_i^*; R_i^*)$, where R_i^* is the unique maximizer of the leader utility in (13) over $R_i^* \in (0, +\infty)$.

Proof. We can get first order derivative of Φ,as follows

$$\frac{\partial\Phi}{\partial R_i} = \frac{\eta c_i - 2R_i + b_i s}{2a_i s} - \frac{\eta d_i R_i - \eta b_i d_i s}{2a_i^2 s^2} \tag{14}$$

Therefore, the second order derivative of Φ is

$$\frac{\partial^2\Phi}{\partial R_i^2} = -\frac{\eta d_i}{2a_i^2 s^2} - \frac{1}{a_i s} < 0 \tag{15}$$

Therefore, according to (15), we can obtain the unique optimal strategy of consumer i by solving $\dfrac{\partial\Phi}{\partial R_i} = 0$.

$$R_i^* = \frac{\eta a_i c_i s + \eta b_i d_i s + a_i b_i s^2}{2a_i s + \eta d_i} \tag{16}$$

Then, we take (16) into (12) to get the optimal solution of seller i.

$$P_i^* = \frac{\eta a_i c_i - a_i b_i s}{4a_i^2 s + 2\eta a_i d_i} \tag{17}$$

Hence, We can prove that there exists a unique Stackelberg equilibrium $(P_i^*; R_i^*)$. □

So far the whole Stackelberg Game is solved.

Watermark generation: the consumer

1. $V_{pbk_s}(S_{prk_s}(FP_s, ARG))) = (FP_s, ARG))$.
2. produce a one time public-private key pair (pbk_c, prk_c).
3.generates a random positive integer r and calculate the watermark $FP_c = r^{(prk_c - FP_s)} \bmod n$.
4. $encFP_c = E_{pbk_c}(FP_c)$
5.$DS = S_{prk_c}(S_{prk_s}(FP_s, ARG))$.
6.sends $E_{pbk_s}(encFP_c, r, S_{prk_s}(pbk_c), FP_s, ARG, DS)$ to the seller.

Watermark Insertion: the seller

1.$D_{prk_s}(E_{pbk_s}(encFP_c, r, S_{prk_s}(pbk_c), FP_s, ARG, DS))$
$= (encFP_c, r, S_{prk_s}(pbk_c), FP_s, ARG, DS)$.
2. $V_{pbk_s}(V_{pbk_c}(DS)) = (FP_s, ARG)$.
3. $encFP_c \times E_{pbk_c}(r^{FP_s}) = r$.
4. $data' = data \oplus FP_s$
5. sends $E_{pbk_c}(data') \otimes encFP_c = E_{pbk_c}(data^*)$, where $data^* = data \oplus FP_s \oplus FP_c$

Piracy Track:

the seller:
1. $extract(data^*_{pirate}, data) = (FP_s, FP'_c)$.
2. sends the record $(data^*_{pirate}, data, encFP_c, r, S_{prk_s}(pbk_c), FP_s, ARG, DS, extract())$ to the arbiter in the IBCToken.

the arbiter:
3. $extract(data^*_{pirate}, data) = (FP_s, FP'_c)$.
4. $V_{pbk_s}(V_{pbk_c}(DS)) = (FP_s, ARG)$.
5. $encFP_c \times E_{pbk_c}(r^{FP_s}) = r$.

Fig. 4. The Blockchain-based Homomorphic Watermarking Protocol

IV. Blockchain-based Homomorphic Watermarking Protocol (BHWP)

Although there are one seller and many consumers, we can match them and consider one seller and one consumer at a time. And all the asymmetric cryptographic algorithms we used in this part are RSA algorithms. The workflow of the BHWP can be divided into the following three phase:

Phase 1: Registration and Initiation. The consumer asks for an identification from the smart contract IBCToken instead of the traditional CA. Considering the unbinding problem, a mature agreement called argument (ARG) can link the data and watermark and states the rights and obligations of consumer and seller.

When the consumer wants to buy the $data$, he sends the purchase request denoted as PR_{data} to the smart contract IBCToken. Then the seller randomly generates a one-time watermark FP_s. After that, the seller signs (FP_s, ARG) using its private key prk_s and sends $(FP_s, ARG, S_{prk_s}(FP_s, ARG))$ to the consumer through IBCToken.

Phase 2: Watermark generation. Firstly, the consumer uses the seller's public key pbk_s to verify that FP_s and ARG are sent from the consumer, denoted as $V_{pbk_s}(S_{prk_s}(FP_s, ARG))) = (FP_s, ARG))$.

Secondly, the consumer produces a one time public-private key pair (pbk_c, prk_c), where $n = pq$ is part of the public key in the RSA algorithm.

Thirdly, the consumer generates a random positive integer r and calculate the watermark (Fingerprint) $FP_c = r^{prk_c - FP_s} \bmod n$, FP_c is effective during this transaction.

Fourthly, the consumer encrypts the watermark FP_c using its public key pbk_c and get its encrypted watermark $encFP_c$, i.e., $encFP_c = E_{pbk_c}(FP_c)$.

Fifthly, the consumer signs $S_{prk_s}(FP_s, ARG)$ with its private key prk_c and gets a dual signature denoted as $DS = S_{prk_c}(S_{prk_s}(FP_s, ARG))$.

At last, the consumer encrypts all the information to be sent with seller's public key pbk_s and send $E_{pbk_s}(encFP_c, r, S_{prk_s}(pbk_c), FP_s, ARG, DS)$ to the seller through the smart contract IBCToken.

$$\begin{aligned} &D_{prk_s}(E_{pbk_s}(encFP_c, r, S_{prk_s}(pbk_c), FP_s, ARG, DS)) \\ &= (encFP_c, r, S_{prk_s}(pbk_c), FP_s, ARG, DS) \end{aligned} \tag{18}$$

$$V_{pbk_s}(V_{pbk_c}(DS)) = (FP_s, ARG) \tag{19}$$

$$\begin{aligned} &encFP_c \times E_{pbk_c}(r^{FP_s}) = E_{pbk_c}(FP_c) \times E_{pbk_c}(r^{FP_s}) \\ &= E_{pbk_c}(r^{prk_c - FP_s}) \times E_{pbk_c}(r^{FP_s}) = E_{pbk_c}(r^{prk_c}) \\ &= E_{pbk_c}(D_{pbk_c}(r)) = r \end{aligned} \tag{20}$$

Phase 3: Watermark Insertion. Firstly, the seller decrypts what he received from the IBCToken with its private key prk_s, denoted as (18). Then the seller verifies the dual signature denoted as (19) and the consumer's watermark denoted as (20). Only when both the verification is correct, the transaction will continue.

Secondly, since the RSA algorithm has privacy homomorphism, we use the $\oplus$ and $\otimes$ to represent the watermark insertion algorithm, respectively. The seller inserts FP_s into $data$ and gets $data' = data \oplus FP_s$, then he encrypts $data'$ with the consumer's public key pbk_c and gets $E_{pbk_c}(data')$. Next, he inserts the encrypted watermark $encFP_c$ into $E_{pbk_c}(data')$, denoted as (21). And sends $E_{pbk_c}(data^*)$ to the consumer by IDSB. After that, he uploads the record $(encFP_c, r, S_{prk_s}(pbk_c), FP_s, ARG, DS)$ to the IBCToken as well.

$$\begin{aligned} &E_{pbk_c}(data') \otimes encFP_c = E_{pbk_c}(data') \otimes E_{pbk_c}(FP_c) \\ &= E_{pbk_c}(data' \oplus FP_c) = E_{pbk_c}(data \oplus FP_s \oplus FP_c) \\ &= E_{pbk_c}(data^*) \end{aligned} \tag{21}$$

Thirdly, the consumer decrypts $E_{pbk_c}(data^*)$ with its private key prk_c and gets $data^*$.

$$D_{prk_c}(E_{pbk_c}(data^*)) = data^* \tag{22}$$

Phase 4: Piracy Track. Each seller's extract algorithm is different and can only extract the watermark he inserts before.

Firstly, when a pirated data $data^*$ is found from anywhere (e.g., Internet), he can use the watermark extract algorithm to extract and gets FP_s and FP_c, then he can search the record in IBCToken and find pirate consumer. If he cannot extract the data or find the record in IBCToken, it's not a pirated data of his.

Secondly, the seller sends the record, the watermark extract algorithm and the pirtated data to the arbiter in the IBCToken, the arbiter verifies the evidence and make judgment through IBCToken.

V. Experiment

A. Evaluation of Smart Contract

We implement the smart contract by utilizing the Solidity programming language and write a JavaScript test file

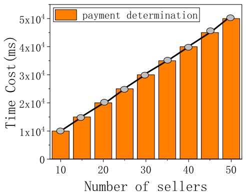

Fig. 5. Time Cost

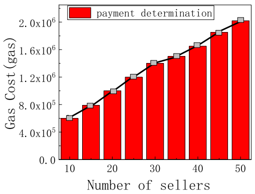

Fig. 6. Gas Cost

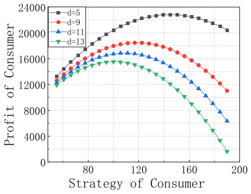

Fig. 7. Profit of Consumer

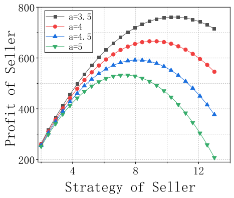

Fig. 8. Profit of Seller

to evaluate the Time cost and Gas cost. We choose the most popular development framework Truffle to simulate the Ethereum network to test smart contract. In Fig.5 and Fig.6, the main time overhead is spent on the functions of payment determination data transportation and calculation. The publish request and participation request will store a lot of data to Ethereum and the gas cost of storing data in Ethereum is very expensive.

B. *Evaluation of Stackelberg game*

For the evaluation criteria, we adopt two main metrics: profit and strategy. Moreover, we use PoC, PoS(s), SoC and SoS(s) to denote the Profit of Consumer, Profit of Seller(s), Strategy of Consumer and Strategy of Seller(s), respectively.First,we evaluate PoC under different consumer's data valuation parameter d when we increase the value of SoC in Fig.7. We can see that each PoC will find a maximum point (i.e., SE point) when increases from 0 to 200. Then, we set optimal SoC to observe the detailed change of PoS of sellerwhen we increase the value of SoS in Fig.8. As illustrated in Fig.8, PoC will find a maximum point (the SE point) of PoS.

VI. Conclusion

In this paper, the proposed CBCT system using smart contract CBCToken combing with other blockchain issues to replace the third-party broker to significantly improve its safety. What's more, we focus on the optimal incentive strategy design in a practical CDT system which is dynamic and with incomplete information. We model it as a Bayasian-Stackelberg Game problem and get the closed-form solution. Finally, we implement a prototype on Ethereum test network and the extensive simulations prove its excellent performance.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (NSFC) (Grant No. 61872330, 61572457, 61379132) , and the NSF of Jiangsu Province in China (Grant No. BK20191194, BK20131174, BK2009150).

References

[1] T. Jung et al., "AccountTrade: Accountable protocols for big data trading against dishonest consumers," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, 2017, pp. 1-9, doi: 10.1109/INFOCOM.2017.8057004.

[2] Z. Zheng, Y. Peng, F. Wu, S. Tang and G. Chen, "Trading Data in the Crowd: Profit-Driven Data Acquisition for Mobile Crowdsensing," in IEEE Journal on Selected Areas in Communications, vol. 35, no. 2, pp. 486-501, Feb. 2017, doi: 10.1109/JSAC.2017.2659258.

[3] Hui Zhao, Mingjun Xiao, Jie Wu, Yun Xu, He Huang, and Sheng Zhang. Differentially Private Unknown Worker Recruitment for Mobile Crowdsensing Using Multi-Armed Bandits. IEEE Transactions on Mobile Computing, DOI: 10.1109/TMC.2020.2990221, 2020.

[4] Mingjun Xiao, Baoyi An, Jing Wang, Guoju Gao, Sheng Zhang, and Jie Wu. CMAB-Based Reverse Auction for Unknown Worker Recruitment in Mobile Crowdsensing. IEEE Transactions on Mobile Computing, DOI: 10.1109/TMC.2021.3059346, Feb. 15, 2021.

[5] Q. Li, C. Huang, H. Bao, B. Fu, and X. Jia, "A game-based combinatorial double auction model for cloud resource allocation," in 2019 28th International Conference on Computer Communication and Networks (ICCCN). IEEE, 2019, pp. 1–8.

[6] Mingjun Xiao, Jie Wu, Liusheng Huang, Ruhong Cheng, and Yunsheng Wang. Online Task Assignment for Crowdsensing in Predictable Mobile Social Networks. IEEE Transactions on Mobile Computing, Aug., 2017, 16(8): 2306-2320.

[7] 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.

[8] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H.Jin, "Sdte: A secure blockchain-based data trading ecosystem" IEEE Transactions on Information Forensics and Security, vol. 15, pp.725– 737, 2020.

[9] C. Cai, Y. Zheng, Y. Du, Z. Qin, and C.Wang, "Towards private, robust, and verifiable crowdsensing systems via public blockchains," IEEE Transactions on Dependable and Secure Computing, pp. 1– 1, 2019.

[10] 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.

[11] M. H. Cheung, F. Hou, and J. Huang, "Make a difference: Diversity-driven social mobile crowdsensing," in IEEE INFOCOM, 2017.

[12] D. Easley and J. Kleinberg, Networks, Crowds, and Markets: Reasoning about a Highly Connected World. Cambridge, U.K.: Cambridge Univ.Press, 2010

[13] 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

[14] X. Duan, C. Zhao, S. He, P. Cheng, and J. Zhang, "Distributed algorithms to compute walrasian equilibrium in mobile crowdsensing," IEEE Trans. Ind. Electron., vol. 64, no. 5, pp. 4048–4057, 2017.

[15] Y. Zhan, C. H. Liu, Y. Zhao, J. Zhang, and J. Tang, "Free market of multi-leader multi-follower mobile crowdsensing: An incentive mechanism design by deep reinforcement learning," IEEE. Trans. Mob.Comput., vol. 19, no. 10, pp. 2316–2329, 2020.

[16] 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 of Things Journal, vol. 6, no. 6, pp. 9748– 9761, 2019.

[17] 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.

[18] Aitzhan, N.Z., Svetinovic, D.: Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans. Dependable Secure Comput. 15(5), 840–852 (2018)

[19] O. Candogan, K. Bimpikis, and A. Ozdaglar, "Optimal pricing in networks with externalities," Oper. Res., vol. 60, no. 4, pp. 883–905,2012.

[20] C. Joe-Wong, S. Ha and M. Chiang, "Sponsoring mobile data: An economic analysis of the impact on users and content providers," in Proc. IEEE INFOCOM, 2015