Decentralized Privacy-Preserving Reputation Management for Mobile Crowdsensing

Lichuan Ma[1,2], Qingqi Pei[1,2], Youyang Qu[3], Kefeng Fan[4], and Xin Lai[5]

[1] Xidian University, Xi'an 710071, China
[2] Shaanxi Key Laboratory of Blockchain and Security Computing
lcma@xidian.edu.cn, qqpei@mail.xidian.edu.cn
[3] Deakin University, Melbourne VIC 3125, Australia
quyo@deakin.edu.au
[4] China Electronics Standardization Institute, Beijing 100007,China
fankf@126.com
[5] Xunlei Network Technologies Limited, Shenzhen 518057, China
laixin@xunlei.com

Abstract. In mobile crowdsensing, mobile devices can be fully utilized to complete various sensing tasks without deploying thousands of static sensors. This property makes that mobile crowdsensing has been adopted by a wide range of practical applications. Since most crowdsensing platforms are open for registration, it is very possible that some participants might be motivated by financial interest or compromised by hackers to provide falsified sensing data. Further, the urgent privacy-preserving need in this scenario has brought more difficulty to deal with these malicious participants. Even though there have existed some approaches to tackle to problem of falsified sensing data while preserving the partipants' privacy, these approaches rely on a centralized entity which is easy to be the bottleneck of the security of the whole system. Hence in this paper, we propose a decentralized privacy-preserving managemen-t scheme to address the problem above. At first, the system model is present based on the consortium blockchain. Then, a novel metric to evaluate the reliability degree of the sensing data efficiently and privately is designed by leveraging the Paillier crytosystem. Based on this metric, how to update reputation values is given. Extensive experiments verify the effectiveness and efficiency of the proposed scheme.

Keywords: Reputation management · Privacy-preserving · Blockchain · Mobile crowdsensing.

1 Introduction

Benefiting from the sensing and communicating technologies, mobile crowdsensing (MCS) has attracted great attentions from both academia and industry. The key idea of MCS is to fully utilize the sensing capabilities of mobile devices to undertake various sensing tasks without deploying thousands of static sensors [1]. Due to this property, MCS has been already adopted by different practical

applications, like road surface alarming [2], air condition monitoring [3], smart city [4] and electronic healthcare [5].

In order to encourage more people to be enrolled in undertaking MCS tasks, most of the MCS platforms are open for registration and anyone that possesses a mobile device can contribute sensing data according to different tasks. This property of MCS makes it difficult to guarantee all the participants to be reliable. Motivated by financial interest or compromised by hackers, some participants can become malicious to provide falsified sensing data to change the final aggregating result. As stated by [6], even a single forged data can make the aggregating result very different from the original one.

When deal with malicious participants that provide falsified data, reputation management mechanisms are usually introduced and their effectiveness has been verified in different scenarios, like cognitive radio networks [7] and online social networks [8]. In these mechanisms, the reliability of sensing data is firstly evaluated. After that, a reputation value for the corresponding data provider is updated according to the degree of the sensing data reliability. By doing this, the reputation value of a malicious participant can be very low. When setting the reputation value as the weight of the according sensing data for aggregation, the influence of the sensing data from malicious participants can be constrained.

However, designing a reputation mechanism for MCS is never a simple task. The first reason is that urgent privacy-preserving needs make it difficult to update the reputation values of the MCS participants. The sensing data in MCS are usually tagged with time and location information. When directly sharing these data, a lot of private information can be inferred and this leads to high privacy leakage risks [2]. A great number of approaches have been proposed to preserve privacy for MCS participants. Generally, these approaches can be classified into two categories: anonymity-based, and encryption-based. The goal of anonymity-based approaches is to make it impossible to link particular sensing data to the related providers [9]. In encryption-based approaches, all the original sensing data are encrypted and only legitimate entities can decrypt them [10]. Without specific settings, anonymity-based approaches make it impossible link the reliability degree of the sensing data to the corresponding provider and cryptography-based approaches make it difficult to evaluate the reliability degree of the sensing data.

The second reason is that centralized reputation management mechanisms would not satisfy the requirements of MCS. Currently, most reputation management mechanisms rely on a centralized reputation manager to evaluate the reliability of the provided data and update the reputation values. As the sensing data become increasingly fine-grained and complicated, more and more data should be sensed, transmitted, and processed [2]. Meanwhile, many MCS tasks are geographically distributed and many sensing tasks might happen in a parallel manned at the same time. Only utilizing one reputation manager would lead to high delays for updating reputation values and this offers more chances for malicious participants to provide falsified sensing data. Moreover, such a cen-

tralized entity can be an obvious target of different cyber attacks and thus is the bottleneck of the whole system security [11].

Hence in this paper, we aim to work out a distributed privacy preserving reputation management scheme for MCS to conquer the above two challenges. The contributions of our work are summarized as follows.

- By adopting the concept of edge computing and the consortium blockchain, we firstly present system model for the decentralized privacy-preserving reputation management scheme.
- By leveraging the Paillier cryptosystem, we then design a novel metric to evaluate the reliability degree of the sensing data efficiently and privately. Based on that, the rule for updating reputation values are proposed.
- Extensive experiments verify that the proposed scheme are efficient and effective to deal with the malicious participants that provide falsified sensing data.

The rest of the paper is organized as follows. Related work is summarized in Section 2. Preliminaries utilized in this paper are introduced in Section 3. A simple introduction of the proposed system model is presented in Section 4. The distributed privacy-preserving reputation management scheme is introduced in Section 5. The efficiency and effectiveness of the proposed scheme are verified in Section 6. Section 7 concludes the paper.

2 Related Works

To motivate more owners of mobile devices to undertake MCS tasks, most MCS platforms are open for registration. This leads to that the reliability of the participants in MCS can not be guaranteed. Motivated by financial interests or compromised by hackers, some participants would become malicious to provide falsified sensing data to influence the final aggregating results. Reputation management schemes have been proved to be effective to deal with the malicious participants that provide falsified data in different scenarios, like [7] and [8].

As the sensing data are usually tagged with time and space information, directly sharing these data can bring great privacy leakage concerns among the participants. As a result, how well the privacy of the participants can be preserved determines how far MCS can go. This makes privacy preserving for MCS a hot research topic recently and a great number of approaches have been proposed. Generally, these approaches can be classified into two categories, anonymity-based and encryption-based. The goal of anonymity-based approaches is to make it impossible to link particular sensing data to the corresponding provider [9]. However, according to [12], this kind of approaches are fragile to tracing attack. With respect to the encryption-based approaches, the original data are encrypted and higher security level can be guaranteed via introducing many complex computations [10]. These two kinds of approaches have shown their high performance to preserve the privacy of MCS participants. But without of specific settings, these approaches make it difficult to introduce reputation

management schemes to deal with malicious participants. In other words, pure privacy-preserving approaches might increase the chance that the whole MCS system suffer from falsified sensing data of malicious participants.

Recently, there have been some works trying to propose reputation management schemes with privacy preserving functions. In [9], the authors propose a anonymity-based privacy-preserving reputation management scheme for crowdsensing. As the participants are kept anonymous during the process of undertaking tasks, an additional redeeming phase is utilized to update reputation values. Apart from the fact that anonymity-based approaches are vulnerable to tracking attacks, malicious participants are possible to provide falsified sensing data with high reputation values for somewhile if they refuse to go on the redeeming phase. To conquer the drawbacks in [9], our previous work [13] introduces the somewhat homomorphic encryption scheme to achieve privacy-preserving reputation management. Compared with [9], this work achieves higher security level but homomorphic encryption based sensing data aggregation and reputation values update are time-consuming. Moreover, both in [9] and [13], the reputation management schemes rely on a centralized manager. According to our previous analysis, such a centralized entity cannot satisfy the requirements of geographically distributed MCS tasks and is easy to be the bottleneck of the security of the whole system.

To eliminate the centralized reputation manager in the reputation management scheme, the blockchain technology is considered to be very promising to achieve this. As the underlying foundation of Bitcoin, blockchain is an open and distributed ledger maintained by all the entities in the network [14]. This makes blockchain to be reliable and tamper-proof. Due to its high security and reliability, blockchain has been introduced to different practical application scenarios, like mobile crowdsourcing [11], smart grid [14], IoT [15], and vehicular networks [16]. In [11], the authors design a private and anonymous crowdsourcing system based on open blockchain. The goal of this work is to guarantee the workers that undertake outsourcing tasks can really gain the payoff as claimed. The authors of [14] design a blockchain-based energy trading system for typical energy trading scenarios with moderate cost. In [15], blockchain is introduced to achieve secure distributed data storage for IoT. The above approaches in [11, 14, 15] verify the great potential for blockchain to realize distributed schemes to conquer the drawbacks of centralized ones. With respect to reputation management, the authors of [16] proposes a decentralized reputation management for vehicular networks. This work is inspiring but the privacy of the original data are not considered when updating the reputation values.

To summarize, designing a decentralized privacy-preserving reputation management scheme is of far-reaching significance for the further development of MCS. Although there have existed some works trying to solve this problem, they cannot satisfy the requirements of privacy-preserving and decentralizing goals for reputation management in MCS.

3 Preliminaries

In this section, we will simply introduce the blockchain technology and the Paillier cryptosystem that are essential to achieve our decentralized privacy-preserving reputation management scheme for MCS.

3.1 Blockchain

The definition of blockchain is first proposed in [17]. As the building foundation of Bitcoin, blockchain is a decentralized ledger maintained by all the entities in the network. As each entity stores a copy of this ledger, the blockchain would be tampered if and only if the adversaries compromise more than half of the total nodes in the whole network. This equips blockchain with high reliability and security.

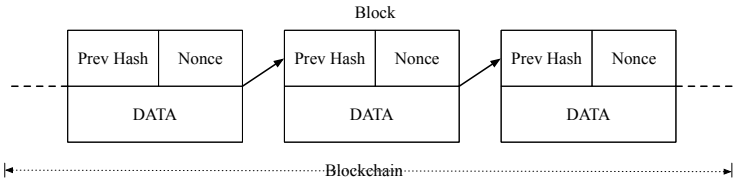

Fig. 1. The blockchain architecture

As is shown in Fig. 1, a blockchain is constructed by sequentially linked blocks. Each block stores the hash value of the previous one, the nonce value and data. Here, the data stored in a block are determined by the actual application scenario, for example, transactions data are stored in Bitcoin and the sensing data would be stored in MCS [11]. Since each block contains the hash value of its previous block, the blocks are linked in a sequential order. This setting is the origin of the tamper-proof property of blockchain. If the adversaries want to tamper a block, all the subsequent blocks should be tampered.

The nonce value is the solution of a mathematical problem and only the nodes that compute the right nonce value are selected as miners. The miners are responsible for writing data in a new block and broadcast this block across the whole network. When other nodes verify the validity of this block, it would be added to the blockchain maintained by these nodes themselves. When more than half of the total nodes add this block to its own blockchain, this block is admitted by all the nodes in the network. This process is the so-called proof-of-work (PoW).

Generally, blockchain is classified into three categories, public blockchain, private blockchain and consortium blockchain. As the participants in MCS are geographically distributed and their number can be very large, it is impossible to let each participant maintain a copy of the blockchain. For this reason, we

introduce the concept of consortium blockchain where only authorized nodes are responsible for maintaining the blockchain.

3.2 Paillier Cryptosystem

To update the reputation values without revealing the original sensing data, we utilize the Paillier cryptosystem in our scheme which supports addition homomorphism. This cryptosystem is first introduced in [18] and it consist of the following three algorithms [19]:

- **Paillier.KeyGen**: this algorithm outputs the public and secret key pair (pk, sk). The detailed process is (1) randomly choose two large primes p and q such that $gcd(pq, (p-1)(q-1)) = 1$; (2) compute $N = pq$ and $\lambda = lcm(p-1, q-1)$; (3) randomly choose g from Z_{N^2} such that $\mu = (L(g^{\lambda}(modN^2)))^{-1}(modN)$, where $L(x) = (x-1)/N$. In this way, we have $pk = (N, g)$ and $sk = (\lambda, \mu)$.
- **Paillier.Enc**: this algorithm outputs the ciphertexts of any message. Given any message m in Z_N, randomly choose r and compute the ciphertext c as $c = g^m \cdot r^N(modN^2)$.
- **Paillier.Dec**: this algorithm is used to decrypt any ciphertext via $m = L(c^{\lambda}(modN^2)) \cdot \mu(modN)$.

One important property of the Paillier cryptosystem is its support of addition homomorphism. Given any two messages m_1 and m_2, their encryptions are $E(m_1, pk) = g^{m_1} r_1^N(modN^2)$ and $E(m_2, pk) = g^{m_2} r_2^N(modN^2)$. Now, the following two equations hold:

$$D(E(m_1, pk) \cdot g^{m_2}) = m_1 + m_2(modN) \tag{1}$$

$$D(E(m_1, pk)^k) = km_1(modN) \tag{2}$$

4 System Model

In our MCS system, we introduce the paradigm of edge computing by deploying geographically distributed edge nodes (ENs) for collecting sensing data and maintaining the blockchain for decentralized privacy-preserving reputation management. As is shown in Fig. 2, the participants can act as both a sensing requester and the one that undertakes sensing tasks. A certificate authority (CA) is adopted to generate the public and secret key pairs for privacy-preserving goals. Here, there is no need to assume ENs following the semi-honest model and they can be considered untrusted. The data stored in a block are the parameters for updating reputation values, the encrypted sensing data and the historical reputation values of the participants.

To achieve the privacy-preserving goal, the proposed scheme is capable of defending against the following two attacks (**A1**) and (**A2**).

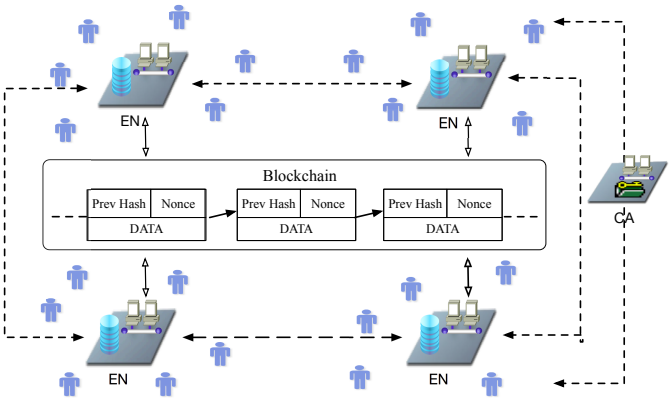

Fig. 2. The system model

- **A1**: The ENs might try to infer the original sensing data from the received encryptions.
- **A2**: The ENs might collude with some participants to infer the original sensing data of the others.

To deal with the malicious participants that provide falsified sensing data, our scheme is designed to conquer the attacks of (**A3**)∼(**A6**).

- **A3**: The malicious participants always provide falsified sensing data when undertaking different sensing tasks.
- **A4**: The malicious participants alternately provide falsified and original sensing data to keep their reputation values at a certain level.
- **A5**: The ENs do not aggregate the sensing data in the predefined manner.
- **A6**: The requester do not provide the actual data that are used to update the reputation values.

In the following sections, we will present the decentralized privacy-preserving reputation management scheme in detail and analyze how the proposed scheme can defend against the attacks of **A1**∼**A6**

5 The Decentralized Privacy-Preserving Reputation Management Scheme

In this paper, the proposed decentralized privacy-preserving reputation management scheme has two phases: completing sensing tasks and updating reputation values. Without loss of generality, how the proposed scheme works is illustrated by completing one sensing task. Before the MCS system begins to work, CA would first send each registered participant i the secret key $sk_{s,i}$ for signing sensing data and the public key $pk_{s,i}$ for verifying the data signed by i. As for any edge node EN_j, sk_{s,EN_j} and pk_{s,EN_j} are also generated by CA.

5.1 Completing the Sensing Tasks

Let q and τ denote the requester and the current sensing task respectively. The edge node near the requester q is denoted by EN_q and the one located in the target area of the sensing task is denoted by EN_τ. Further, let $\mathcal{P}_\tau$ denote the set of the participants that undertake the sensing task τ. The whole process to complete the sensing task τ is shown in Fig. 3.

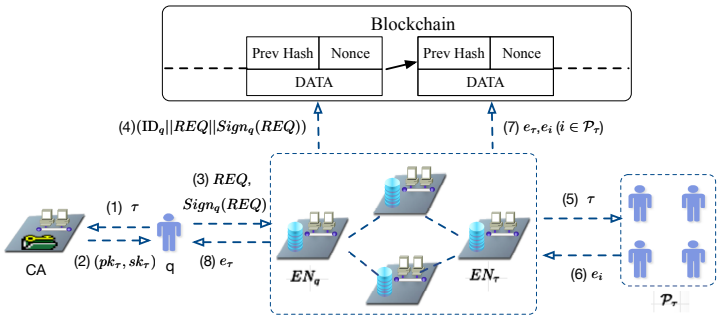

Fig. 3. The whole process to complete the sensing task τ.

Step 1: Before q publish the sensing task τ, he first request the CA to generate the public and secret key pair (sk_τ, pk_τ) for encrypting and decrypting sensing data in the Paillier cryptosystem.

Step 2: Then, the CA runs the **Paillier.KeyGen** algorithm to generate (sk_τ, pk_τ) and send it to q.

Step 3: On receiving the keys, q constructs the sensing task request as $REQ = (\tau||pk_\tau)$ and signs this request to obtain $Sign_q(REQ)$ with $sk_{s,q}$. After that, both REQ and $Sign_q(REQ)$ are sent to EN_q.

Step 4: When REQ and $Sign_q(REQ)$ are received, EN_q verifies the correctness of $Sign_q(REQ)$. If $Sign_q(REQ)$ passes verification, EN_q initiates a request to add $(ID_q||REQ||Sign_q(REQ))$ to the blockchain. After running the PoW consensus process, the miner EN_m is elected to generate a new block that stores $(ID_q||REQ||Sign_q(REQ))$ and broadcasts this newly generated block across the network. The remaining edge nodes verify the correctness of $Sign_q(REQ)$ and determine whether to accept this block. If $Sign_q(REQ)$ passes the verification of all the other edge nodes, the block is finally generated and the sensing task is formally published.

Step 5: The edge node in the target area EN_τ broadcasts REQ to the participants nearby. The ones that want to undertake the sensing task τ form the set $\mathcal{P}_\tau$.

Step 6: For $i \in \mathcal{P}_\tau$, let $\hat{\boldsymbol{s}}_i$ denote its sensing data, which is an m-dimension vector. Here, $\hat{\boldsymbol{s}}_i = \{s_{i,j}, j = 1, \ldots, m\}$. Since the Paillier cryptosystem only

supports positive integers, we need to normalize the original sensing data to obtain $\bar{\boldsymbol{s}}_i$ by:

$$\bar{s}_{i,j} = \frac{\hat{s}_{i,i} - lb_j}{ub_j - lb_j} \tag{3}$$

where, lb_j and ub_j are the lower and upper bounds of the sensing vector's jth dimension. Each element in $\bar{\boldsymbol{s}}_i$ multiplies 10^ϵ and the round number of the result are the values of $\boldsymbol{s}_i$. To preserve the privacy of the participants, $\boldsymbol{s}_i$ would be encrypted with pk_τ to obtain $\boldsymbol{e}_i = \{\textbf{Paillier.Enc}(s_{i,j}), j = 1, \ldots, m\}$. Then, $\boldsymbol{e}_i$ would be sent to EN_τ.

Step 7: After receiving the sensing data, EN_τ gets the reputation values of the participants in $\mathcal{P}_\tau$ from the blockchain. Let rep_i denote the reputation value of i. Next, EN_τ would aggregate the sensing data to get the encryption of the final result $\boldsymbol{e}_\tau$ via ($j = 1, \ldots, m$):

$$e_{\tau,j} = \prod_{i \in \mathcal{P}_\tau} e_{i,j}^{\frac{rep_i}{\sum_{i \in \mathcal{P}_\tau} rep_i}} \tag{4}$$

Then, EN_τ initiates a request to add $\boldsymbol{e}_\tau = \{e_{\tau,j}, j = 1, \ldots, m\}$ and $\boldsymbol{e}_i$ to the request.

Step 8: $\boldsymbol{e}_\tau$ is returned to q with the help of EN_q. After decrypting $\boldsymbol{e}_\tau$ with sk_τ, q can get $\boldsymbol{s}_\tau$. By ϵ, lb_j and ub_j, q can recover the original aggregated sensing data.

Up to now, the sensing task τ is completed. In the following, we will describe how to update the reputation values of the participants in $\mathcal{P}_\tau$.

5.2 Updating Reputation Values

Since the Paillier cryptosystem only supports addition homomorphism, the reputation values cannot be updated based on the deviation from the aggregated result as in our previous work [13]. According to the fact that the sensing data from ordinary participants are very similar but very different from those of malicious participants, we design the rule for updating reputation values as is shown in Fig. 4.

Step 1: At first, the edge node in the target area EN_τ randomly generates a base vector $\boldsymbol{s}_b$ which has m dimensions (as $\boldsymbol{s}_b = \{s_{b,j}, j = 1, \ldots, m\}$) such that $s_{b,j} > max\{s_{i,j} : i \in \mathcal{P}_\tau\}$. Since the values of $\boldsymbol{s}_i$ are obtained by the normalized values multiplying 10^ϵ, we can set the message space of the Paillier cryptosystem to be at least $[0, 2 \cdot 10^\epsilon]$. By this way, the values of $\boldsymbol{s}_b$ can be randomly selected from the range $[10^\epsilon + 1, 2 \cdot 10^\epsilon]$. Then, EN_τ encrypts $\boldsymbol{s}_b$ with pk_τ to get $\boldsymbol{e}_b$.

Step 2: As EN_τ has $\boldsymbol{e}_i$ ($i \in \mathcal{P}_\tau$), $E(d_i)$ can be computed by:

$$E(d_i) = \prod_{j=1}^{m} \frac{e_{b,j}}{e_{i,j}} \tag{5}$$

where $d_i = \sum_{j=1}^{m}(s_{b,j} - s_{i,j})$. For the reason that each elements in $\boldsymbol{s}_b$ is greater than those of the original sensing vector, the value of d_i reflects the deviation

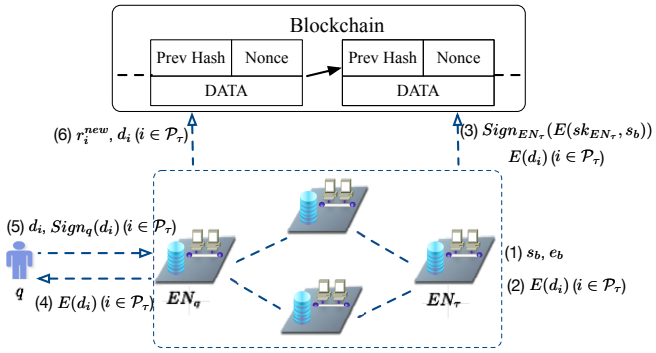

Fig. 4. The whole process to update the reputation values of the participants in $\mathcal{P}_\tau$.

of i's sensing vector from the base vector to some extent. Since the sensing data from ordinary participants are very similar, the values computed through (5) are also very similar.

Step 3: EN_τ encrypts $\boldsymbol{s}_b$ with sk_{EN_τ} and sign the encryption to get $Sign_{EN_\tau}(E(sk_{EN_\tau}, \boldsymbol{s}_b))$. Together with $E(d_i)$ $(i \in \mathcal{P}_\tau)$, EN_τ initiates a request to add them to the blockchain.

Step 4: After the newly generated block is admitted by the whole network, EN_q obtains $E(d_i)$ $(i \in \mathcal{P}_\tau)$ and sends them to the requester q which has the secret key sk_τ.

Step 5: After receiving $E(d_i)$ $(i \in \mathcal{P}_\tau)$, q decrypts them to get d_i $(i \in \mathcal{P}_\tau)$ and signs the decryptions to get $Sign_q(d_i)$ $(i \in \mathcal{P}_\tau)$. Both d_is and $Sign_q(d_i)$s are sent back to EN_q.

Step 6: When obtaining d_is, EN_q classifies them into two groups by the k-means method. At this time, the group with more members is denoted by ***Rew*** and the other is denoted by ***Pel***. According to the assumption in Section 4, the proportion of the malicious participants is less than a half and this make ***Rew*** mainly contain normal participants and their reputation values would be increased. The reputation values of the participants in ***Pel*** would be decreased. Hence, two parameters, α and β, are introduced to control the increase and decrease of the reputation values, respectively. As a result, the rule for updating reputation values is:

$$r_i^{new} = \begin{cases} r_i + (1 - r_i) \cdot \alpha & \text{if } i \in \boldsymbol{Rew} \\ r_i \cdot (1 - \beta) & \text{if } i \in \boldsymbol{Pel} \end{cases} \tag{6}$$

where both α and β are positive and $\beta < 1$. After getting the updated reputation values r_i^{new} $(i \in \mathcal{P}_\tau)$, EN_q would sign them and initiates a request to add d_is and signed r_i^{new}s to the blockchain. When passing all the verifications, the block containing all the updated reputation values are admitted by the network. Up to now, the phase of updating reputation values is finished.

5.3 Security Analysis

In the following, we will show how the proposed scheme can defend against the attacks of **A1**∼**A6**.

A. When $m \geq 2$, the proposed scheme can defend agains the attacks of A1 and A2.

Analysis: From the description above, the edge nodes (both EN_q and EN_τ) can only hold the encrypted sensing data $\boldsymbol{e}_i$ ($i \in \mathcal{P}_\tau$), the encrypted aggregated result $\boldsymbol{e}_\tau$, the base vector $\boldsymbol{s}_b$, and d_i ($i \in \mathcal{P}_\tau$). Since no edge nodes can obtain the private key sk_τ, $\boldsymbol{e}_i$ ($i \in \mathcal{P}_\tau$) and $\boldsymbol{e}_\tau$ cannot be decrypted. Since the base vector is generated by EN_τ and d_is are open to all the edge nodes, EN_τ can compute $s_b - d_i$ to get the original sensing data when $m = 1$. But when $m \geq 2$, there can be infinite choices and no edge nodes can recover the original sensing data. Hence, the proposed scheme can defend agains the attack **A1**. When updating the reputation values, d_i is computed via (5). This computation is irrelevant to other participants' sensing data. Since the edge nodes cannot decrypt $\boldsymbol{e}_i$ ($i \in \mathcal{P}_\tau$), it is impossible for them to collude with some participants to recover the sensing data of the other participants. So, the proposed scheme can also defend agains the attack **A2** when $m \geq 2$.

B. When malicious participants keep providing falsified sensing data, their reputation values converge to 0.

Analysis: Assume that the ith participant in $\mathcal{P}_\tau$ is malicious and he keeps providing falsified sensing data. When updating his reputation values, the computed d_i via (5) is very different from that of normal participants. Since the proportion of malicious participants is less than a half, this participant would be classified into the group ***Pel***. After being updated by (6), his new reputation value would be $r_i \cdot (1 - \beta)$. Since $1 - \beta < 1$ obviously holds, his reputation value would converge to 0 if he keeps providing falsified sensing data when undertaking the following sensing tasks. From this sense, the proposed scheme can defend against the attack of **A3**.

C. When $\frac{\alpha}{\alpha+\beta} < 0.5$, the influence of the attack A4 can be mitigated.

Analysis: Note that the attack of **A4** is known as the famous On-Off attack. As stated in [20], the influence of this attack can be mitigated when the reputation value increase for providing right sensing data is less than the decrease when providing falsified sensing data. Generally, the reputation value of the participant that launches the attack of **A4** is above a certain level (e.g. greater than 0.5). Otherwise, it is of no use for the participants with small reputation values to launch such an attack. Assume that the ith participant in $\mathcal{P}_\tau$ launches this attack and its reputation value r_i is larger than 0.5. According to (6), his new reputation value becomes $r_i + (1 - r_i) \cdot \alpha$ after providing right sensing data. Here, the increase of his reputation value is $(1 - r_i) \cdot \alpha$. When this participant provides falsified sensing data, his new reputation value changes to $r_i \cdot (1 - \beta)$ and the decrease is $r_i \cdot \beta$. If $(1 - r_i) \cdot \alpha < r_i \cdot \beta$ holds, we can get $\frac{\alpha}{\alpha+\beta} < 0.5$.

D. The proposed scheme is able to defend against the attack of A5.

Analysis: In the proposed scheme, it is required that the sensing data should be signed by the providers to prevent being tampered. In the meanwhile, $\boldsymbol{s}_\tau$

should be signed EN_τ so that the aggregated results cannot be denied by EN_τ. Moreover, the encrypted sensing data and the aggregated result should be stored in the blockchain. In this way, the correctness of $\boldsymbol{s}_\tau$ can be also verified. Only when more than half of the edge nodes are compromised, the attack of **A5** cannot be detected. Actually, compromising more than half of the edge nodes is almost impossible to be achieved. Thus, the proposed scheme can defend agains the attack of **A5**.

E. The attack of A6 can be traced through the data stored in the blockchain.

Analysis: When the requester decrypts $E(d_i)$ to get d_i ($i \in \mathcal{P}_\tau$), they should be signed by q. This setting avoids q denying the results of decryption. Since all the data related to $E(d_i)$s are stored in the blockchain and they are open for consulting, any participant that is not satisfied with his reputation value can initiate a request to check whether his reputation value is correctly updated. This request can be fulfilled with the help of CA who generates the keys. In this way, the attack of **A6** can be detected if it really exists.

6 Experimental Results

In this section, we will carry out extensive experiments to verify the effectiveness of the proposed scheme. At first, we show the influence of different values of α and β on updating reputation values. Then, the running time efficiency of the proposed scheme is analyzed. At the end of this section, the effectiveness to defend against malicious participants is demonstrated.

For the following experiments, we simulate the scenario of air quality monitoring. As stated by [21], a 13-dimension vector is provided by each participant that undertakes the sensing task.

6.1 Choices of α and β

According to (6), the parameter α determines the increase speed of the reputation values when providing right sensing data and β determines the decrease speed when the provided sensing data are falsified. Fig. 5 and Fig. 6 show the influence of different values of α and β on updating reputation values.

In Fig. 5, the values of α are 0.02, 0.04, 0.06, 0.08, and 1. From this figure, it is obvious that α determines speed that the reputation value converges to 1 when the participant always provides right sensing data. The larger the value of α, the faster the reputation value converges to 1. However, when α is too large, the participant can obtain a high reputation value only by undertaking a small number of sensing tasks. This makes the proposed scheme fragile to the attack of **A4** (On-Off attack). Hence, α should not be too large in practical applications. In the following experiments, we set $\alpha = 0.2$.

Next, we will analyze the influence of the parameter β. As is shown in Fig. 6, the values of β are 0.1, 0.15, 0.2, 0.25, and 0.3. When the participants persistently provide falsified sensing data, the values of β determines the speed that the

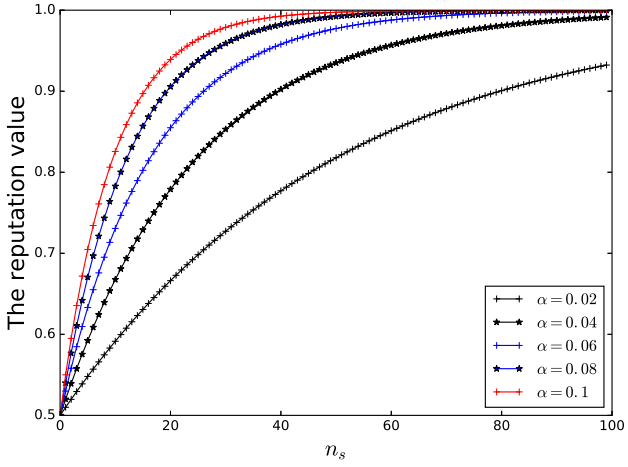

Fig. 5. The reputation value v.s. n_s where α varies from 0.02 to 0.1 with a step size of 0.02.

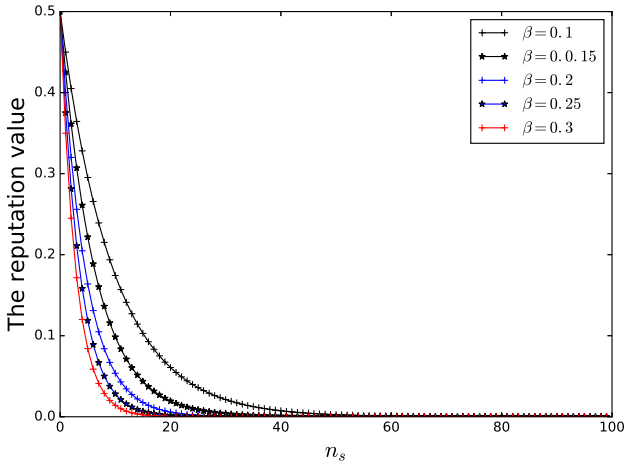

Fig. 6. The reputation value v.s. n_s where β varies from 0.1 to 0.3 with a step size of 0.05.

reputation value converges to 0. The larger β, the faster the reputation value converges to 0. In order to avoid the case that the reputation value drops too much when the participant provides falsified sensing data occasionally due to the failure of the device or affected by the environment, β should not be too large. Moreover, β should not be too small for defending against the On-Off attack. In the following experiments, we set $\beta = 0.2$.

6.2 Running Time Analysis

In this part, we will analyze the time efficiency of the proposed scheme. According to the processes of completing sensing tasks and updating reputation values, the most time-consuming parts are the aggregation of encrypted sensing data and computation of d_i ($i \in \mathcal{P}_\tau$). Thus here, we focus on the running times of these two phase. Let t_{aggre} and t_d denote the time for aggregating encrypted sensing data and computing d_is, respectively. To achieve the Paillier cryptosystem, we use the open source Paillier Library and run this library on a virtual machine with 64-bit Ubuntu operating system, 1G memory and 20G hardware. For the Paillier cryptosystem, the bit length of the modulus is set 2048.

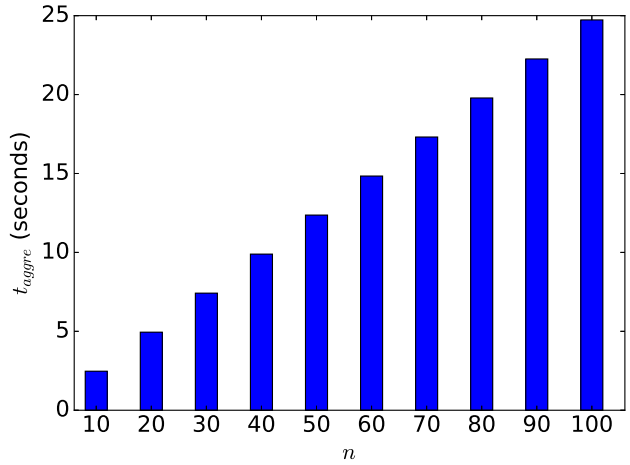

Fig. 7. t_{aggre} v.s. n.

As is shown in Fig. 7, t_{aggre} increases as n increases. Obviously, the larger n, the more encrypted sensing data would be aggregated. This leads to the increase of t_{aggre}. Since the number of multiplication and addition operations linearly increases as n increases, t_{aggre} linearly increases as n increases. When $n = 10$, t_{aggre} is approximately 2.47s. As n reaches 100, t_{aggre} is 24.73s. Comparing with [13], the time efficiency is greatly improved for aggregating encrypted sensing data.

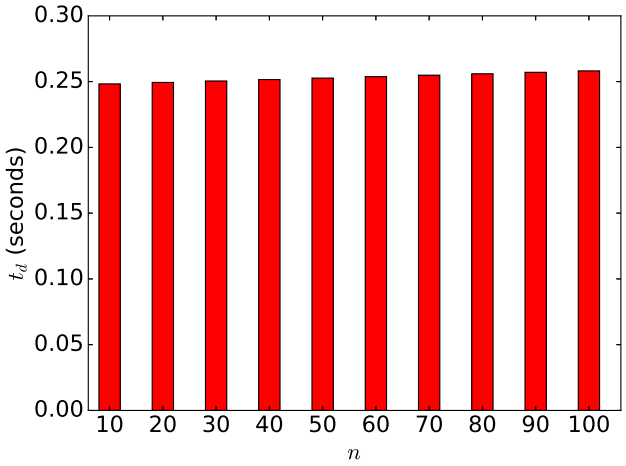

Fig. 8. t_d v.s. n.

Fig. 8 presents how t_d varies as n increases. From this figure, the value of t_d keeps steady as n increases (t_d is around 0.25s). This is because that when computing d_is through (5), we should first encrypt the base vector and then multiply related encrypted values. Here, the most operations are multiplying large numbers which can be easily and efficiently achieved by the Paillier Library. In our experiments, multiplying two large numbers costs about 0.0043ms. Compared with the remaining operations, the running time for multiplications can be neglected.

6.3 The Effectiveness to Defend Against Malicious Participants

At the end of this section, we will verify the effectiveness of the proposed scheme to defend agains malicious participants. Here, the total number of participants that undertake sensing tasks is 100 and the number of sensing tasks to be undertake increases from 1 to 100. The proportion of malicious participants, denoted by P_{mal} increases from 0.1 to 0.4 with a stepsize of 0.1. Moreover, the absolute mean error between the final aggregating results and actual sensing data, referred as MAE, is introduced to reflect the effectiveness of the proposed scheme.

Fig. 9 presents how MAE varies as the number of sensing tasks n_s increases. Since the initial reputation values of the participants are all set to 0.5, the weights of the sensing data from normal and malicious participants are very similar when n_s is small. Here, the values of MAE are the largest ones. As n_s increases, the proposed scheme begins to work. The reputation values of normal participants converges to 1 and those of malicious ones converges to 0. At this time, the weights of the sensing data from malicious participants converge to 0. This makes the values of MAE decrease and converge to 0 as n_s increases. Note

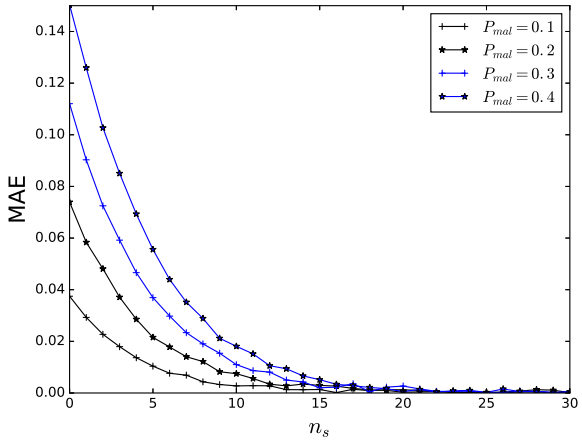

Fig. 9. MAE v.s. n_s.

that in order to more clearly present the variation of MAE, we only give the cases when $n_s \leq 30$.

7 Conclusion

In this paper, we propose a decentralized privacy-preserving reputation management scheme for MCS on the basis of blockchain. From the perspective of performance consideration, we introduce the consortium blockchain and the edge computing paradigm in our system model. The geographically distributed edge nodes are responsible for processing sensing data and maintaining the blockchain. To efficiently and privately aggregating sensing data, we design a novel rules for updating reputation values by leveraging the Paillier crytosystem. In the experiments, the cost efficiency and the effectiveness to deal with malicious participants are verified. These experimental results give a direct guidance to how the proposed scheme can be adopted by practical applications.

8 Acknowledgements

This work is supported by the National Key Research and Development Program of China under Grant 2016YFB0800601 and the Key Program of NSFC-Tongyong Union Foundation under Grant U1636209. Q. Pei is the corresponding author.

References

1. Yang D , Xue G , Fang X , et al. Incentive Mechanisms for Crowdsensing: Crowdsourcing With Smartphones. IEEE/ACM Transactions on Networking **24**(3):1732-1744 (2016).
2. Ni J , Zhang A , Lin X , et al. Security, Privacy, and Fairness in Fog-Based Vehicular Crowdsensing. IEEE Communications Magazine **55**(6):146-152 (2017).
3. Min M, Reddy S, Shilton K, et al. PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In: the 7th ACM International Conference on Mobile Systems, Applications, and Services, pp. 55-68. ACM, Krakow, Poland (2009).
4. Basudan S, Lin X, Sankaranarayanan K. A Privacy-Preserving Vehicular Crowdsensing-Based Road Surface Condition Monitoring System Using Fog Computing. IEEE Internet of Things Journal **4**(3): 772-782 (2017).
5. Hicks J, Ramanathan N, Kim D, et al. AndWellness:an open mobile system for activity and experience sampling. In: the 1st ACM SIGMOBILE International Conference on Pervasive Computing Technologies for Healthcare, pp. 34-43. ACM, Munich, Germany (2010).
6. Fan J, Li Q, Cao G. Privacy-aware and trustworthy data aggregation in mobile sensing. In: the 3rd IEEE Conference on Communications and Network Security, pp. 31-39. IEEE, Florence, Italy (2015).
7. Pei Q, Ma L, Li H, et al. Reputation-based coalitional games for spectrum allocation in distributed cognitive radio networks. In: 2015 IEEE International Conference on Communications, pp. 7269-7274.. IEEE, London, United Kingdom.
8. Liu Y, Sun Y L. Securing digital reputation in online social media. IEEE Signal Processing Magazine **31**(1): 149-155 (2014).
9. Wang X O, Cheng W, Mohapatra P, et al. Enabling reputation and trust in privacy-preserving mobile sensing. IEEE Transactions on Mobile Computing **13**(12): 2777-2790 (2014).
10. Li Q, Cao G, La Porta T F. Efficient and privacy-aware data aggregation in mobile sensing. IEEE Transactions on dependable and secure computing **11**(2): 115-129 (2014).
11. Lu Y, Tang Q, Wang G. Zebralancer: Private and anonymous crowdsourcing system atop open blockchain. In: the 38th IEEE International Conference on Distributed Computing Systems, pp. 853-865. IEEE, Vienna, Austria (2018).
12. Li H , Chen Q , Zhu H , et al. Privacy Leakage via De-anonymization and Aggregation in Heterogeneous Social Networks. IEEE Transactions on Dependable and Secure Computing, early access (2017).
13. Ma L , Liu X , Pei Q , et al. Privacy-Preserving Reputation Management for Edge Computing Enhanced Mobile Crowdsensing. IEEE Transactions on Services Computing, early access (2018).
14. Li Z, Kang J, Rong Y, et al. Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things IEEE Transactions on Industrial Informatics, early access (2017).
15. Ruinian L , Tianyi S , Bo M , et al. Blockchain For Large-Scale Internet of Things Data Storage and Protection. IEEE Transactions on Services Computing, early access (2018).
16. Yang Z , Yang K , Lei L , et al. Blockchain-based Decentralized Trust Management in Vehicular Networks. IEEE Internet of Things Journal, early access (2018).
17. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Working paper (2008).

18. Paillier P . Public-key cryptosystems based on composite degree residuosity classes. Advances in Cryptology EUROCRYPT99: 223-238 (1999).
19. Yi X , Bouguettaya A , Georgakopoulos D , et al. Privacy Protection for Wireless Medical Sensor Data. IEEE Transactions on Dependable and Secure Computing **13**(3): 369-380 (2016).
20. Kang X , Wu Y . A trust-based pollution attack prevention scheme in peer-to-peer streaming networks. Computer Networks **72**: 62-73 (2014).
21. Vito S D, Piga M, Martinotto L, et al. On Field Calibration of an Electronic Nose for Benzene Estimation in an Urban Pollution Monitoring Scenario. Sensors and Actuators B Chemical, **143**(1):182-191 (2009).